



Cisco BTS 10200 Softswitch PCSP Programmer's Guide, Release 7.0

July 2010

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

Americas 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: 408 527-0883

Customer Order Number:
Text Part Number: OL-23044-01

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco BTS 10200 Softswitch PCSP Programmer's Guide, Release 7.0
© 2010 Cisco Systems, Inc. All rights reserved.



CONTENTS

Preface vii

Organization vii

Obtaining Documentation and Submitting a Service Request vii

Document Change History viii

CHAPTER 1

Overview 1-1

Scope 1-1

Strategy 1-1

Terms 1-2

CHAPTER 2

SOAP/XML Environment 2-1

Compiler Tools 2-1

WSDL Stub Generation 2-2

CHAPTER 3

Data Model 3-1

Core Data Model 3-1

PcspService 3-3

PcspMta 3-4

PcspEndpoint 3-4

PcspCms 3-4

PcspImportExport 3-4

PcspRelation 3-5

Cisco Extensions 3-5

Standard PCSP Extensions 3-5

PcspService 3-5

PcspCms 3-9

PcspMta 3-10

PcspEndpoint 3-10

PcspObject Common Response 3-10

PCSP Standards Compatible Extension for BTS/XML 3-10

CHAPTER 4

PCSP SOAP API 4-1

Login/Logout: User Security API 4-1

EMS Admissions Control Policy 4-2

PCSP API 4-3

PcspCiscoBtsCli API 4-3

CHAPTER 5

PCSP Rules and Mapping 5-1

PcspService 5-1

PcspMta 5-4

PcspEndPoint 5-5

PcspCms 5-6

Provisioning Rules 5-7

Subscriber 5-7

MTA 5-8

Endpoint 5-9

PCSP Extensions Requirements 5-9

PcspCms Extension 5-10

PcspMta Extension 5-11

PcspService Extension 5-12

PcspEndpoint Extension 5-15

CHAPTER 6

PCSP Objects – XML Example (PUT) 6-1

WSUS 6-1

PcspCms 6-1

PcspMta 6-2

PcspService 6-3

PcspEndpoint 6-5

PcspCiscoBtsCli 6-6

Request Message, Example 1 6-6

Response Message, Example 1 6-7

Request Message, Example 2 6-8

Response Message, Example 2 6-9

CHAPTER 7

Performance Considerations 7-1

Performance 7-1

CHAPTER 8

PCSP Subscriber Provisioning 8-1

Provisioning a Subscriber with the Minimum Number of Features 8-1

Provisioning a Subscriber with a Feature 8-4

Modifying a Subscriber Feature 8-7

Adding Features to a Subscriber 8-8

CHAPTER 9**Troubleshooting 9-1**

Tcpmon Tool 9-1

SDK Tool 9-3

Common Security Certificates 9-4

APPENDIX A**Cisco BTS 10200 Extensions for PcsP Objects A-1**

PcsPcMs A-1

PcsPMta A-2

PcsPService A-3

PcsPEndpoint A-10

PcsPCiscoBtsCli A-10

APPENDIX B**Sample Entity Encodings B-1**

PcsPService Object Example B-1

PcsPEndpoint Object Example B-5

PcsPMta Object Example B-6

PcsPCms Object Example B-7



Preface

Revised: July 2010, OL-23044-01

This document serves as the PacketCable Subscriber Provisioning (PCSP) Programmer's Guide for the Cisco BTS 10200 Softswitch.

Organization

This document is divided into the following chapters and appendixes:

- Chapter 1—[Overview](#)
- Chapter 2—[SOAP/XML Environment](#)
- Chapter 3—[Data Model](#)
- Chapter 4—[PCSP SOAP API](#)
- Chapter 5—[PCSP Rules and Mapping](#)
- Chapter 6—[PCSP Objects – XML Example \(PUT\)](#)
- Chapter 7—[Performance Considerations](#)
- Chapter 8—[Troubleshooting](#)
- Appendix A—[Cisco BTS 10200 Extensions for Posp Objects](#)
- Appendix B—[Sample Entity Encodings](#)

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at

<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS Version 2.0.

Document Change History

The following table lists the revision history for the *Cisco BTS 10200 Softswitch PCSP Programmer's Guide, Release 7.0*.

Version Number	Issue Date	Status	Reason for Change
OL-23044-01	July 2010	Initial	Initial document for Release 7.0.



CHAPTER 1

Overview

Revised: July 2010, OL-23044-01

The purpose of this document is to provide high level description of the provisioning interface for the Cisco BTS 10200 Softswitch using the PacketCable provisioning specification (PKT-SP-CMSPROV1.5-I01-050128). This document is not intended to restate the PacketCable specification. However, sufficient detail is provided so that you can understand the relationship between PacketCable objects and the Cisco BTS 10200 command set and database.

Scope

The scope of this document is to provide a description of the provisioning interface that is sufficient to allow for you to provision subscribers and to manage services as described in the customer *PacketCable Simple Object Access Protocol/Extensible Markup Language (SOAP/XML) Product Requirements Document (PRD)*. This includes the ability to execute Cisco BTS 10200 XML commands over extensions to the PacketCable interface.

Strategy

The basic PacketCable specification does not enough definitions for you to manage a Cisco BTS 10200 in its entirety. Using the standards based extensions as defined by the PacketCable specification, sufficient additions are provided in order for a subscriber to manage the Cisco BTS 10200 system. In addition, a standardized format for regular Cisco BTS 10200 command execution is also provided.



Note

The reader of this document needs to have a working understanding of the PacketCable specification and of the Web Services general operation.

Terms

The PacketCable specification contains common terms and definitions used in this book. Additionally, the following terms are used in this book.

- **BTS/XML**—A Cisco BTS 10200 SOAP/XML implementation using a Cisco BTS 10200 proprietary XML schema derived from the Cisco BTS 10200 command semantics
- **CLI**—Command-line interface
- **PCSP**—Packet Cable Subscriber Provisioning
- **WSUS**—Web Services User Security



CHAPTER 2

SOAP/XML Environment

Revised: July 2010, OL-23044-01

The Simple Object Access Protocol/Extensible Markup Language (SOAP/XML) Interface Adapter conforms to the SOAP 1.2 specification.

Compiler Tools

The minimum required compiler is the J2SE Development Kit (JDK)-1.4.1. The Cisco BTS 10200 Softswitch uses JDK 1.5.0 for compilation and JDK 1.5.0 or JDK5 for the Java Runtime Environment (JRE). Additional tools that might be required for the client side application are

- Xerces Parsers
- Event Control System (ECS) Report Builder
- Apache Coyote
- Apache Tomcat
- Apache Web Services including AXIS 1.3

Cisco Systems, Inc. does not provide these tools. They are generally available on the Internet from various sources. For example, <http://www.apache.org>.

WSDL Stub Generation

The SOAP/XML package uses the Apache AXIS toolkit to generate skeletons and stubs from the Web Service Description Language (WSDL) file. These skeletons and stubs are used to implement the abstraction between the SOAP Interface with the Cisco BTS 10200 middleware and code.

This code example uses the Java package tree as developed in the Cisco BTS 10200 product. This code can vary. Other clients can specify a different package tree to contain the WSDL interface objects. See the software development kit (SDK) for a detailed breakdown of this script.

```
#!/bin/sh
#####
# Copyright (c) 2002, 2006 by Cisco Systems, Inc.
#
#
#####
set -e
set -a
#set -x
#
# List required jar files
#
AXISLIB=/opt/Tomcat/webapps/axis/WEB-INF/lib
CLASSPATH=$AXISLIB/axis.jar:$AXISLIB/commons-logging-1.0.4.jar:$AXISLIB/jaxrpc:$AXISLIB/commons-discovery-0.2.jar:$AXISLIB/saaj:$AXISLIB/wsdl4j-1.5.1.jar
export CLASSPATH
java -classpath $CLASSPATH org.apache.axis.wsdl.WSDL2Java -o . -p
com.sswitch.oam.soap.client bts10200.wsdl
```

Java files are generated in a local directory tree specified in the package directory. This package path is required in the bind logic to find the object interface implementation.

```
#!/bin/sh
#####
# Copyright (c) 2002, 2006 by Cisco Systems, Inc.
#
#
#####
set -e
set -a
#set -x
#
# List required jar files
#
AXISLIB=/opt/Tomcat/webapps/axis/WEB-INF/lib
CLASSPATH=$AXISLIB/axis.jar:$AXISLIB/commons-logging-1.0.4.jar:$AXISLIB/jaxrpc:$AXISLIB/commons-discovery-0.2.jar:$AXISLIB/saaj:$AXISLIB/wsdl4j-1.5.1.jar
export CLASSPATH
javac -classpath $CLASSPATH -d ./ com/sswitch/oam/soap/client/*.java
```

Compile a package of Java files to generate the required class files. These class files must exist in the client classpath.

Sample Makefile(s) and Apache jar files are included in the Cisco BTS 10200 XSDK.



CHAPTER 3

Data Model

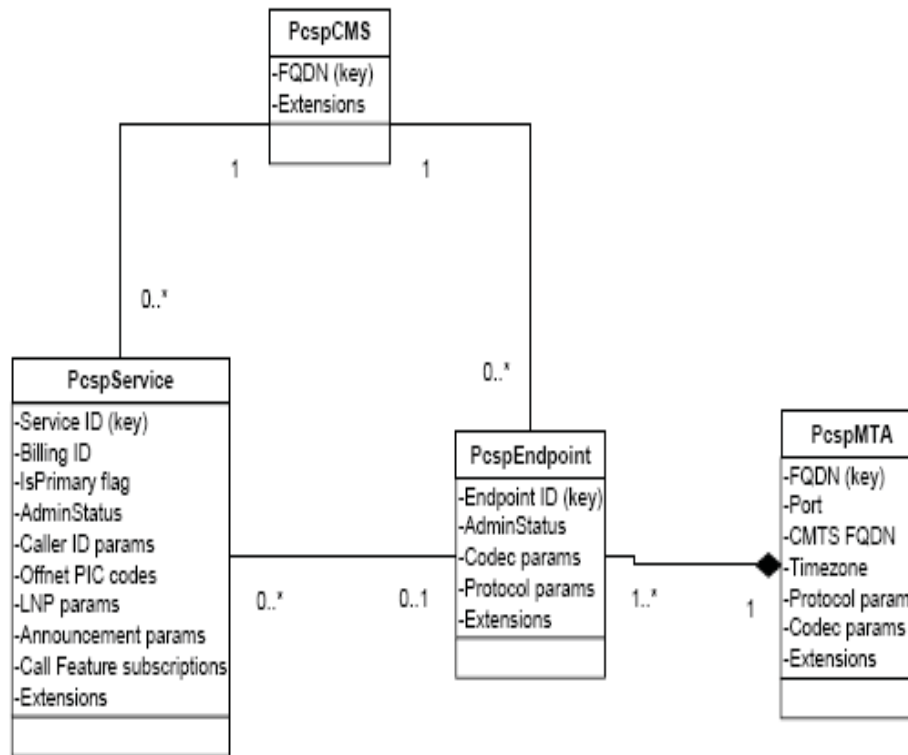
Revised: July 2010, OL-23044-01

This document does not cover the basics of PacketCable (PC) data model provisioning. These items are covered in different specifications. Provisioning operations for the Cisco BTS 10200 Softswitch utilize the PKT-PROV-P1 interface only. All objects referenced in this document are considered part of this interface only. Note that all Call Management Server (CMS) references here relate to the Cisco BTS 10200 complex.

Core Data Model

The Cisco BTS 10200 leverages the basics of the PC data model for subscribers and services. [Figure 3-1](#) shows the defined relationships of the various objects and their references.

Figure 3-1 CMS Provisioning Data Model



The Cisco BTS 10200 implementation requires the following behavior from these objects:

- The PcpService object is the entity to which a PacketCable 1.5 customer subscribes. It represents a phone number and all related functionality (call features and so on).
- A PcpMta object represents a media terminal adapter (MTA), which aggregates one or more endpoints physically contained within the MTA. For the Cisco BTS 10200 implementation, this object supports the primary protocols such as Session Initiation Protocol (SIP) and Media Gateway Control Protocol (MGCP).
- The PcpEndpoint object represents a physical endpoint on an MTA/Gateway. This object encompasses the behaviors of subscriber terminations (endpoints) in the Cisco BTS 10200 for residential and small business media gateway (MGW) types expressed in the PcpMta.
- A PcpCms object maintains associations between endpoints/CMSs and services/CMSs. This object within the scope of the Cisco BTS 10200 is effectively the foreign key linkage between the endpoint and subscriber. This is already established in the schema of the Cisco BTS 10200. However, this is leveraged for functions like equipage of a subscriber endpoint.
- PcpRelations represent associations between objects. In Figure 3-1, they are presented as connections between objects.

These objects cover the primitives of the following tables. This is not complete coverage of the required fields as defined in the Cisco BTS 10200 system configuration. Therefore, extensions are required to complete the coverage. One key item to note for this interface is that the addition of features is not in scope. Table 3-1 lists the PacketCable object to Cisco BTS 10200 table comparisons.

Table 3-1 PacketCable Object Name to Cisco BTS 10200 Table Name Comparisons

PacketCable Object Name	Cisco BTS 10200 Table Name
MTA	MGW
Endpoint	TERMINATION
Service	SUBSCRIBER SUBSCRIBER_SERVICE_PROFILE SERVICE SUBSCRIBER_FEATURE_DATA SLE SC1D SC2D DN2SUBSCRIBER
Cms	CALL_AGENT

PcspService

The PcspService object must support the following attributes. The following definitions for the field might apply to the view of the PC and not necessarily the Cisco BTS 10200.

- ServiceId—A unique identifier for the service.
- BillingId—The identifier of another service, which will be billed for activity on this service.
- IsPrimary Flag—With multiple services provisioned upon an endpoint, one service *must* have this flag set to indicate the default service to use for outgoing calls.
- PrimaryRingPattern—Index into MTA cadence table and ring pattern selection for this service.
- Administrative status of this service—Suspended, enabled, number has changed. and so forth).
- DisplayName—The display information used for Call Name Delivery feature (CNAM).
- DisplayNumber—The display information used for Call Number Delivery feature (CND).
- Announcement settings—Enable, language, time zone, and so forth.
- Carrier codes—Long distance carrier code, intra-lata carrier code, and international carrier code.
- Local number portability control—Porting status, signal transfer point (STP) lookup flag, and so forth.
- Call features—A service includes a list of subscribed call feature objects
- Extensions—This object is extensible in two locations: the main body of the object and the call feature list.

PcspMta

The PcspMta object must support the the following attributes:

- MTA's Fully Qualified Domain Name (FQDN), uniquely identifying this MTA.
- MTA's Network-Based Call Signaling protocol (NCS) listener port (default: 2427)
- FQDN of controlling cable modem termination system (CMTS).
- The time zone within which this MTA is physically located.
- Signaling protocol designation—This is the default protocol selection for all contained endpoints, unless overridden by an individual endpoint.
- Codec designation—Default codec selection for all contained endpoints, unless overridden by an individual endpoint.
- Internet Protocol (IP) security (IPSec) Control Flag—The IPSec Control Flag indicates whether IPSec is used for NCS Signaling between the CMS and the MTA. By default, IPSec is turned on for all endpoints, but it can be provisioned on a per endpoint basis.
- MTA Profile Name—Optional; an MTA Profile Indicator identifiable by the CMS.
- A single point for extension.

PcspEndpoint

The PcspEndpoint object must support the the following attributes:

- EndpointId—Uniquely identifies this endpoint.
- Signaling protocol selection—Optionally overrides MTA setting.
- Administrative status of the endpoint—Disconnected, normal service, test mode, and so forth.
- Codec selection—Optionally overrides MTA setting.
- IPSec Control Flag—Optionally overrides the MTA setting.
- A single point for extension.

PcspCms

The PcspCms object must support the following attributes:

- FQDN uniquely identifying this CMS
- A single point for extension

PcspImportExport

The PcspImportExport is a general purpose document format that can contain a large number of objects or relations. It is typically used to export full data sets from a Provisioning Server (PS) to a CMS. The PcspImportExport document format is not a supported object in this release of the Cisco BTS 10200.

PcspRelation

The PcspRelation is not a supported object in this release of the Cisco BTS 10200.

Cisco Extensions

To complete the coverage of subscriber management with PacketCable Subscriber Provisioning (PCSP), the required extensions refine the Cisco BTS 10200 vendor-specific behavior and attributes. These are generally part of the basic object listed above. However, there are not enough attributes to cover the full behavioral expression of the Cisco BTS 10200.

The detailed schema definitions of the extensions are listed in [Appendix A, “Cisco BTS 10200 Extensions for Pcs Objects.”](#)

Standard PCSP Extensions

For a more detailed description of these components and how they relate to the Cisco BTS 10200 database, see the [Cisco BTS 10200 Softswitch CLI Database](#).

PcspService

This section provides the PcspService extensions.

- sub_profile_id—Foreign key: Subscriber Profile table. Subscriber profile ID.
- account_id—Specifies a subscriber account ID for one or more subscribers. An account id is only for identification purposes. The account ID is not associated with any feature grouping behavior during call processing, although it is included in a billing record.
- address1—Street name, apartment number.
- address2—Street name, apartment number.
- aor_id—Mandatory and permitted term-type = SIP. Foreign key: Address of Record table. The address of record id. Uses the domain name parser. Domain name portion of the AOR-ID must exist in the Serving Domain Name table. AOR =USER +@+DOMAIN-NAME.
- billing_type—Specifies the subscriber billing type. Permitted values are
 - FR1—Flat Rate Billing 1
 - FR2—Flat Rate Billing 2
 - MR1—Measured Rate Billing 1
 - MR2—Measured Rate Billing 2
 - NONE (Default)—Billing type is not used

- category—Subscriber category. Permitted values are
 - INDIVIDUAL (Default)—Individual subscriber
 - CTXG—Assigned to the main subscriber ID of a Centrex group
 - CTXG-INDIVIDUAL—Assigned to a Centrex subscriber
 - CTXG-MLHG—Assigned to a Centrex multiline hunt group (MLHG) (for example, attendant)
 - CTXG-TG—Assigned to a Centrex trunk group
 - IVR—Access directory number (DN) for interactive voice response (IVR)
 - MLHG-INDIVIDUAL—Subscriber within an MLHG
 - MLHG—Main subscriber ID of an MLHG
 - MLHG-PREF-INDIV—Main subscriber ID of a preferential hunt list
- PBX—Assigned to the main subscriber ID of a private branch exchange (PBX).
- RACF—Access DN for remote activation of call forwarding (RACF).
- city—Subscriber's city.
- cos_restrict_id—Foreign key: class of service (COS) Restrict table. COS restriction ID.
- country—Country of subscriber.
- ctxg_id—Foreign key: Centrex Group table. Same as ID in Centrex Group table.
- cwt_type—Call Waiting Tone type. For the Multiple DN (MDN) feature, a different Call Waiting Tone type can be assigned to each DN.
- esrn—Emergency service routing number. The subscriber level emergency service routing number (ESRN) is used if available; otherwise, the ESRN assigned to the subscriber profile is used.
- forced—Use to bypass rule checking during add or change operations. When this extension is used in a delete command, the Cisco BTS 10200 deletes all the information associated with the subscriber from the Subscriber table as well as from all associated tables such as the Subscriber Service Profile table and the Subscriber Feature Data table.
- grp—Specifies whether a subscriber is an individual or a group. Applies to Multiline Hunt or Centrex subscribers. (Default = N.)
 - Y—Group
 - N—Individual
- h323_term_id—Mandatory if term-type = H323. Foreign key: H.323 Termination table. The termination ID for an H.323 terminal.
- immediate_release—Specifies whether to apply the immediate release procedure when a calling party hangs up. (Default = N.)
- mac_id (EMS only)—A MAC-ID is allowed only if term-type=sip. This token is case sensitive and should be entered in uppercase. Foreign key: MAC to Subscriber table. MAC-ID of SIP (IP) Phone. Specifies the MAC address of the subscriber's SIP phone from which services to control features are invoked.
- mgw_id—Mandatory and permitted if term-type = term. Foreign key: Media Gateway, Termination tables. Media Gateway ID. The mgw-id is used as a termination ID as well as a combined foreign key to the termination table.
- mlgh_id—Foreign key: Multiline Hunt Group (MLHG) table. Same as ID in Multiline Hunt Group table.

- `mlhg_pref_list_id`—Foreign key: MLGH Preference List table. Same as ID in MLGH Preference List table.
- `ohd_timer`—Off-hook Delay Timer. If an off-hook trigger is set, the Cisco BTS 10200 looks at the `ohd-timer` token and applies a dial-tone timer equal to `ohd-timer`. If a user dials any digit within this timer value, the remaining digits are collected as normal. If no digit is dialed before the `ohd-timer` expires, the Cisco BTS 10200 Softswitch performs off-hook trigger processing.



Note Applies only if `offhook-trigger-type=OHD`. Default 0, or 2-16 seconds.



Note If 0 is provisioned, then the default value in the Call Agent Configuration table is used.

- `offhook_trigger_type`—Indicates whether an off-hook trigger applies to this subscriber. A string of 1-8 ASCII characters (Default = NONE). Permitted values are
 - OHD—Off-hook Delay trigger.
 - OHI—Off-hook Immediate trigger.
 - NONE (Default)—No Off-hook trigger.



Note Only allowed if `term-type=term`.

- `policy_id`—Mandatory and permitted `term-type = SIP`. The route policy id.
- `privacy`—Specifies how calling party information (Name, DN) is displayed to the called party. Permitted values are
 - NONE (Default)—Name and number.
 - FULL—Do not display name or number.
 - NAME—Do not display name.
 - USER—Use user-provided privacy information. Applies only to SIP endpoints that can include privacy information. If information is not received for either name or number, then privacy is indicated as "unspecified." `PRIVACY = USER` is not valid if `term-type = term` or `none`.
- `privacy_manager_id`—Foreign key: Application Server table. The privacy manager id.
- `qos_id`—Foreign key: QOS table. Specifies whether to use QOS index for codec selection.
- `sdt_mwi`—Specifies whether a stutter tone is provided as a Message Waiting Indication. A value of Y or N (Default = Y).
 - Y—A stutter dial tone is provided to a user when the subscriber goes off-hook and subscriber has Message Waiting.
 - N—A stutter dial tone is not provided.
- `secure_fqdn`—(EMS Only.) Permitted only if `term-type=sip`. Use to resolve an IP address and compare it with the IP address received from an endpoint during registration or during call setup (INVITE). A `secure-fqdn` is assigned to the AOR.

- `send_bdn_as_cpn`—Specifies whether to send the billing DN for all calls (EMG, LOCAL, ON-NET, and OFF-NET) as the CPN in the outgoing setup messages. If a billing DN is not provisioned, the billing DN (present) or the DN1 of the main subscriber (for PBX/MLHG subscribers) is sent as the CPN in the outgoing setup message. If none of them are found, the DN1 is sent. The selection of Y or N is permitted. (Default = N.)
 - Y—Send the billing DN of a subscriber as a CPN in the outgoing setup message.
 - N—Send the DN1 of a subscriber as a CPN in the outgoing setup message.
- `send_bdn_for_emg`—Specifies whether to send the billing DN value of emergency calls for PBX over PRI, or CAS and ANALOG, and subscribers information as the CPN in the outgoing setup messages. The selection of Y or N is permitted. (Default = N.)
 - Y—Send the billing DN of the subscriber as a CPN in the outgoing setup message.
 - N—Send the DN1 of the subscriber as a CPN in the outgoing setup message.
- `state`—State designation.
- `static_contact_host`—Hostname portion of the contact id as in user@hostname.
- `static_contact_port`—Port number if different from default SIP port number (5060).
- `static_contact_user`—User portion of the contact id as in user@hostname.
- `term_id`—Mandatory and permitted term-type = term. Unique key. Foreign key: Termination table. Termination ID. Required if term-type=term. The term-id is used as a termination ID and as a combined foreign key to the termination table.
- `termination_immediate_rel`—Specifies whether a call is immediately released when a called party terminates a call.
- `term_type`—Specifies the termination type. Permitted values are
 - TERM (Default)—MGCP termination
 - TG—Trunk group
 - ROUTE—Route
 - RG—Route guide (Not supported)
 - SIP—SIP termination
 - NONE—There is no termination associated with the subscriber.
 - H323—H.323 virtual termination
- `tgn_id` (or `tg`)—Mandatory and permitted term-type=tg. Foreign key: Trunk Group table. Trunk group ID. This field can also be provisioned by the use of `tg` instead of `tgn-id`. The EMS looks up the `tgn-id` based on the trunk group and then provisions it.
- `tg` (or `tgn_id`)—Mandatory and permitted term-type=tg. Foreign key: Trunk Group table. Trunk group ID. This field can also be provisioned by the use of `tg` instead of `tgn-id`. The EMS looks up the `tgn-id` based on the trunk group and then provisions it.

- `usage_sens`—Specifies if usage-sensitive features are allowed. Usage-sensitive features are not subscribed but are available for use. The subscribers are charged for each use. The following features are available as usage-sensitive:
 - AR—Automatic recall
 - AC—Automatic callback
 - USTWC—Three-way callingWhere:
 - Y—Usage-sensitive features are allowed.
 - N—Usage-sensitive features are not allowed.
- `vmwi`—The visual message waiting indication. This indication is sent when a message waiting indicator (MWI) notify is received from the voice-mail system.
- `voice_mail_id`—Foreign key: Application Server table. Specifies the default voice-mail id for all subscribers belonging to a subscriber profile.
- `zipcode`—Subscriber's zip code.
- `user_type`—Defines the type of user. Permitted values are
 - IP (Default)
 - PHONE
- `host_ip_address`—IP address of the Element Management System (EMS) machine servicing the response. This extension is only applicable in response object.
- `host_id`—Configured EMS hostname of the EMS machine servicing the response. The extension is only applicable in response object.

PcspCms

This section provides the PcspCms extensions.

- `id`—Primary key. Call Agent ID (domain name). Service provider assigns, based on network configuration.
- `cli`—Foreign key: CLLI Code table. The Common Language Location Identifier (CLLI) for the Call Agent.
- `mgw_monitoring_enabled`—Media Gateway monitoring enabled/disabled indicator.
- `host_ip_address`—IP Address of the EMS machine servicing the response. This extension is only applicable in response object.
- `host_id`—Configured EMS hostname of the EMS machine servicing the response. This extension is only applicable in response object.

PcspMta

This section provides the PcspMta extensions.

- `id`—Primary key. Media gateway identifier, assigned by the service provider.
- `type`—Type of gateway.
 - `RGW`—Residential gateway
 - `TGW`—Trunking gateway
- `aggr_id`—ID of the aggregation device cable modem termination system (CMTS). This token is mandatory if PacketCable DQoS is supported; it is how the Cisco BTS 10200 call management server (CMS) determines the CMTS to which a media terminal adapter (MTA) is attached, so it can issue gate control commands to the correct CMTS.
- `node`—Defines the hybrid fiber coax (HFC) fiber node the MTA is homed to. The HFC fiber node sits between the CMTS and the MTA. Every MTA is assigned to a node.
- `host_ip_address`—IP Address of the EMS machine servicing the response. Is only applicable in response object.
- `host_id`—Configured EMS hostname of the EMS machine servicing the response. Is only applicable in response object.

PcspEndpoint

This section provides the PcspEndpoint extensions.

- `sub_id`—Foreign key: Subscriber table. Subscriber ID of line termination: same as the ID in the Subscriber table.

PcspObject Common Response

This section provides the PcspObject common responses.

- `timestamp`—The reply is timestamped when the Cisco BTS 10200 EMS is finished processing the request. This extension is only applicable in a response object.
- `reply_string`—Reply string as seen from Cisco BTS 10200 EMS notifying success or failure and details of failure if available. This extension is only applicable in a response object.

PCSP Standards Compatible Extension for BTS/XML

The extensible markup language (XML) format of data used in the BTS/XML and Common Object Request Broker Architecture (CORBA)/XML interfaces is still valid. The Cisco BTS 10200 provides backward compatible behavior for all existing interfaces and includes support for this data construct as an extension to the PacketCable interface as well.

The full nature of the interface is defined in the following chapters. The schema is defined in [Appendix A, “Cisco BTS 10200 Extensions for Pcsp Objects.”](#)



CHAPTER 4

PCSP SOAP API

Revised: July 2010, OL-23044-01

The PacketCable specification details the Simple Object Access Protocol (SOAP) message interface. This message specification is the “defacto application programming interface (API)” to the Cisco BTS 10200 Softswitch through this interface.

User level security or Web Services User Security (WSUS) is a separate service offering that enables the Cisco BTS 10200 user security within the confines of PacketCable provisioning.

Login/Logout: User Security API

The Web Services Description Language (WSDL) for the security interface is defined below. A full listing of the WSDL is provided in the appendix. The WSUS service is provided to create user login sessions that parallel the behavior of login sessions of the Cisco BTS 10200 CLI. This allows for the application of command authorization and authentication for the PacketCable service.



Note

In the PacketCable specification, user or session level security is not addressed. There is a need for these capabilities which allow for differentiation of users. This need may be addressed in a future release.

The interface offers two basic SOAP message containers and one generic fault container. The service name is “Bts10200SecurityService” and the default URL for the service is <https://localhost/wsus/services/bts10200Security>

The service coexists with the PacketCable Subscriber Provisioning (PCSP) service and has the following messages:

- **login**—This is the basic login function that authenticates and returns a session key.
- **loginWithStatus**—This function processes a normal login. It also checks for warnings on the expiration of the password and returns a status string indicating the status of the password. For example, if the password is about to expire, a warning status is returned. A session key is returned as part of the function.

- **loginResetPassword**—This function allows the user to access the system and log in under normal authentication. It resets the user password to a new value. This clears any warning indication about password expiration.
- **logout**—This terminates any session associated with the supplied session key.
- **BtsUserException** – This is the general fault defined in the Bts10200SecurityService. It contains a descriptive string and numeric value for the associated fault that occurred in the WSUS service.

EMS Admissions Control Policy

The EMS Admissions Control Policy (EACP) feature defines the policies basis on which it is determined whether a BTS EMS user is allowed to access and use the OAM&P functions of the BTS 10200. One policy is used to apply a user based limit on EMS sessions. It determines the limit on number of simultaneous sessions (through a specific adapter) that can be opened for a user. Once the policy set limit is reached, no more new sessions from that user are allowed. This policy is applied on each adapter independently.

Current specifications of the EACP are:

- EMS users are subject to the EACP based on a per adapter-type basis. Currently only the PCSP (Packet Cable for SOAP) addapter is supported.
- The EMS user session-limit is configurable on a combination of user and adapter-type basis.
- For each admissions-denied condition, an appropriate event (no clearance) is raised (OSS 24). The event information includes user-id, adapter-type, max-limit, #denials-so-far (since adapter restart), date, time, and EMS-SIDE. Additionally, a trace log with similar information is also logged.
- Only users with command-level 10 are able to reconfigure (CHANGE) the user session limit. But all EMS users are able to view (SHOW) the session limits.
- Only users with command-level 10 are able to reconfigure (CHANGE) the default user session limit for an adapter. But all EMS users are able to view (SHOW) the default session limit.
- The EACP feature is able to show EMS users a counter of the number of times a user has been denied entry since the adapter was started or restarted. This information is viewable using the **show user_session_limit** command.
- On switchover, the static status of EACP takes effect on the new active side. The static status includes user changeable provisioning SESSION_LIMIT and USE_DEFAULT_LIMIT parameters.
- On dual-outage conditions and subsequent restart, the dynamic status is reset. The dynamic status is defined as denials.
- On a restart not involving a dual-outage, the static status of the mate EMS (active-side) takes effect on the standby side.
- The EACP is limited to PC 1.5 Simple Object Access Protocol/Extensible Markup Language (SOAP/XML) sessions only in BTS 6.0. The EACP for generic SOAP/XML and CORBA will be added later.
- The default EMS user session-limit is configurable on adapter-type basis. The initial default user session-limit of PC 1.5 SOAP/XML is 10. The range of available sessions is 1–20, with the default of 10 sessions.

The EACP feature capability is part of the standard BTS 10200 deployment for Release 6.0 and later.

PCSP API

The WSDL for this interface is defined below. A full listing of the WSDL is provided in the appendix. The interface offers three basic SOAP messages and one generic fault container.

The basic PacketCable Subscriber Provisioning or PCSP service is accessible through the general URL format: <https://localhost/pcsp/services/PcspI01Service>.

The interface offers three basic message constructs (listed below). The three constructs provide all the defined input and output for the basic PCSP service.

- **PUT**—The PUT message accepts an array of homogenous objects to be processed as either an add or a change function. The type of function is defined in the command status field. This alters or adds elements in the Cisco BTS 10200 database. A response consisting of homogeneous argument objects is returned.
- **GET**—The GET message accepts a homogenous array of arguments defined as “keys” to indicate which database elements are to be returned in this request. The reply is a homogenous array of objects generated from the show or display request.
- **DELETE**—The DELETE message accepts an array of homogenous arguments defined as “keys” to indicate which elements of the database are to be removed. A response consisting of argument objects is returned to indicate the success or failure of the request.

PcspCiscoBtsCli API

The provisioning of PcspCiscoBtsCli is a propriety extension to PacketCable1.5 specific for Cisco Cisco BTS 10200. Only the PUT messaging interface is supported. The DELETE and GET messaging interfaces do not allow XmlEncoding to support the necessary information for the request. Inside the PUT messaging interface, a PcspObj is passed in as the parameter. For the interface to invoke the PcspCiscoBtsCli object, the following must be set in the PcspObj:

- PcspObj.EntityName=PcspCiscoBtsCli (case-insensitive)
- PcspObj.key=<bts.noun>
- PcspObj.XmlEncoding=<bts.cli_xml>

The PcspObj.cmdStatus field is *not* used and is ignored. The key is the noun portion as specified in Cisco BTS 10200 CLI syntax: “<verb> <noun> <parameters>”. The XmlEncoding is the request (or response) and is exactly the same syntax as in the Cisco BTS 10200 generic SOAP/XML and CORBA interface currently available. For further details of this existing syntax, please refer to the programmer’s guide for the CORBA interface or the Generic SOAP/XML.



CHAPTER 5

PCSP Rules and Mapping

Revised: July 2010, OL-23044-01

This chapter describes the PacketCable schema defined for the “Pcsp...” object schema and the required Vendor-specific extensions for full subscriber provisioning execution.

The Extensible Markup Language (XML) syntax example is not displayed in depth here. The full XML schema is given in the [Appendix A, “Cisco BTS 10200 Extensions for Pcsp Objects.”](#)

PcspService

The PcspService object is a base subscriber and a service object for the Cisco BTS 10200 Softswitch. The individual components map as follows:

- ServiceId
 - AdminStatus—This maps the subscriber status and dn2subscriber status into a single field.
0 = SUBSCRIBER.TEMP_UNAVAILABLE or SUBSCRIBER.TEMP_DISCONNECT
1 = SUBSCRIBER.ACTIVE
2 = SUBSCRIBER.TEMP_OOS and SUBSCRIBER.SEASONAL_SUSPEND
 - BillingId—This maps to the subscriber’s BILLING_DN.
 - ExternalId—This maps to the actual subscriber ID.
 - IsPrimary—This value is always set to true. (This would need to change on multiline subscribers.)
 - PrimaryRing—This maps to the subscriber’s RING_TYPE_DN1.
 - Displayname—This maps to the subscriber’s NAME.
 - DisplayNumber—This is the subscriber’s DN1 by default.
 - Password
 - Announcements—This maps to the subscriber’s LANGUAGE and TIMEZONE settings.
 - InterExchange—This maps to the subscriber’s PIC1, PIC2 and PC3 settings.
 - LNP—This maps to the subscriber’s PORTED_IN and PORTED_OUT flags and to true if neither is set.
 - List Of Call Features—This maps to the subscriber’s LNP_TRIGGER flag.

- CfBase—Common feature object base.
 - Subscribed—Each instance of the CfBase extension is derived from the subscriber's service profile and associated features.
 - BillingUsage—This is always set to true.
 - AdminStatus derived from the feature activation, such as STATUS of the subscriber's feature data. TRUE is defined and FALSE is not.
 - CfCND—Base object.
 - CfCIDCW—Base object.
 - CfCW—Base object.
 - CfCFV—Base object.
 - CFCAR—Base object.
 - CfAC—Base object.
 - CfMWI—Simple reflection of stutter dial and message wait indicator from MGW.
 - CfCOT—Base object.
 - CfTWC—Base object.
 - CfRACF—Basic feature and activation status.
 - CfOCCC—Distinctive Ring might be compatible.
 - CfCIES—Enhanced call screening could be done through the Privacy SLE.
 - CfACR—Base object with feature activation status.
 - CfACRestrict—Not supported.
 - CfACB—Not supported.
 - CfCIDB—This makes use of CNDB with Public and Anonymous flags.
 - CfCFBL—Basic object for CFB. This provides an activation flag and the forward to DN derived from SUBSCRIBER_FEATURE_DATA.
 - CfCFDA—Basic object for CFNA. This provides an activation flag Ring Period and the forward to DN derived from SUBSCRIBER_FEATURE_DATA.
 - CfCFC—This is also the basic object for CFU. This provides an activation flag Ring Period and the forward to DN derived from SUBSCRIBER_FEATURE_DATA.
 - CfSCF—This is also the basic object for SCF. This provides an activation flag and a list of forward DN(s) derived from SUBSCRIBER_FEATURE_DATA.
 - CfSCA—This is also the basic object for SCA. This provides an activation flag and a list of forward DN(s) derived from SUBSCRIBER_FEATURE_DATA.
 - CfSCR—This is also the basic object for SCR. This provides an activation flag and a list of forward DN(s) derived from SUBSCRIBER_FEATURE_DATA.
 - CfCRDW—This is also the basic object for SCA. This provides an activation flag and a list of forward DN(s) derived from SUBSCRIBER_FEATURE_DATA. This data be provided by the Application Server (AS) and not stored on the EMS.
 - CfSPCALL—This is the basic object for SC1D and SC2D. It provides the list of numbers but with no limitation or an indication of location. The list of numbers must include additional extensions.

- CfrDA—This is the basic object for the MDN.
BlkDomLongDist—This is derived from the COS_RESTRICT of the subscriber using NATIONAL_RESTRICT_TYPE.
BlkIntLongDist—This is derived from the COS_RESTRICT of the subscriber using BLOCK_INTL.
BlkPayPerView—This is derived from the COS_RESTRICT of the subscriber using BLOCK_900 and BLOCK_976.
BlkOperatorAssist—This is derived from the COS_RESTRICT of the subscriber using BLOCK_NAPA_ASSIST.
BlkDirAssist—This is derived from the COS_RESTRICT of the subscriber using BLOCK_INTL_ASSIST.
BlkTollFree—This could be derived from the COS_RESTRICT. The BlkTolFree object needs a deep dive into the BW list and NOD tables. These are less standard and could vary from SP to SP.
PIN—This is derived from the AUTH_CODE of COS_RESTRICT.
- CfDND—This is generally controlled by application servers. This is not supported in the Cisco BTS 10200.
- CdCOC—This is generally controlled by application servers. This is not supported in the Cisco BTS 10200.
- CfNSA—This basic object is derived from the NSA feature. The ACTIVE indications, StartTod (BEGIN_DOW/TOD) and EndTod (END_DOW/TOD) are all part of the SUBSCRIBER_FEATURE_DATA.

Table 5-1 lists the PacketCable-to-Cisco BTS 10200 translations for PcspService.

Table 5-1 PacketCable Element-to-Cisco BTS 10200 Translations for PcspService

PacketCable Element	Cisco BTS 10200 Translation
ServiceId	Table: subscriber; column: dn1
AdminStatus	<ul style="list-style-type: none"> • 0: suspended (that is, bill not paid)—Table: subscriber; column: status=TEMP_DISCONNECTED. • 1: enabled (normal state)—Table: subscriber; column: status=ACTIVE. • 2: number has changed—Table: changed_number; column: new_dn. AdminStatus is in this state if at least one entry is found. • 3: out of service—Table: subscriber; column: status=TEMP-OOS or TEMP_UNAVAILABLE or SEASONAL_SUSPEND. • 4: unassigned—Table: dn2subscriber; column: sub_id. AdminStatus is in this state if no entries are found.
BillingId	Table: subscriber; column: billing_dn
ExternalId	Table: subscriber; column: id
IsPrimary	Always TRUE
PrimaryRing	Table: subscriber; column: ring_type_dn1
DisplayName	Table: subscriber; column: name
DisplayNumber	Table: subscriber; column: dn1

Table 5-1 PacketCable Element-to-Cisco BTS 10200 Translations for PcspService (continued)

PacketCable Element	Cisco BTS 10200 Translation
Password	No Op
Announcements	—
Language	Table: language; column: id Current supported PC language to Cisco BTS 10200 language: EN=eng; FR=fra; SP=spa; DF=def
Timezone	Table: pop; column: id for GET Cmd No Op on PUT or DELETE Cmds
InterExchange	—
PIC	Table: subscriber; column: pic1
LPIC	Table: subscriber; column: pic2
IPIC	Table: subscriber; column: pic3
LNP	—
PortingStatus	<ul style="list-style-type: none"> • 0: not ported—Table: subscriber; column: ported_in=N • 1: ported in—Table: subscriber; column: ported_in=Y • 2: ported out—No Op; this state is not supported by Cisco BTS 10200
LNPT	Table: subscriber; column: lnpt

PcspMta

The multimedia terminal adapter (MTA) object maps to the media gateway (MGW) table for the Cisco BTS 10200.

- MtaFqdn—This maps to TSAP-ADDR.
- MtaPort—This maps to MGW_PORT.
- CmtsFqdn—This maps to CALL_AGENT_ID.
- MtaProfile—This maps to MGW)PROFILE_ID.
- Timezone—This maps to TIMEZONE of POP table for SUBSCRIBER_PROFILE for RGW or TRUNK_GRP for TGW.
- Protocol—MGCP 1.0 NCS 1.0.
- Codec—This maps to CODEC of QOS table for SUBSCRIBER for RGW or TRUNK_GRP for TGW.
- IPsecControl—True.

Table 5-2 lists the PacketCable elements-to-Cisco BTS 10200 translations for PcspMta.

Table 5-2 PacketCable Elements-to-Cisco BTS 10200 Translations for PcspMta

PacketCable Element	Cisco BTS 10200 Translation
MtaFqdn	Table: mgw; column: tsap_addr
MtaPort	Table: mgw; column: mgw_port
CmsFqdn	Table: call_agent; column: tsap_addr
MtaProfile	Table: mgw; column: mgw_profile_id
Timezone	No Op—Uses CMS default
Protocol	No Op—Uses default of “MGCP 1.0 NCS 1.0”
Codec	Table: qos; column: codec_type
IPSecControl	No Op—Uses default of true

PcspEndPoint

The endpoint object maps to the TERMINATION table for the Cisco BTS 10200.

- EndpointId—This maps to ID column.
- AdminStatus—This maps to STATUS of SUBSCRIBER_TERMINATION/TRUNK_TERMINATION.
- Protocol—MGCP 1.0 NCS 1.0.
- Codec—This maps to CODEC of QOS table for SUBSCRIBER for RGW or TRUNK_GRP for TGW.
- IPSecControl—True.

Table 5-3 lists the PacketCable elements-to-Cisco BTS 10200 translations for PcspEndPoints.

Table 5-3 PacketCable Elements-to-Cisco BTS 10200 Translations for PcspEndPoints

PacketCable Element	Cisco BTS 10200 Translation
EndpointId	Is in the form: <prefix><port>@<mgw-tsap-addr> <ul style="list-style-type: none"> • prefix—Table: termination; column: prefix • port—Table: termination; column: port_start, port_end • mgw-tsap-addr—Table: mgw; column: tsap_addr
AdminStatus	Table: termination; column: status <ul style="list-style-type: none"> • 0: endpoint is disconnected—Table: termination; column: status=OOS or OOS_PENDING • 1: normal—Endpoint is in service. Table: termination; column: status=INS • 2: test mode—Endpoint is under test. Table: termination; column: status=MAINT or MAINT_PENDING
Protocol	No Op—Uses MTA default setting
Codec	No Op—Uses MTA default setting
IPSecControl	No Op—Uses MTA default setting

PcspCms

The call management system (CMS) object maps to the CALL_AGENT table for the Cisco BTS 10200.

- CmsFqdn—This maps to TSAP_ADDR.

Table 5-4 lists the PacketCable elements-to-Cisco BTS 10200 translations for PcspCMS.

Table 5-4 PacketCable Elements-to-Cisco BTS 10200 Translations for PcspCms

PacketCable Element	Cisco BTS 10200 Translation
CmsFqdn	Table: call_agent; column: tsap_addr

Provisioning Rules

This section provides the subscriber and MTA provisioning rules.

Subscriber

This section provides the subscriber provisioning rules.

- On Put commands
 - PcpArg.key must be a valid dn.
 - PcpArg.key must match the PcpService.ServiceId within XML Encoding.
 - PcpArg.key must match the PcpService.displaynum within XML Encoding.
 - PcpService.isPrimary *cannot* be “false”.
 - PcpService.Password is *not* used.
 - PcpService.Announcements.language is mapped as follows in Cisco BTS 10200: EN=eng, SP=spa, FR=fra, DF=def.



Note The Cisco BTS 10200 must have these entries preprovisioned or a failure occurs.

- PcpService.InterExchange.PIC1/2/3 must have corresponding preprovisioned entries in bts.carrier table.
- PcpService.LNP.PortingStatus cannot be 2 (ported out).
- ListOfServicePkg:
 - For Add: all the entries in PcpService.ext:ListOfServicePkg are added to bts.service table for this subscriber.
 - For Modify: all the entries in bts.service table are removed for this subscriber and then all the entries in PcpService.ext:ListOfServicePkg are added to bts.service table for this subscriber.
- On Delete commands
 - PcpArg.key must be a valid DN.
- On Get commands
 - PcpArg.key must be a valid DN.
 - PcpService.Password is *not* used.
- Feature provisioning
 - LSR
 - When BlkTollFree is *false*, then ListOfServicePkg must contain service id that contains Cisco BTS 10200 8XX feature.
 - When BlkTollFree is *true*, then ListOfServicePkg must *not* contain any service that subscriber is subscribed to which contains Cisco BTS 10200 8XX feature.

MTA

This section provides the MTA provisioning rules.

- On Put commands
 - PcpArg.key is mapped to bts.mgw.tsap_addr.
 - PcpArg.key must match PcpMta.mtaFqdn within XML Encoding.
 - PcpMta.cmtsFqdn is mapped to bts.call_agent.tsap_addr and must exist.
 - PcpArg.key is mapped to bts.call_agent.tsap_addr.
 - PcpArg.key must match PcpCms.cmsFqdn within XML Encoding.
 - ListOfServicePkg: For any feature provisioned in the listOfCallFeatures, the extension ListOfServicePkg *must* be completed to include the services related to the feature in the XML listOfCallFeatures. If any feature in the XML listOfCallFeatures is not contained in any of the services, the add or modify fails unless the subscriber ID and service ID are provisioned in the bts.subscriber_service_profile table.
 - PcpEndpoint.EndpointId is non-NULL; if NULL the appropriate PC error is returned.
 - PcpEndpoint.EndpointId is of the form <prefix(string)><N(digits)>@<mgw-tsap-addr(string)>.
 - For ADD, PcpEndpoint.Extension.sub_id is a mandatory extension.
- On Delete commands
 - PcpArg.key is mapped to bts.call_agent.tsap_addr.
 - PcpArg.key is mapped to bts.mgw.tsap_addr.
 - PcpEndpoint.EndpointId is non-NULL; if NULL the appropriate PC error is returned.
 - PcpEndpoint.EndpointId is of the form <prefix(string)><N(digits)>@<mgw-tsap-addr(string)>.
- On Get commands
 - PcpArg.key is mapped to bts.mgw.tsap_addr.
 - CMS.
 - PcpArg.key is mapped to bts.call_agent.tsap_addr.
 - Features.
 - PcpArg.key must be a valid DN.
 - PcpArg.key must match PcpService.ServiceId within XML Encoding.
 - PcpArg.key must match PcpService.displaynum within XML Encoding.
 - Endpoint.
 - For MODIFY, PcpEndpoint.Extension.sub_id is an optional extension, but if it is provided, it improves performance.
 - PcpEndpoint.EndpointId is non-NULL; if NULL return appropriate PC Error.
 - PcpEndpoint.EndpointId is of the form <prefix(string)><N(digits)>@<mgw-tsap-addr(string)>.
 - CiscoBtsCli.

- PespObj.entityName must be equal to PespCiscoBtsCli (case-insensitive).
- PespObj.key must equal the bts.noun and be equivalent to the CLI noun.
- PespObj.cmdStatus is ignored.
- XmlEncoding for Request is exactly the same as the generic Cisco BTS 10200 SOAP/XML and CORBA XML request.
- XmlEncoding for Response is exactly the same as the generic Cisco BTS 10200 SOAP/XML and CORBA XML response.

Endpoint

This section provides the Endpoint provisioning rules.

- On PUT commands.
 - Subscriber must be provisioned before Endpoint is provisioned.

PCSP Extensions Requirements

The following sections provide these PCSP extensions:

- [PespCms Extension](#)
- [PespMta Extension](#)
- [PespService Extension](#)
- [PespEndpoint Extension](#)

PcspCms Extension

Table 5-5 lists the parameter matrix for the PcspCms extensions.

Table 5-5 Parameter Matrix for PcspCms Extensions

Parameter	ADD (PUT cmdStatus=1)	MODIFY (PUT cmdStatus=2)	SHOW (GET)	DELETE (DELETE)
id	Request => Required Response => Mandatory	Request => Optional Response => Optional	—	—
cli	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
mgw_monitoring_enabled	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
host_ip_address	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—
host_id	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—
timestamp	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—
reply_string	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—
Key	Request => Required Response => Mandatory	Request => Required Response => Mandatory	—	—

PcspMta Extension

Table 5-6 lists the parameter matrix for the PcspMta extensions.

Table 5-6 Parameter Matrix PcspMta Extensions

Parameter	ADD (PUT cmdStatus=1)	MODIFY (PUT cmdStatus=2)	SHOW (GET)	DELETE (DELETE)
id	Request => Required Response => Mandatory	Request => Optional Response => Optional	—	—
aggr_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
node	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
type	Request => Required Response => Mandatory	Request => Optional Response => Optional	—	—
host_ip_address	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—
host_id	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—
timestamp	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—
reply_string	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—
key	Request => Required Response => Mandatory	Request => Required Response => Mandatory	—	—

PcspService Extension

Table 5-7 lists the parameter matrix for the PcspService extensions.

Table 5-7 Parameter Matrix PcspService Extensions

Parameter	ADD (PUT cmdStatus=1)	MODIFY (PUT cmdStatus=2)	SHOW (GET)	DELETE (DELETE)
account_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
address1	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
address2	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
aor_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
city	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
cos_restrict_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
country	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
ctxg_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
esrn	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
forced	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
grp	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
h323_term_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
immediate_release	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
mac_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
mgw_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
mlhg_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—

Table 5-7 Parameter Matrix Pcpservice Extensions (continued)

Parameter	ADD (PUT cmdStatus=1)	MODIFY (PUT cmdStatus=2)	SHOW (GET)	DELETE (DELETE)
mlhg_pref_list_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
policy_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
privacy_manager_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
qos_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
sdt_mwi	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
secure_fqdn	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
send_bdn_as_cpn	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
send_bdn_for_emg	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
state	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
static_contact_host	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
static_contact_port	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
static_contact_user	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
sub_profile_id	Request => Required Response =>Mandatory	Request => Optional Response => Optional	—	—
term_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
terminating_immediate_rel	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
tg	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
tgn_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—

Table 5-7 Parameter Matrix Pcservice Extensions (continued)

Parameter	ADD (PUT cmdStatus=1)	MODIFY (PUT cmdStatus=2)	SHOW (GET)	DELETE (DELETE)
usage_sens	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
vmwi	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
voice_mail_id	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
zipcode	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
billing_type	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
category	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
cwt_type	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
offhook_trigger_type	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
ohd_timer	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
privacy	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
term_type	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
user_type	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
ListOfServicePkg	Request => Optional Response => Optional	Request => Optional Response => Optional	—	—
host_ip_address	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—
host_id	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—
timestamp	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—

Table 5-7 Parameter Matrix Pcpservice Extensions (continued)

Parameter	ADD (PUT cmdStatus=1)	MODIFY (PUT cmdStatus=2)	SHOW (GET)	DELETE (DELETE)
reply_string	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—
Key	Request => Required Response => Mandatory	Request => Required Response =>Mandatory	—	—

PcspEndpoint Extension

Table 5-8 lists the parameter matrix for the PcspEndpoint estensions.

Table 5-8 Parameter Matrix Pcpservice Extensions

Parameter	ADD (PUT cmdStatus=1)	MODIFY (PUT cmdStatus=2)	SHOW (GET)	DELETE (DELETE)
sub_id	Request => Required Response => Mandatory	Request => Optional Response => Optional	—	—
host_ip_address	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—
host_id	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—
timestamp	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—
reply_string	Request => Not Used Response => Optional	Request => Not Used Response => Optional	—	—
Key	Request => Required Response =>Mandatory	Request => Required Response =>Mandatory	—	—



CHAPTER 6

PCSP Objects – XML Example (PUT)

Revised: July 2010, OL-23044-01

WSUS

User level security or Web Services User Security (WSUS) is a separate service offering that enables Cisco BTS 10200 Softswitch user security within the confines of PacketCable provisioning.

PcspCms

This section provides an example of PcspCms Extensible Markup Language (XML).

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    Copyright (c) 2005-2007 by Cisco Systems, Inc.
-->
<PcspObj>
  <!--PcspCms is the only service supported by BTS10200-->
  <EntityName>PcspCms</EntityName>
  <!--Subscriber Id-->
  <Key>brandontsapaddr</Key>
  <!--Create new object-->
  <CmdStatus>1</CmdStatus>
  <!--XMLEncoding as per PCSP Data Model Schema or 0 (null)-->
  <XmlEncoding>

  <![CDATA[
    <PcspCms>
      <CmsFqdn>brandontsapaddr</CmsFqdn>
      <Extension xmlns:ext="http://www.cisco.com/CmsExtension">
        <ext:id>CA147</ext:id>
        <ext:cli>12345678900</ext:cli>
        <ext:mw_monitoring_enabled>true</ext:mw_monitoring_enabled>
      </Extension>
    </PcspCms>
  ]]>
  </XmlEncoding>
</PcspObj>
```

PcspMta

This section provides an example of PcspMta XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    Copyright (c) 2005-2007 by Cisco Systems, Inc.
-->
<PcspObj>
  <!--PcspMta is the only service supported by BTS10200-->
  <EntityName>PcspMta</EntityName>
  <!--Subscriber Id-->
  <Key>brandont1</Key>
  <!--Create new object-->
  <CmdStatus>1</CmdStatus>
  <!--XMLEncoding as per PCSP Data Model Schema or 0 (null)-->
  <XmlEncoding>

  <![CDATA[
    <PcspMta>
      <MtaFqdn>brandont1</MtaFqdn>
      <ListenPort>2427</ListenPort>
      <CmtsFqdn>sim-SYS14CA146.ipclab.cisco.com:9146</CmtsFqdn>
      <MtaProfile>c5350-59-140</MtaProfile>
      <Timezone>-0500</Timezone>
      <Protocol>MCGP 1.0 NCS 1.0</Protocol>
      <Codec>1</Codec>
      <IPSecControl>>true</IPSecControl>
      <Extension xmlns:ext="http://www.cisco.com/MtaExtension">
        <ext:id>bdmta</ext:id>
        <ext:aggr_id>c7246-227-77</ext:aggr_id>
        <ext:node>test</ext:node>
        <ext:type>RGW</ext:type>
      </Extension>
    </PcspMta>
  ]]>

  </XmlEncoding>
</PcspObj>
```

PcspService

This section provides an example of PcspService XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    Copyright (c) 2005-2007 by Cisco Systems, Inc.
-->
<PcspObj>
  <!--PcspService is the only service supported by BTS10200-->
  <EntityName>PcspService</EntityName>
  <!--Subscriber Id-->
  <Key>2146255555</Key>
  <!--Create new object-->
  <CmdStatus>1</CmdStatus>
  <!--XMLEncoding as per PCSP Data Model Schema or 0 (null)-->
  <XmlEncoding>

  <![CDATA[
    <PcspService xmlns="http://www.cablelabs.com/Pcsp/I01/schema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:pcsp="http://www.cablelabs.com/Pcsp/I01/schema">
      <ServiceId format="NSN">2146255555</ServiceId>
      <AdminStatus>1</AdminStatus>
      <BillingId>2146255555</BillingId>
      <ExternalId>brandontest</ExternalId>
      <IsPrimary>true</IsPrimary>
      <PrimaryRing>R1</PrimaryRing>
      <DisplayName>John Q Public</DisplayName>
      <DisplayNumber>2146255555</DisplayNumber>
      <Password>45hjg3j6gkg6h54j6gkj3g6</Password>
      <Announcements>
        <Language>EN</Language>
        <Timezone>+0500</Timezone>
      </Announcements>
      <InterExchange>
        <PIC>0123</PIC>
        <LPIC>0123</LPIC>
        <IPIC>0123</IPIC>
      </InterExchange>
      <LNP>
        <PortingStatus>0</PortingStatus>
        <LNPT>0</LNPT>
      </LNP>
      <ListOfCallFeatures>
        <CfAC>
          <Subscribed>true</Subscribed>
          <AdminStatus>1</AdminStatus>
        </CfAC>
        <CfCFBL>
          <Subscribed>true</Subscribed>
          <AdminStatus>1</AdminStatus>
          <UserParams>
            <Active>true</Active>
            <Always>0</Always>
            <ForwardTo>9785551212</ForwardTo>
          </UserParams>
        </CfCFBL>
      </ListOfCallFeatures>

      <Extension xmlns:ext="http://www.cisco.com/ServiceExtension">
        <ext:account_id>test</ext:account_id>
        <ext:address1>test</ext:address1>
      </Extension>
    </CDATA!>
  </XmlEncoding>
</PcspObj>
```

```

    <ext:address2>test</ext:address2>
    <ext:city>test</ext:city>
    <ext:country>test</ext:country>
    <ext:esrn>2345</ext:esrn>
    <ext:forced>true</ext:forced>
    <ext:grp>true</ext:grp>
    <ext:immediate_release>true</ext:immediate_release>
    <ext:sdt_mwi>true</ext:sdt_mwi>
    <ext:send_bdn_as_cpn>true</ext:send_bdn_as_cpn>
    <ext:send_bdn_for_emg>true</ext:send_bdn_for_emg>
    <ext:state>test</ext:state>
    <ext:sub_profile_id>tb14-cable</ext:sub_profile_id>
    <ext:terminating_immediate_rel>test</ext:terminating_
immediate_rel>
    <ext:usage_sens>true</ext:usage_sens>
    <ext:vmwi>true</ext:vmwi>
    <ext:zipcode>75082</ext:zipcode>
    <ext:billing_type>FR1</ext:billing_type>
    <ext:category>INDIVIDUAL</ext:category>
    <ext:cwt_type>WT1</ext:cwt_type>
    <ext:offhook_trigger_type>NONE</ext:offhook_trigger_type>
    <ext:ohd_timer>0</ext:ohd_timer>
    <ext:privacy>FULL</ext:privacy>
    <ext:term_type>NONE</ext:term_type>
    <ext:ListOfServicePkg>
      <ext:pkgid>bus-gold</ext:pkgid>
      <ext:pkgid>ivr</ext:pkgid>
    </ext:ListOfServicePkg>
  </Extension>
</PcspService>
]]>
</XmlEncoding>
</PcspObj>

```

PcspEndpoint

This section provides an example of PcspEndpoint XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    Copyright (c) 2005-2007 by Cisco Systems, Inc.
-->
<PcspObj>
  <!--PcspEndpoint is the only service supported by BTS10200-->
  <EntityName>PcspEndpoint</EntityName>
  <!--Subscriber Id-->
  <Key>DS/S-0/DS1-24/12@c8880-227-01.ipclab.cisco.com</Key>
  <!--Create new object-->
  <CmdStatus>1</CmdStatus>
  <!--XMLEncoding as per PCSP Data Model Schema or 0 (null)-->
  <XmlEncoding>

  <![CDATA[
    <PcspEndpoint>
      <EndpointId>DS/S-0/DS1-24/12@c8880-227-01.ipclab.cisco.com</EndpointId>
      <AdminStatus>2</AdminStatus>
      <Protocol>MtaDefault</Protocol>
      <Codec>2</Codec>
      <IPSecControl>true</IPSecControl>
      <Extension xmlns:ext="http://www.cisco.com/EndpointExtension">
        <sub_id>215-224-2234</sub_id>
      </Extension>
    </PcspEndpoint>
  ]]>

  </XmlEncoding>
</PcspObj>
```

PcspCiscoBtsCli

This section provides two examples of PcspCiscoBtsCli XML.

Request Message, Example 1

This section provides the request message example 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    Copyright (c) 2005-2007 by Cisco Systems, Inc.
-->
<PcspObj>
  <!--PcspCiscoBtsCli service supported by BTS10200-->
  <EntityName>PcspCiscoBtsCli</EntityName>
  <!--Subscriber Id-->
  <Key>dn2subscriber</Key>
  <!--CmdStatus is IGNORED-->
  <CmdStatus>1</CmdStatus>
  <!--XMLEncoding as per PCSP Data Model Schema or 0 (null)-->
  <XmlEncoding>

  <![CDATA[
    <CiscoBtsCliObj>
      <Request Verb='show' Noun='dn2subscriber'>
        <Entry Key='limit' Value='2'></Entry>
      </Request>
      <Extension xmlns:ext="http://www.cisco.com/BtsCliExtension">
        <ext:key>535480450</ext:key>
      </Extension>
    </CiscoBtsCliObj>
  ]]>
  </XmlEncoding>
</PcspObj>
```

Response Message, Example 1

This section provides the response message example 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Copyright (c) 2005-2007 by Cisco Systems, Inc.
-->
<PcspObj>
  <EntityName>PcspCiscoBtsCli</EntityName>
  <Key>dn2subscriber</Key>
  <CmdStatus>1</CmdStatus>
  <XmlEncoding>
    <CiscoBtsCliObj>
      <Reply id='Reply'>
        <Status>true</Status>
        <Reason>Success: at 2007-11-08 07:20:01 by btsapp Entries 1-2 of 169
        returned.</Reason>
        <Size>2</Size>
        <AbsoluteSize>169</AbsoluteSize>
        <StartRow>1</StartRow>
        <DataTable>
          <Row id='0'>
            <Column id='office_code_index'>2</Column>
            <Column id='dn'>0101</Column>
            <Column id='status'>ASSIGNED</Column>
            <Column id='ring_type'>R1</Column>
            <Column id='lnp_trigger'>N</Column>
            <Column id='np_reserved'>N</Column>
            <Column id='annc_id'></Column>
            <Column id='sub_id'>201-222-0101</Column>
            <Column id='virtual_dn'>N</Column>
            <Column id='cwt_type'>WT1</Column>
            <Column id='admin_dn'>N</Column>
            <Column id='ported_in'>N</Column>
            <Column id='LAST_CHANGED'>2007-10-26 08:27:58</Column>
          </Row>
          <Row id='1'>
            <Column id='office_code_index'>2</Column>
            <Column id='dn'>0102</Column>
            <Column id='status'>ASSIGNED</Column>
            <Column id='ring_type'>R1</Column>
            <Column id='lnp_trigger'>N</Column>
            <Column id='np_reserved'>N</Column>
            <Column id='annc_id'></Column>
            <Column id='sub_id'>201-222-0102</Column>
            <Column id='virtual_dn'>N</Column>
            <Column id='cwt_type'>WT1</Column>
            <Column id='admin_dn'>N</Column>
            <Column id='ported_in'>N</Column>
            <Column id='LAST_CHANGED'>2007-10-26 08:27:58</Column>
          </Row>
        </DataTable>
      </Reply>
      <Extension xmlns:ext="http://www.cisco.com/BtsCliExtension">
        <ext:key>1444850966</ext:key>
        <ext:host_ip_address>10.89.224.178</ext:host_ip_address>
        <ext:host_id>priems16</ext:host_id>
        <ext:timestamp>2008-01-16 11:11:26</ext:timestamp>
      </Extension>
    </CiscoBtsCliObj>
  </XmlEncoding>
</PcspObj>
```

Request Message, Example 2

This section provides the request message example 2.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    Copyright (c) 2005-2007 by Cisco Systems, Inc.
-->
<PcspObj>
  <!--PcspCiscoBtsCli service supported by BTS10200-->
  <EntityName>PcspCiscoBtsCli</EntityName>
  <!--Subscriber Id-->
  <Key>trunk_termination</Key>
  <!--CmdStatus is IGNORED-->
  <CmdStatus>1</CmdStatus>
  <!--XMLEncoding as per PCSP Data Model Schema or 0 (null)-->
  <XmlEncoding>

  <![CDATA[
    <CiscoBtsCliObj>
      <Request Verb='status' Noun='trunk_termination'>
        <Entry Key='cic' Value='8'></Entry>
        <Entry Key='tgn_id' Value='30076'></Entry>
      </Request>
    </CiscoBtsCliObj>
  ]]>
  </XmlEncoding>
</PcspObj>
```

Response Message, Example 2

This section provides the response message example 2.

```
<PcspObj>
  <EntityName>PcspCiscoBtsCli</EntityName>
  <Key>dn2subscriber</Key>
  <CmdStatus>1</CmdStatus>
  <XmlEncoding>
    <CiscoBtsCliObj>
      <Reply id='Reply'>
        <Status>true</Status>
        <Reason>Success: at 2007-11-08 07:26:11 by btsapp Entry 1 of 1
        returned.</Reason>
        <Size>1</Size>
        <AbsoluteSize>1</AbsoluteSize>
        <StartRow>1</StartRow>
        <DataTable>
          <Row id='0'>
            <Column id='tgn_id'>30076</Column>
            <Column id='cic'>8</Column>
            <Column id='RESULT'>ADM configure result in success</Column>
            <Column id='REASON'>ADM executed successfully</Column>
            <Column id='TERM_ADMIN_STATE'>ADMIN_INS</Column>
            <Column id='TERM_OPER_STATE'>Term is available for new calls
            </Column>
            <Column id='TERM_REASON'>No fault reason available</Column>
            <Column id='TRUNK_STATIC_STATE'>ACTV</Column>
            <Column id='TRUNK_DYNAMIC_STATE'>IDLE</Column>
            <Column id='TRUNK_REASON'>NON_FAULTY</Column>
          </Row>
        </DataTable>
      </Reply>
      <Extension xmlns:ext="http://www.cisco.com/BtsCliExtension">
        <ext:key>1444850966</ext:key>
        <ext:host_ip_address>10.89.224.178</ext:host_ip_address>
        <ext:host_id>priems16</ext:host_id>
        <ext:timestamp>2008-01-16 11:20:34</ext:timestamp>
      </Extension>
    </CiscoBtsCliObj>
  </XmlEncoding>
</PcspObj>
```




CHAPTER 7

Performance Considerations

Revised: July 2010, OL-23044-01

Performance

Table 7-1 indicates some rules to consider when building PacketCable Subscriber Provisioning (PCSP) Simple Object Access Protocol (SOAP)/Extensible Markup Language (XML) commands. These rules can improve the overall performance and execution time for the given commands.

Table 7-1 Considerations When Building PCSP SOAP/XML Commands

Entity	ADD (PUT cmdStatus=1)	MODIFY (PUT cmdStatus=2)	SHOW (GET)	DELETE (DELETE)
PcspCms				
PcspMta				
PcspService				
PcspEndpoint		Specify sub_id in extension to improve performance		
PcspCisoBtsCli				



CHAPTER 8

PCSP Subscriber Provisioning

Revised: July 2010, OL-23044-01

Provisioning a Subscriber with the Minimum Number of Features

From the PCSP interface, use the following instructions to provision a subscriber with the minimum number of features and extensions and place the subscriber and features and extensions in service.

While the following steps are executed, the PSCP adapter

- Changes termination.id to subscriber-id
- Changes subscriber.mgw-id to termination.mgw-id
- Changes subscriber.term-id to termination.id
- Changes subscriber.term-type to none
- Equips or unequips and places the subscriber-termination in the requested AdminStatus state

Step 1 Add a new MTA (bts: mgw).

```
"put mta"
```

```
CmdStatus=1
```

Example:

```

<PcspMta>
  <MtaFqdn>brandont1</MtaFqdn>
  <ListenPort>2427</ListenPort>
  <CmtsFqdn>sim-SYS14CA146.ipclab.cisco.com:9146</CmtsFqdn>
  <MtaProfile>c5350-59-140</MtaProfile>
  <Timezone>-0500</Timezone>
  <Protocol>MCGP 1.0 NCS 1.0</Protocol>
  <Codec>1</Codec>
  <IPSecControl>>true</IPSecControl>
  <Extension xmlns:ext="http://www.cisco.com/CmsExtension">
    <ext:id>bdmta</ext:id>
    <ext:aggr_id>c7246-227-77</ext:aggr_id>
    <ext:node>test</ext:node>
    <ext:type>RGW</ext:type>
  </Extension>
</PcspMta>

```

Step 2 Place the MTA in-service.

```
"control mta" in-service
```

Example:

```

<CiscoBtsCliObj>
  <Request Verb='control' Noun='mgw'>
    <Entry Key='id' Value='brandont1'></Entry>
    <Entry Key='target_state' Value='INS'></Entry>
    <Entry Key='Wait' Value='y'></Entry>
    <Entry Key='mode' Value='FORCED'></Entry>
  </Request>
  <Extension xmlns:ext="http://www.cisco.com/BtsCliExtension">
    <ext:key>1266645536</ext:key>
  </Extension>
</CiscoBtsCliObj>

```

Step 3 Add a new service (bts: subscriber) with term-type=NONE.

```
"put subscriber"
```

```
CmdStatus=1
```

```
ext:term_type=none
```

Example:

```
<PcspService xmlns="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:pcsp="http://www.cablelabs.com/Pcsp/I01/schema">
  <ServiceId format="NSN">2146255555</ServiceId>
  <AdminStatus>1</AdminStatus>
  <BillingId>2146255555</BillingId>
  <ExternalId>brandontest</ExternalId>
  <IsPrimary>true</IsPrimary>
  <PrimaryRing>R1</PrimaryRing>
  <DisplayName>John Q Public</DisplayName>
  <DisplayNumber>2146255555</DisplayNumber>
  <Password>45hjg3j6gkg6h54j6gkj3g6</Password>
  <Announcements>
    <Language>EN</Language>
    <Timezone>+0500</Timezone>
  </Announcements>
  <InterExchange>
    <PIC>0123</PIC>
    <LPIC>0123</LPIC>
    <IPIC>0123</IPIC>
  </InterExchange>
  <LNP>
    <PortingStatus>0</PortingStatus>
    <LNPT>0</LNPT>
  </LNP>
  <ListOfCallFeatures>
    <CfAC>
      <Subscribed>true</Subscribed>
      <AdminStatus>1</AdminStatus>
    </CfAC>
  </ListOfCallFeatures>
  <Extension xmlns:ext="http://www.cisco.com/CmsExtension">
    <ext:sub_profile_id>tb14-cable</ext:sub_profile_id>
    <ext:term_type>NONE</ext:term_type>
    <ext:ListOfServicePkg>
      <ext:pkgid>bus-gold</ext:pkgid>
    </ext:ListOfServicePkg>
  </Extension>
</PcspService>
```

Step 4 Add a new endpoint (bts: termination) and specify the subscriber-id that was added in Step 3.

```
"put endpoint"
```

```
CmdStatus=1
```

```
ext:sub_id=<newly_created_subscriber_above>
```

```
AdminStatus=1 (in-service)
```

Example:

```
<PcspEndpoint>
  <EndpointId>brandon1@bdmta</EndpointId>
  <AdminStatus>0</AdminStatus>
  <Extension>
    <ext:sub_id>brandontest</ext:sub_id>
    <ext:key>1337593914</ext:key>
  </Extension>
</PcspEndpoint>
```

Provisioning a Subscriber with a Feature

From the PCSP interface, use the following instructions to provision a subscriber with a feature and place the subscriber and feature in-service.

While the following steps are executed, the PSCP adapter

- Changes termination.id to subscriber-id
- Changes subscriber.mgw-id to termination.mgw-id
- Changes subscriber.term-id to termination.id
- Changes subscriber.term-type to none
- Places the subscriber-termination in the requested AdminStatus state

Step 1 Add a new MTA (bts: mgw).

```
"put mta"
```

```
CmdStatus=1
```

Example:

```
<PcspMta>
  <MtaFqdn>brandont1</MtaFqdn>
  <ListenPort>2427</ListenPort>
  <CmtsFqdn>sim-SYS14CA146.ipclab.cisco.com:9146</CmtsFqdn>
  <MtaProfile>c5350-59-140</MtaProfile>
  <Timezone>-0500</Timezone>
  <Protocol>MCGP 1.0 NCS 1.0</Protocol>
  <Codec>1</Codec>
  <IPSecControl>>true</IPSecControl>
  <Extension xmlns:ext="http://www.cisco.com/CmsExtension">
    <ext:id>bdmta</ext:id>
    <ext:aggr_id>c7246-227-77</ext:aggr_id>
    <ext:node>test</ext:node>
    <ext:type>RGW</ext:type>
  </Extension>
</PcspMta>
```

Step 2 Place the MTA in-service.

```
"control mta" in-service
```

Example:

```
<CiscoBtsCliObj>
  <Request Verb='control' Noun='mgw'>
    <Entry Key='id' Value='brandont1'></Entry>
    <Entry Key='target_state' Value='INS'></Entry>
    <Entry Key='Wait' Value='y'></Entry>
    <Entry Key='mode' Value='FORCED'></Entry>
  </Request>
  <Extension xmlns:ext="http://www.cisco.com/BtsCliExtension">
    <ext:key>1266645536</ext:key>
  </Extension>
</CiscoBtsCliObj>
```

Step 3 Add a new service (bts: subscriber) with term-type=NONE.

```
"put subscriber"
```

```
CmdStatus=1
```

```
ext:term_type=none
```

Example:

```
<PcspService xmlns="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:pcsp="http://www.cablelabs.com/Pcsp/I01/schema">
  <ServiceId format="NSN">2146255555</ServiceId>
  <AdminStatus>1</AdminStatus>
  <BillingId>2146255555</BillingId>
  <ExternalId>brandontest</ExternalId>
  <IsPrimary>true</IsPrimary>
  <PrimaryRing>R1</PrimaryRing>
  <DisplayName>John Q Public</DisplayName>
  <DisplayNumber>2146255555</DisplayNumber>
  <Password>45hjg3j6gkg6h54j6gkj3g6</Password>
  <Announcements>
    <Language>EN</Language>
    <Timezone>+0500</Timezone>
  </Announcements>
  <InterExchange>
    <PIC>0123</PIC>
    <LPIC>0123</LPIC>
    <IPIC>0123</IPIC>
  </InterExchange>
  <LNP>
    <PortingStatus>0</PortingStatus>
    <LNPT>0</LNPT>
  </LNP>
  <ListOfCallFeatures>
    <CfAC>
      <Subscribed>true</Subscribed>
      <AdminStatus>1</AdminStatus>
    </CfAC>
    <CfCFBL>
      <Subscribed>true</Subscribed>
      <AdminStatus>1</AdminStatus>
      <UserParams>
        <Active>true</Active>
        <Always>0</Always>
        <ForwardTo>9785551212</ForwardTo>
      </UserParams>
    </CfCFBL>
  </ListOfCallFeatures>
  <Extension xmlns:ext="http://www.cisco.com/CmsExtension">
    <ext:account_id>test</ext:account_id>
    <ext:address1>test</ext:address1>
    <ext:address2>test</ext:address2>
    <ext:city>test</ext:city>
    <ext:country>test</ext:country>
    <ext:esrn>2345</ext:esrn>
    <ext:forced>true</ext:forced>
    <ext:grp>true</ext:grp>
    <ext:immediate_release>true</ext:immediate_release>
    <ext:sdt_mwi>true</ext:sdt_mwi>
    <ext:send_bdn_as_cpn>true</ext:send_bdn_as_cpn>
    <ext:send_bdn_for_emg>true</ext:send_bdn_for_emg>
    <ext:state>test</ext:state>
```

```

    <ext:sub_profile_id>tb14-cable</ext:sub_profile_id>
    <ext:terminating_immediate_rel>test</ext:terminating_immediate_rel>
    <ext:usage_sens>true</ext:usage_sens>
    <ext:vmwi>true</ext:vmwi>
    <ext:zipcode>75082</ext:zipcode>
    <ext:billing_type>FR1</ext:billing_type>
    <ext:category>INDIVIDUAL</ext:category>
    <ext:cwt_type>WT1</ext:cwt_type>
    <ext:offhook_trigger_type>NONE</ext:offhook_trigger_type>
    <ext:ohd_timer>0</ext:ohd_timer>
    <ext:privacy>FULL</ext:privacy>
    <ext:term_type>NONE</ext:term_type>
    <ext:ListOfServicePkg>
    <ext:pkgid>bus-gold</ext:pkgid>
    <ext:pkgid>ivr</ext:pkgid>
    </ext:ListOfServicePkg>
  </Extension>
</PcspService>

```

Step 4 Add a new endpoint (bts: termination) and specify the subscriber-id that was added in Step 3.

"put endpoint"

CmdStatus=1

ext:sub_id=<newly_created_subscriber_above>

AdminStatus=1 (in-service)

Example:

```

<PcspEndpoint>
  <EndpointId>brandon1@bdmta</EndpointId>
  <AdminStatus>0</AdminStatus>
  <Extension>
    <ext:sub_id>brandon1</ext:sub_id>
    <ext:key>1337593914</ext:key>
  </Extension>
</PcspEndpoint>

```

Step 5 Set the MTA to command status 1.

"put mta; CmdStatus=1"

Modifying a Subscriber Feature

To modify existing features of a current subscriber, use the following instructions:

Execute a **put service** command for the feature modifications.

"put service"

CmdStatus=2

Example:

```
<PcspService xmlns="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:pcsp="http://www.cablelabs.com/Pcsp/I01/schema">
  <ServiceId format="NSN">2146255555</ServiceId>
  <AdminStatus>1</AdminStatus>
  <BillingId>2146255555</BillingId>
  <ExternalId>brandontest</ExternalId>
  <IsPrimary>true</IsPrimary>
  <PrimaryRing>R1</PrimaryRing>
  <DisplayName>John Q Public</DisplayName>
  <DisplayNumber>2146255555</DisplayNumber>
  <Password>45hbj3j6gkg6h54j6gkj3g6</Password>
  <Announcements>
    <Language>EN</Language>
    <Timezone>+0500</Timezone>
  </Announcements>
  <InterExchange>
    <PIC>0123</PIC>
    <LPIC>0123</LPIC>
    <IPIC>0123</IPIC>
  </InterExchange>
  <LNP>
    <PortingStatus>0</PortingStatus>
    <LNPT>0</LNPT>
  </LNP>
  <ListOfCallFeatures>
    <CfSPCALL>
      <Subscribed>true</Subscribed>
      <AdminStatus>1</AdminStatus>
      <UserParams>
        <ListOfSpeedDial>
          <SdPair>
            <SdNum>2</SdNum>
            <ServiceId>2762223900</ServiceId>
          </SdPair>
          <SdPair>
            <SdNum>3</SdNum>
            <ServiceId>2762223901</ServiceId>
          </SdPair>
        </ListOfSpeedDial>
      </UserParams>
    </CfSPCALL>
    <CfCND>
      <Subscribed>true</Subscribed>
      <AdminStatus>1</AdminStatus>
    </CfCND>
    <CfACR>
      <Subscribed>true</Subscribed>
      <AdminStatus>1</AdminStatus>
      <UserParams>
        <Active>true</Active>
      </UserParams>
    </CfACR>
  </ListOfCallFeatures>
</PcspService>
```

```
        </UserParams>
    </CfACR>
</ListOfCallFeatures>
<Extension xmlns:ext="http://www.cisco.com/CmsExtension">
    <ext:ListOfServicePkg>
        <ext:pkgid>bus-gold</ext:pkgid>
        <ext:pkgid>ivr</ext:pkgid>
    </ext:ListOfServicePkg>
</Extension>
</PcspService>
```

Adding Features to a Subscriber

To add features to an existing subscriber, use the [“Modifying a Subscriber Feature”](#) section on page 8-7.



CHAPTER 9

Troubleshooting

Revised: July 2010, OL-23044-01

This section describes the use of various tools on the client side for determining the state of operations on the Simple Object Access Protocol (SOAP)/Extensible Markup Language (XML) interface of the Cisco BTS 10200 Softswitch. This includes the use of the BTSXSDK software development kit (SDK) package tools.

Full coverage of the SDK is provided in the *Cisco BTS 10200 Softswitch SOAP Adapter Interface Specification Programmer's Guide*.

Tcpmon Tool

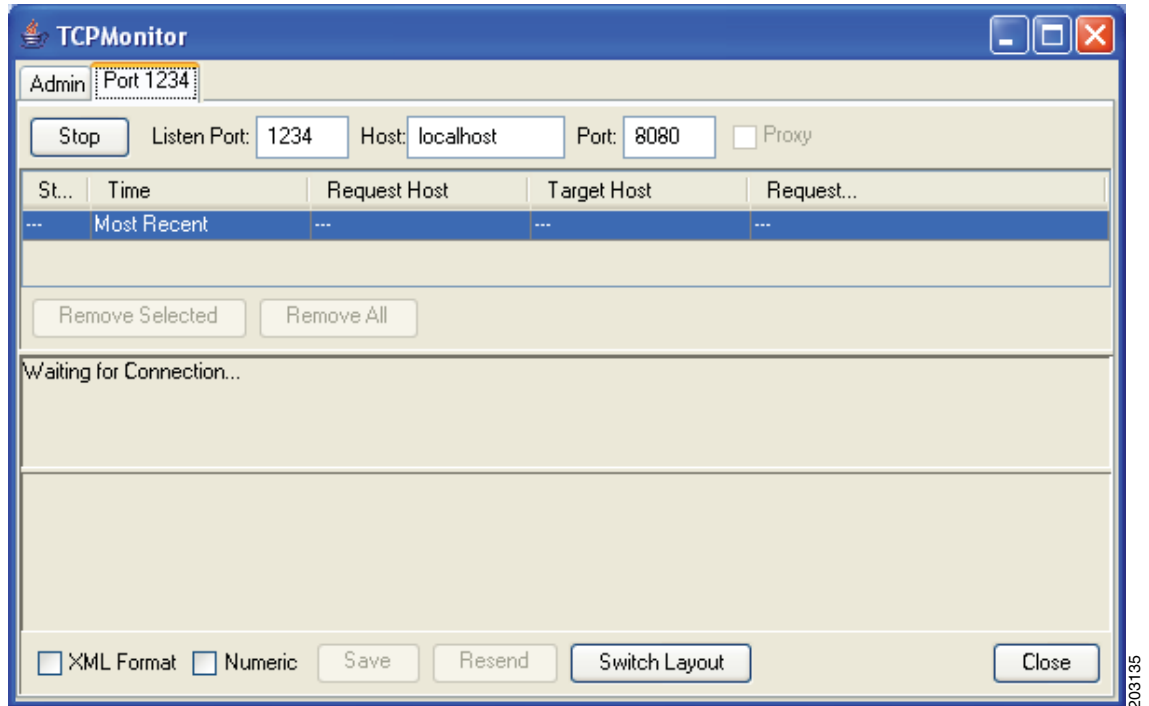
The tcpmon utility is found in the org.apache.axis.utils package and is used to test SOAP requests and responses. The tcpmon tool is invoked from the command line. (See [Figure 9-1](#).)

Figure 9-1 *Launch of Tcpmon Tool*



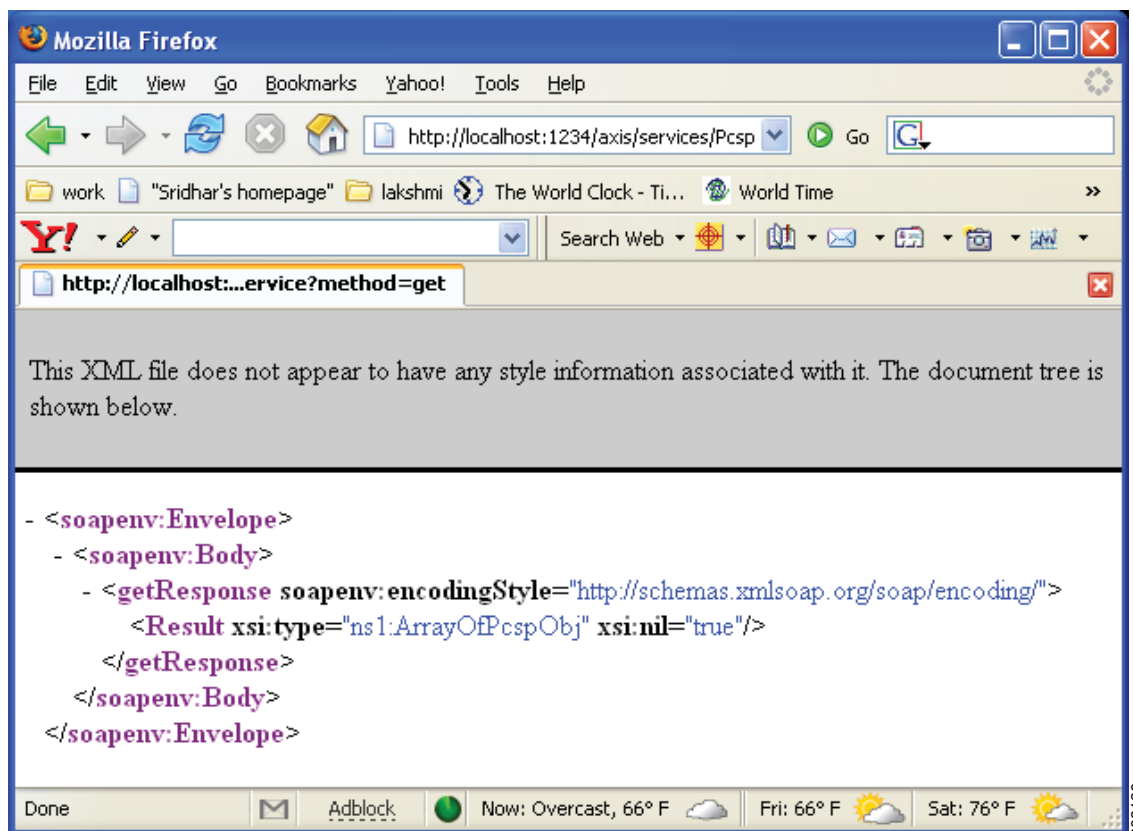
The graphical user interface (GUI) shown in [Figure 9-2](#) is displayed.

Figure 9-2 Basic Tcpmon Screen



Now each time a SOAP connection is made to the local port. The user sees the request in the Request panel and the server response in the Response panel (see [Figure 9-3](#)).

Figure 9-3 Request and Response



SDK Tool

The SDK provides a Solaris- or Linux-based Java command line implementation. This package contains sample code as working examples for utilizing the SOAP/XML interface. For Solaris, this package is automatically installed on each Element Management System (EMS) and can also be installed on a separate Solaris box by use of the package add (**pkgadd**) command. Only a few basic items are required to get this tool up and running.

To enable the SDK environment, use the following commands examples.

```

# pkgadd -d . BTSxsdk <ENTER>
# cd /opt/BTSxsdk
# export PROJECTDIR=`pwd`
# export JAVA_HOME=/usr/java
# ./bin/make all
# cd bin
# ./bts-secure-cli 192.168.2.120 -t SOAP -n btsadmin -p btsadmin

```

Common Security Certificates

Each Cisco BTS 10200 should utilize a common Apache secure sockets layer (SSL) key file and certificate. These are included in the SMCapache package installation for Cisco BTS 10200 5.0 MR2 and above. However, on loads prior to this, the key and certificate must be applied by use of the BTS_SOAP_Common_Certificate Method of Procedure (MOP). This MOP is not covered in detail here.

Verification for the presence of the common certificate can be done with the following steps. Without the common certificates, the client side can receive application authentication failures for SSL.

- Apache Configuration:

```
# grep -i SSLCert /opt/SMCapache/conf/httpd.conf
- SSLCertificateFile /opt/SMCapache/conf/ssl.crt/bts10200.crt
- SSLCertificateKeyFile /opt/SMCapache/conf/ssl.crt/bts10200.key
```

- Key File store Location:

```
# ls /opt/SMCapache/conf/ssl.crt/bts10200.key
# ls /opt/SMCapache/conf/ssl.crt/bts10200.crt
```

- Certificate Verification:

```
# /opt/BTSoss1/bin/openssl x509 -noout -text -in bts10200.crt
```

Certificate:

Data:

```
Version: 1 (0x0)
Serial Number: 1079728370 (0x405b58f2)
Signature Algorithm: md5WithRSAEncryption
Issuer: C=US, ST=Texas, L=Richardson, O=Cisco Systems Inc., OU=IPCBU,
CN=BTS 10200
Validity
  Not Before: Mar 19 20:32:50 2004 GMT
  Not After : Dec 28 20:32:50 2011 GMT
Subject: C=US, ST=Texas, L=Richardson, O=Cisco Systems Inc., OU=IPCBU,
CN=BTS 10200
Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
  RSA Public Key: (1024 bit)
    Modulus (1024 bit):
      00:c7:61:c1:45:51:03:cc:db:1b:06:9d:7b:cf:c4:
      b1:6a:55:a8:c0:a7:fd:ef:25:aa:c3:02:87:66:fc:
      76:77:b3:c1:3f:74:23:94:8d:5e:ce:60:5c:e6:5b:
      02:51:52:31:9c:c6:da:cc:03:24:ab:92:a8:87:f1:
      f7:b8:7a:c6:90:3f:7d:7f:3e:81:23:a1:32:1d:12:
      bf:36:4f:94:67:34:98:4d:44:55:e7:1f:5f:8d:23:
      9e:dd:9a:d8:5b:29:b4:20:86:36:17:80:22:92:90:
      13:8c:29:c9:c1:9a:dd:1e:8f:10:cb:35:4c:49:c3:
      51:87:47:89:b4:8f:83:41:5b
    Exponent: 65537 (0x10001)
Signature Algorithm: md5WithRSAEncryption
40:ca:3d:68:cc:4d:6c:4e:e9:67:61:84:04:80:43:c9:6b:9d:
51:2e:a7:61:97:0d:4b:75:74:8f:6e:75:cf:30:fd:f2:4d:e0:
cd:94:89:84:6b:f5:f7:b0:de:1b:6d:5f:31:97:af:8f:73:7d:
f4:ca:a5:cf:e8:5f:ee:b6:83:15:0f:45:bc:bc:32:6f:e6:dc:
4b:cf:b4:87:96:aa:6e:c4:ec:dd:1f:be:aa:b1:48:45:44:4f:
30:67:d7:be:ce:cd:d4:8e:7a:1f:17:68:ee:c6:52:ef:b7:e6:
bf:46:ca:bb:20:da:e4:ef:09:a9:d8:90:e1:df:0b:db:90:1e:
ee:85
```



APPENDIX A

Cisco BTS 10200 Extensions for Pcps Objects

Revised: July 2010, OL-23044-01

This appendix provides the Cisco BTS 10200 Softswitch extensions for the Pcps objects. Use the [Cisco BTS 10200 Softswitch CLI Database](#) to specify what each field denotes in the comment section.

PcpsCms

This section provides the extensions for the PcpsCms objects.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
- Copyright (c) 2007 by Cisco Systems, Inc.
- All rights reserved.

id - Primary key. Call Agent ID (domain name). Service provider assigns, based on
network configuration.
    VARCHAR(8): 1.8 ASCII characters. The form of the id is CAnnn or cannn
    where nnn = 000 to 999.

clli - Foreign key: CLLI Code table. The Common Language Location Identifier
(CLLI) for the Call Agent.
    CHAR(11): 11 ASCII characters.

mgw_monitoring_enabled - Media Gateway monitoring enabled/disabled indicator.
    CHAR(1): Y/N (Default = Y).

host_ip_address - IP Address of the EMS machine servicing the response. Is only
applicable in response object.

host_id - configured EMS hostname of the EMS machine servicing the response. Is
only applicable in response object.

timestamp - Timestamp the reply from BTS EMS is done with processing the request.
Is only applicable in response object.

reply_string - Reply string as seen from BTS EMS notifying success or failure and
details of failure if available. Is only applicable in response object.

key - Security Key of the user session.

-->
<xs:schema targetNamespace="http://www.cisco.com/CmsExtension"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```

xmlns="http://www.cisco.com/CmsExtension"
elementFormDefault="qualified">
  <xs:element name="id" type="xs:string"/>
  <xs:element name="cli" type="xs:string"/>
  <xs:element name="mgw_monitoring_enabled" type="xs:boolean"/>
  <xs:element name="host_ip_address" type="xs:string"/>
  <xs:element name="host_id" type="xs:string"/>
  <xs:element name="timestamp" type="xs:string"/>
  <xs:element name="reply_string" type="xs:string"/>
  <xs:element name="key" type="xs:string"/>
</xs:schema>

```

PcspMta

This section provides the extensions for the PcspMta objects.

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
- Copyright (c) 2007 by Cisco Systems, Inc.
- All rights reserved.
  id - Primary key. Media gateway identifier, assigned by the service provider.
      VARCHAR(32): 1-32 ASCII characters.

  type - Type of gateway.
      VARCHAR(3): 1-3 ASCII characters. Permitted values are:
          RGW-Residential gateway.
          TGW-Trunking gateway.

  aggr_id - ID of the aggregation device cable modem termination system (CMTS). This
  token is mandatory if supporting PacketCable DQoS; it is how the Cisco BTS10200
  Softswitch call management server (CMS) determines the CMTS to which a media
  terminal adapter (MTA) is attached, so it can issue gate control commands to the
  correct CMTS.
      VARCHAR(16): 1-16 ASCII characters.

  node - Defines the hybrid fiber coax (HFC) fiber node the MTA is homed to. The HFC
  fiber node sits between the CMTS and the MTA. Every MTA is assigned to a
  particular node-then one or more nodes are assigned to a given CMTS.
      VARCHAR(20): 1-20 ASCII characters.

  host_ip_address - IP Address of the EMS machine servicing the response. Is only
  applicable in response object.

  host_id - configured EMS hostname of the EMS machine servicing the response. Is
  only applicable in response object.

  timestamp - Timestamp the reply from BTS EMS is done with processing the request.
  Is only applicable in response object.

  reply_string - Reply string as seen from BTS EMS notifying success or failure and
  details of failure if available. Is only applicable in response object.

  key - Security Key of the user session.

-->
<xs:schema targetNamespace="http://www.cisco.com/MtaExtension"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.cisco.com/MtaExtension"
elementFormDefault="qualified">
  <xs:element name="id" type="xs:string"/>

```

```

<xs:element name="aggr_id" type="xs:string"/>
<xs:element name="node" type="xs:string"/>
<xs:element name="type">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="RGW"/>
      <xs:enumeration value="TGW"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="host_ip_address" type="xs:string"/>
<xs:element name="host_id" type="xs:string"/>
<xs:element name="timestamp" type="xs:string"/>
<xs:element name="reply_string" type="xs:string"/>
<xs:element name="key" type="xs:string"/>
</xs:schema>

```

PcspService

This section provides the extensions for the PcspService objects.

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

```
<!--
```

- Copyright (c) 2007 by Cisco Systems, Inc.
- All rights reserved.

sub_profile_id - Foreign key: Subscriber Profile table. Subscriber profile ID.
 VARCHAR(16): 1-16 ASCII characters.

account_id - Specifies a subscriber account ID for one or more subscribers. An account id is only for identification purposes. The account id is not associated with any feature grouping behavior during call processing, although it is included in a billing record.

VARCHAR(20): 1-20 ASCII characters.

address1 - Street name, apartment number.

VARCHAR(32): 1-32 ASCII characters.

address2 - Street name, apartment number.

VARCHAR(32): 1-32 ASCII characters.

aor_id - Mandatory and permitted term-type = SIP. Foreign key: Address of Record table. The address of record id. Uses the domain name parser. Domain name portion of the AOR-ID must exist in the Serving Domain Name table.AOR =USER +@+DOMAIN-NAME.

VARCHAR(64): 1-64 ASCII characters.

billing_type Specifies the subscriber billing type.

VARCHAR(16): 1-16 ASCII characters (Default = NONE). Permitted values are:

- FR1-Flat Rate Billing 1
- FR2-Flat Rate Billing 2
- MR1-Measured Rate Billing 1
- MR2-Measured Rate Billing 2
- NONE (Default)-Billing type is not used.

category - Subscriber category.

VARCHAR(15): 1-15 ASCII characters. Permitted values are:

- INDIVIDUAL (Default)-Individual subscriber.
- CTXG-Assigned to the main subscriber ID of a Centrex group.
- CTXG-INDIVIDUAL-Assigned to a Centrex subscriber.

CTXG-MLHG-Assigned to a Centrex MLHG (for example, attendant).
 CTXG-TG-Assigned to a Centrex trunk group.
 IVR-Access DN for IVR.
 MLHG-INDIVIDUAL- Subscriber within an MLHG.
 MLHG-Main subscriber ID of an MLHG.
 MLHG-PREF-INDIV-Main subscriber ID of a preferential hunt list.
 PBX-Assigned to the main subscriber ID of a PBX.
 RACF-Access DN for remote activation of call forwarding.

city - Subscriber's city.

VARCHAR(16): 1-16 ASCII characters.

cos_restrict_id - Foreign key: COS Restrict table. COS restriction ID.

VARCHAR(16): 1-16 ASCII characters.

country - Country of subscriber.

VARCHAR(16): 1-16 ASCII characters.

ctxg_id - Foreign key: Centrex Group table. Same as ID in Centrex Group table.

VARCHAR(16): 1-16 ASCII characters.

cwt_type - Call Waiting Tone type. For Multiple DN (MDN) feature, a different Call Waiting Tone Type can be assigned to each DN.

VARCHAR(8): WT1, WT2, WT3, WT4 (Default = WT1).

esrn - Emergency service routing number. The subscriber level ESRN is used if available, else the ESRN assigned to the subscriber profile is used.

VARCHAR(7): 1-7 numeric digits.

forced - Use to bypass rule checking during add or change operations. When used in a delete command, the Cisco BTS 10200 Softswitch deletes all the information associated with the subscriber from the Subscriber table as well as all associated tables such as the Subscriber Service Profile table, the Subscriber Feature Data table, and so forth.

CHAR(1): Y/N (Default = N).

grp - Specifies whether a subscriber is an individual or a group. Applies to Multiline Hunt or Centrex subscribers.

CHAR(1): Y/N (Default = N).

Y-Group

N-Individual

h323_term_id - Mandatory if term-type = H323. Foreign key: H.323 Termination table. The termination ID for an H.323 terminal.

VARCHAR(16): 1-16 ASCII characters.

immediate_release - Specifies whether to apply the immediate release procedure when a calling party hangs up.

VARCHAR(1): Y/N (Default = N).

mac_id - (EMS only) A MAC-ID is allowed only if term-type=sip. This token is case sensitive-enter in uppercase. Foreign key: MAC to Subscriber table. MAC-ID of SIP (IP) Phone. Specifies the MAC address of the subscriber's SIP phone from which services to control features are invoked.

VARCHAR(16): 1-16 ASCII characters.

mgw_id - Mandatory and permitted if term-type = term. Foreign key: Media Gateway, Termination tables. Media Gateway ID. The mgw-id is used as a termination ID as well as a combined foreign key to the termination table.

VARCHAR(32): 1-32 ASCII characters.

mlgh_id - Foreign key: Multiline Hunt Group (MLHG) table. Same as ID in Multiline Hunt Group table.

VARCHAR(16): 1-16 ASCII characters.

mlhg_pref_list_id - Foreign key: MLGH Preference List table. Same as ID in MLGH Preference List table.

VARCHAR(16): 1-16 ASCII characters.

ohd_timer - Off-hook Delay Timer. If an off-hook trigger is set, then the Cisco BTS 10200 Softswitch looks at the ohd-timer token and applies a dial-tone timer equal to ohd-timer. If a user dials any digit within this timer value, the remaining digits are collected as normal. If no digit is dialed before the ohd-timer expires, the Cisco BTS 10200 Softswitch performs off-hook trigger processing. Note Applies only if offhook-trigger-type=OHD.

INTEGER: 0, or 2-16 seconds (Default = 0). Note If 0 is provisioned, then the default value in the Call Agent Configuration table is used.

offhook_trigger_type - Indicate whether an off-hook Trigger applies to this subscriber.

VARCHAR(8): 1-8 ASCII characters (Default = NONE). Permitted values are:

OHD-Off-hook Delay trigger.

OHI-Off-hook Immediate trigger.

NONE (Default)-No Off-hook trigger. Note Only allowed if term-type=term.

policy_id - Mandatory and permitted term-type = SIP. The route policy id.

VARCHAR(16): 1-16 ASCII characters.

privacy - Specifies how calling party information (Name, DN) is displayed to the called party.

CHAR(4). Permitted values are:

NONE (Default)-Name and number.

FULL-Do not display name or number.

NAME-Do not display name.

USER-Use user-provided privacy information. Applies only to SIP endpoints that can include privacy information. If information is not received for either name or number, then privacy is indicated as "unspecified." Note PRIVACY = USER is not valid if term-type = term | none.

privacy_manager_id - Foreign key: Application Server table. The privacy manager id.

VARCHAR(16): 1-16 ASCII characters.

qos_id - Foreign key: QOS table. Specifies whether to use QOS index for codec selection.

CHAR(16): 1-16 ASCII characters.

sdt_mwi - Specifies whether a stutter tone is provided as a Message Waiting Indication.

CHAR(1): Y / N (Default = Y).

Y-A stutter dial tone is provided to a user when the subscriber goes off-hook and subscriber has Message Waiting.

N-A stutter dial tone is not provided.

secure_fqdn - (EMS Only) Permitted only if term-type=sip. Use to resolve an IP address and compare it with the IP address received from an endpoint during registration or during call setup (INVITE). A secure-fqdn is assigned to the AOR.

VARCHAR(64): 1-64 ASCII characters. Also provisioned in the AOR2SUB table.

Note This token is valid only if term-type=sip. If secure-fqdn is provisioned, the AOR token is required.

send_bdn_as_cpn - Specifies whether to send the billing DN for all calls (EMG, LOCAL, ON-NET, OFF-NET) as the CPN in the outgoing setup messages. If a billing DN is not provisioned, the billing DN (present) or the DN1 of the main subscriber (for PBX/MLHG subscribers) is sent as the CPN in the outgoing setup message. If none of them are found, the DN1 is sent.

CHAR(1): Y/N (Default = N).
 Y-Send the billing DN of a subscriber as a CPN in the outgoing setup message.
 N-Send the DN1 of a subscriber as a CPN in the outgoing setup message.

send_bdn_for_emg - Specifies whether to send the billing DN value of emergency calls for PBX over PRI, or CAS and ANALOG, subscribers as the CPN in the outgoing setup messages.
 (1): Y/N (Default = N).
 Y.Send the billing DN of the subscriber as a CPN in the outgoing setup message.
 N.Send the DN1 of the subscriber as a CPN in the outgoing setup message.

state - State designation.
 VARCHAR(16): 1-16 ASCII characters.

static_contact_host - Hostname portion of the contact id as in user@hostname.
 VARCHAR(32): 1-32 ASCII characters.

static_contact_port - Port number if different from default SIP port number (5060).
 INTEGER: 1-4 numeric characters (Default = 5060).

static_contact_user - User portion of the contact id as in user@hostname.
 VARCHAR(32): 1-32 ASCII characters.

term_id - Mandatory and permitted term-type = term. Unique key. Foreign key: Termination table. Termination ID. Required if term-type=term. The term-id is used as a termination ID as well as a combined foreign key to the termination table.
 VARCHAR(32): 1-32 ASCII characters.

termination_immediate_rel - Specifies whether to release a call immediately when a called party releases a call.
 CHAR(1): Y/N (Default = N).

term_type - Specifies the termination type.
 VARCHAR(5): 1-5 ASCII characters. Permitted values are:
 TERM (Default)-MGCP termination
 TG-Trunk group
 ROUTE-Route
 RG-Route guide (Not supported)
 SIP-SIP termination
 NONE-There is no termination associated with the subscriber.
 H323-H.323 virtual termination

tgn_id - (or tg) Mandatory and permitted term-type=tg. Foreign key: Trunk Group table. Trunk group ID. This field can also be provisioned using tg instead of tgn-id. The EMS looks up the tgn-id based on the trunk group and then provisions it.
 INTEGER: 1-99999999.

tg - (or tgn_id) Mandatory and permitted term-type=tg. Foreign key: Trunk Group table. Trunk group ID. This field can also be provisioned using tg instead of tgn-id. The EMS looks up the tgn-id based on the trunk group and then provisions it.
 INTEGER: 1-99999999.

usage_sens - Specifies if usage-sensitive features are allowed. Usage-sensitive features are not subscribed but are available for use. The subscribers are charged per usage. The following features are available as usage-sensitive:
 - AR (automatic recall)
 - AC (automatic callback)

- USTWC (three-way calling)
 CHAR(1): Y/N (Default = Y).
 Y-Usage-sensitive features are allowed.
 N-Usage-sensitive features are not allowed.

vmwi - The Visual Message Waiting Indication. This indication is sent when a message waiting indicator (MWI) Notify is received from the voice-mail system.
 CHAR(1): Y/N (Default = Y).

voice_mail_id - Foreign key: Application Server table. Specifies the default voice-mail id for all subscribers belonging to a subscriber profile.
 VARCHAR(16): 1-16 ASCII characters.

zipcode Subscriber's zip code.
 VARCHAR(10): 1-10 ASCII characters.

user_type - Defines the type of user.
 VARCHAR(16): 1.16 ASCII characters. Permitted values are:
 IP (Default)
 PHONE

host_ip_address - IP Address of the EMS machine servicing the response. Is only applicable in response object.

host_id - configured EMS hostname of the EMS machine servicing the response. Is only applicable in response object.

timestamp - Timestamp the reply from BTS EMS is done with processing the request. Is only applicable in response object.

reply_string - Reply string as seen from BTS EMS notifying success or failure and details of failure if available. Is only applicable in response object.

key - Security Key of the user session.

-->

```
<xs:schema targetNamespace="http://www.cisco.com/ServiceExtension"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.cisco.com/ServiceExtension"
elementFormDefault="qualified">
  <xs:element name="account_id" type="xs:string"/>
  <xs:element name="address1" type="xs:string"/>
  <xs:element name="address2" type="xs:string"/>
  <xs:element name="aor_id" type="xs:string"/>
  <xs:element name="city" type="xs:string"/>
  <xs:element name="cos_restrict_id" type="xs:string"/>
  <xs:element name="country" type="xs:string"/>
  <xs:element name="ctxg_id" type="xs:string"/>
  <xs:element name="esrn" type="xs:int"/>
  <xs:element name="forced" type="xs:boolean"/>
  <xs:element name="grp" type="xs:boolean"/>
  <xs:element name="h323_term_id" type="xs:string"/>
  <xs:element name="immediate_release" type="xs:boolean"/>
  <xs:element name="mac_id" type="xs:string"/>
  <xs:element name="mgw_id" type="xs:string"/>
  <xs:element name="mlhg_id" type="xs:string"/>
  <xs:element name="mlhg_pref_list_id" type="xs:string"/>
  <xs:element name="policy_id" type="xs:string"/>
  <xs:element name="privacy_manager_id" type="xs:string"/>
  <xs:element name="qos_id" type="xs:string"/>
  <xs:element name="sdt_mwi" type="xs:boolean"/>
  <xs:element name="secure_fqdn" type="xs:string"/>
  <xs:element name="send_bdn_as_cpn" type="xs:boolean"/>
  <xs:element name="send_bdn_for_emg" type="xs:boolean"/>
```

```

<xs:element name="state" type="xs:string"/>
<xs:element name="static_contact_host" type="xs:string"/>
<xs:element name="static_contact_port" type="xs:int"/>
<xs:element name="static_contact_user" type="xs:string"/>
<xs:element name="sub_profile_id" type="xs:string"/>
<xs:element name="term_id" type="xs:string"/>
<xs:element name="terminating_immediate_rel" type="xs:boolean"/>
<xs:element name="tg" type="xs:string"/>
<xs:element name="tgn_id" type="xs:int"/>
<xs:element name="usage_sens" type="xs:boolean"/>
<xs:element name="vmwi" type="xs:boolean"/>
<xs:element name="voice_mail_id" type="xs:string"/>
<xs:element name="zipcode" type="xs:string"/>
<xs:element name="billing_type">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="FR1"/>
      <xs:enumeration value="FR2"/>
      <xs:enumeration value="MR1"/>
      <xs:enumeration value="MR2"/>
      <xs:enumeration value="NONE"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="category">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="INDIVIDUAL"/>
      <xs:enumeration value="MLHG"/>
      <xs:enumeration value="MLHG_INDIVIDUAL"/>
      <xs:enumeration value="MLHG_PREF_INDIV"/>
      <xs:enumeration value="CTXG"/>
      <xs:enumeration value="CTXG_INDIVIDUAL"/>
      <xs:enumeration value="PBX"/>
      <xs:enumeration value="CTXG_TG"/>
      <xs:enumeration value="CTXG_MLHG"/>
      <xs:enumeration value="RACF"/>
      <xs:enumeration value="IVR"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="cwt_type">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="WT1"/>
      <xs:enumeration value="WT2"/>
      <xs:enumeration value="WT3"/>
      <xs:enumeration value="WT4"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="offhook_trigger_type">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="NONE"/>
      <xs:enumeration value="OHD"/>
      <xs:enumeration value="OHI"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="ohd_timer">
  <xs:simpleType>
    <xs:restriction base="xs:int">
      <xs:enumeration value="0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

```

```

        <xs:enumeration value="2" />
        <xs:enumeration value="3" />
        <xs:enumeration value="4" />
        <xs:enumeration value="5" />
        <xs:enumeration value="6" />
        <xs:enumeration value="7" />
        <xs:enumeration value="8" />
        <xs:enumeration value="9" />
        <xs:enumeration value="10" />
        <xs:enumeration value="11" />
        <xs:enumeration value="12" />
        <xs:enumeration value="13" />
        <xs:enumeration value="14" />
        <xs:enumeration value="15" />
        <xs:enumeration value="16" />
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="privacy">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="FULL" />
            <xs:enumeration value="NAME" />
            <xs:enumeration value="NONE" />
            <xs:enumeration value="USER" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="term_type">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="TERM" />
            <xs:enumeration value="TG" />
            <xs:enumeration value="ROUTE" />
            <xs:enumeration value="RG" />
            <xs:enumeration value="SIP" />
            <xs:enumeration value="NONE" />
            <xs:enumeration value="H323" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="user_type">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="IP" />
            <xs:enumeration value="PHONE" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<!--
ListOfServicePkg - a list of BTS Service Package IDs.
-->
<xs:element name="ListOfServicePkg">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="pkgid" type="xs:string"
                minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>

```

```

</xs:element>
<xs:element name="host_ip_address" type="xs:string"/>
<xs:element name="host_id" type="xs:string"/>
<xs:element name="reply_string" type="xs:string"/>
<xs:element name="key" type="xs:string"/>
</xs:schema>

```

PospEndpoint

This section provides the extensions for the PospEndpoint objects.

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
- Copyright (c) 2007 by Cisco Systems, Inc.
- All rights reserved.

sub_id - Foreign key: Subscriber table. Subscriber ID of line termination: same as
the ID in the Subscriber table.
    VARCHAR(30): 1-30 ASCII characters.
-->
<xs:schema targetNamespace="http://www.cisco.com/EndpointExtension"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.cisco.com/EndpointExtension"
elementFormDefault="qualified">
  <xs:element name="sub_id" type="xs:string"/>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="LINE"/>
      <xs:enumeration value="TRUNK"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="host_ip_address" type="xs:string"/>
<xs:element name="host_id" type="xs:string"/>
<xs:element name="timestamp" type="xs:string"/>
<xs:element name="reply_string" type="xs:string"/>
<xs:element name="key" type="xs:string"/>
</xs:schema>

```

PospCiscoBtsCli

These are no extensions for the PospCiscoBtsCli objects since the entire XmlEncoding string is request/response.



APPENDIX **B**

Sample Entity Encodings

Revised: July 2010, OL-23044-01

PcspService Object Example

This section contains PcspService object examples.

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

The following text shows an example of service object encoding.

The default and the PCSP namespace are set to PcspI01.

The PCSP namespace is a convenience, allowing vendor extensions to reference elements from the main PCSP schema.

```
->
<PcspService xmlns="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:pcsp="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xsi:noNamespaceSchemaLocation="PcspI01.xsd">
<!--
```

The following text shows a sample service object.

```
-->
<ServiceId format="NSN">9785551212</ServiceId>
<AdminStatus>1</AdminStatus>
<BillingId>9785550000</BillingId>
<ExternalId>0123456789</ExternalId>
<IsPrimary>>true</IsPrimary>
<PrimaryRing>IndexIntoCadenceTable</PrimaryRing>
<DisplayName>John Q Public</DisplayName>
<DisplayNumber>(978)-555-1212</DisplayNumber>
<Password>45hjg3j6gkg6h54j6gkj3g6</Password>
<Announcements>
<Language>EN</Language>
<Timezone>+0500</Timezone>
</Announcements>
<InterExchange>
<PIC>0123</PIC>
<LPIC>0123</LPIC>
<IPIC>0123</IPIC>
</InterExchange>
<LNP>
<PortingStatus>0</PortingStatus>
```

```

<LNPT>0</LNPT>
</LNP>
<ListOfCallFeatures>
<CfCND>
<Subscribed>>true</Subscribed>
<AdminStatus>1</AdminStatus>
</CfCND>
<CfCIDB>
<Subscribed>0</Subscribed>
<AdminStatus>1</AdminStatus>
<UserParams>
<Flag>PUBLIC</Flag>
</UserParams>
</CfCIDB>
<CfCFBL>
<Subscribed>>true</Subscribed>
<AdminStatus>1</AdminStatus>
<UserParams>
<Active>>true</Active>
<Always>0</Always>
<ForwardTo>9785551212</ForwardTo>
</UserParams>
</CfCFBL>
<CfSPCALL>
<Subscribed>0</Subscribed>
<AdminStatus>1</AdminStatus>
<UserParams>
<ListOfSpeedDial>
<SdPair>
<SdNum>1</SdNum>
<ServiceId>9785551212</ServiceId>
</SdPair>
<SdPair>
<SdNum>3</SdNum>
<ServiceId>9785551000</ServiceId>
</SdPair>
</ListOfSpeedDial>
</UserParams>
</CfSPCALL>
<CfRDA>
<Subscribed>1</Subscribed>
<AdminStatus>1</AdminStatus>
</CfRDA>
<CfLSR>
<Subscribed>1</Subscribed>
<AdminStatus>1</AdminStatus>
<UserParams>
<BlkDomLongDist>1</BlkDomLongDist>
<BlkIntLongDist>1</BlkIntLongDist>
<BlkPayPerCall>1</BlkPayPerCall>
<BlkOperatorAssist>1</BlkOperatorAssist>
<BlkDirAssist>1</BlkDirAssist>
<BlkTollFree>1</BlkTollFree>
<ListOfServiceId>
<ServiceId>9895551001</ServiceId>
<ServiceId>9895551002</ServiceId>
<ServiceId>9895551003</ServiceId>
</ListOfServiceId>
</UserParams>
</CfLSR>
<CfDND>
<Subscribed>1</Subscribed>
<AdminStatus>1</AdminStatus>
<UserParams>

```

```

<Active>true</Active>
<WdStartTod1>00:00:00+05:00</WdStartTod1>
<WdStopTod1>06:00:00+05:00</WdStopTod1>
<WdStartTod2>18:00:00+05:00</WdStartTod2>
<WdStopTod2>20:00:00+05:00</WdStopTod2>
<WeStartTod1>00:00:00+05:00</WeStartTod1>
<WeStopTod1>09:00:00+05:00</WeStopTod1>
<WeStartTod2>18:00:00+05:00</WeStartTod2>
<WeStopTod2>20:00:00+05:00</WeStopTod2>
</UserParams>
</CfDND>
<CfCOC>
<Subscribed>1</Subscribed>
<AdminStatus>1</AdminStatus>
<UserParams>
<Active>true</Active>
<StartTod>00:00:00+05:00</StartTod>
<StopTod>06:00:00+05:00</StopTod>
<ListOfServiceId>
<ServiceId>9895551001</ServiceId>
<ServiceId>9895551002</ServiceId>
<ServiceId>9895551003</ServiceId>
</ListOfServiceId>
</UserParams>
</CfCOC>
<CfNSA>
<Subscribed>1</Subscribed>
<AdminStatus>1</AdminStatus>
<UserParams>
<Active>true</Active>
<StartTod>00:00:00+05:00</StartTod>
<StopTod>06:00:00+05:00</StopTod>
<ListOfServiceId>
<ServiceId>9895551001</ServiceId>
<ServiceId>9895551002</ServiceId>
<ServiceId>9895551003</ServiceId>
</ListOfServiceId>
</UserParams>
</CfNSA>
</ListOfCallFeatures>

<!--

```

Here, we extend the data content of the main body of the service object.

```

-->
<Extension xmlns:ext="http://www.cablelabs.com/SampleExtension">
<ext:ServiceAccountId>Test</ext:ServiceAccountId>
<ext:ServiceAddress1>Test</ext:ServiceAddress1>
<ext:ServiceAddress2>Test</ext:ServiceAddress2>
<ext:ServiceAorId>Test</ext:ServiceAorId>
<ext:ServiceBillingType>FR1</ext:ServiceBillingType>>
<ext:ServiceCategory>INDIVIDUAL</ext:ServiceCategory>>
<ext:ServiceCity>Test</ext:ServiceCity>
<ext:ServiceCosRestrictId>Test</ext:ServiceCosRestrictId>
<ext:ServiceCountry>Test</ext:ServiceCountry>
<ext:ServiceCtxgId>Test</ext:ServiceCtxgId>
<ext:ServiceCwtType>WT1</ext:ServiceCwtType>>
<ext:ServiceEsrn>Test</ext:ServiceEsrn>
<ext:ServiceForced>Y</ext:ServiceForced>
<ext:ServiceGrp>Y</ext:ServiceGrp>
<ext:ServiceH323TermId>Test</ext:ServiceH323TermId>
<ext:ServiceSubId>Test</ext:ServiceSubId>
<ext:ServiceImmediateRelease>Y</ext:ServiceImmediateRelease>

```

```

<ext:ServiceLanguageId>Test</ext:ServiceLanguageId>
<ext:ServiceLnpTrigger>Y</ext:ServiceLnpTrigger>
<ext:ServiceMacId>Test</ext:ServiceMacId>
<ext:ServiceMgwId>Test</ext:ServiceMgwId>
<ext:ServiceMlhgId>Test</ext:ServiceMlhgId>
<ext:ServiceMlhgPrefListId>Test</ext:ServiceMlhgPrefListId>
<ext:ServiceOffhookTriggerType>OHD</ext:ServiceOffhookTriggerType>>
<ext:ServiceCwtType>16</ext:ServiceCwtType>>
<ext:ServicePolicyId>Test</ext:ServicePolicyId>
<ext:ServicePortedIn>Test</ext:ServicePortedIn>
<ext:ServicePrivacy>FULL</ext:ServicePrivacy>>
<ext:ServicePrivacyManagerId>Test</ext:ServicePrivacyManagerId>
<ext:ServiceQosId>Test</ext:ServiceQosId>
<ext:ServiceRingTypeDn1>R0</ext:ServiceRingTypeDn1>>
<ext:ServiceSdtMwi>Y</ext:ServiceSdtMwi>
<ext:ServiceSecureFqdn>Test</ext:ServiceSecureFqdn>
<ext:ServiceSendBdnAsCpn>Y</ext:ServiceSendBdnAsCpn>
<ext:ServiceSendBdnForEmg>Y</ext:ServiceSendBdnForEmg>
<ext:ServiceState>Test</ext:ServiceState>
<ext:ServiceStaticContactHost>Test</ext:ServiceStaticContactHost>
<ext:ServiceStaticContactPort>2345</ext:ServiceStaticContactPort>
<ext:ServiceStaticContactUser>Test</ext:ServiceStaticContactUser>
<ext:ServiceSubProfileId>Test</ext:ServiceSubProfileId>
<ext:ServiceTermId>Test</ext:ServiceTermId>
<ext:ServiceTermType>TERM</ext:ServiceTermType>>
<ext:ServiceTerminatingImmediateRel>Y</ext:ServiceTerminatingImmediateRel>
<ext:ServiceTg>Test</ext:ServiceTg>
<ext:ServiceTgnId>254</ext:ServiceTgnId>
<ext:ServiceUsageSens>Y</ext:ServiceUsageSens>
<ext:ServiceUserType>IP</ext:ServiceUserType>>
<ext:ServiceVmwi>Y</ext:ServiceVmwi>
<ext:ServiceVoiceMailId>Test</ext:ServiceVoiceMailId>
<ext:ServiceZipcode>75082</ext:ServiceZipcode>
</Extension>
</PcspService>

```

PcspEndpoint Object Example

This section contains PcspEndpoint object examples.

```
<?xml version="1.0" encoding="UTF-8"?>
<PcspEndpoint xmlns="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:pcsp="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance">
<!--
```

The following text shows a sample endpoint object.

```
-->
<EndpointId>aaln/1@mta01.cablelabs.com</EndpointId>
<AdminStatus>2</AdminStatus>
<Protocol>MtaDefault</Protocol>
<Codec>2</Codec>
<IPSecControl>>true</IPSecControl>

<!--
```

Here, we extend the data content of the main body of the endpoint object.

```
-->
<Extension xmlns:ext="http://www.cablelabs.com/SampleExtension">
<ext:EndpointType>TGW</ext:EndpointType>
<ext:EndpointPortStart>1</ext:EndpointPortStart>
<ext:EndpointPortEnd>32</ext:EndpointPortEnd>
<ext:EndpointPortPrefix>aaln</ext:EndpointPortPrefix>
<ext:EndpointMgwId>MGW_PROF_123</ext:EndpointMgwId>
<ext:EndpointSubId>201-222-0101</ext:EndpointSubId>
<ext:EndpointTgnId>23</ext:EndpointTgnId>
<ext:EndpointTrunkId>7</ext:EndpointTrunkId>
</Extension>
</PcspEndpoint>
```

PcspMta Object Example

This section contains PcspMta object examples.

```
<?xml version="1.0" encoding="UTF-8"?>
<PcspMta xmlns="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:pcsp="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xsi:noNamespaceSchemaLocation="PcspI01.xsd">
<!--
```

The following text shows a sample multimedia terminal adapter (MTA) object.

```
-->
<MtaFqdn>mta01.cablelabs.com</MtaFqdn>
<ListenPort>2427</ListenPort>
<CmtsFqdn>cmta01.cablelabs.com</CmtsFqdn>
<Timezone>-0500</Timezone>
<Protocol>MCGP 1.0 NCS 1.0</Protocol>
<Codec>5</Codec>
<IPSecControl>>true</IPSecControl>

<!--
```

Here, we extend the data content of the main body of the MTA object.

```
-->
<Extension xmlns:ext="http://www.cablelabs.com/SampleExtension">
<ext:MtaId>Testing</ext:MtaId>
<ext:MtaAggrId>TestAGGR</ext:MtaAggrId>
<ext:MtaCallAgentControlPort>0</ext:MtaCallAgentControlPort>
<ext:MtaNode>CATEST</ext:MtaNode>
<ext:MtaType>RGW</ext:MtaType>
</Extension>
</PcspMta>
```

PcspCms Object Example

This section contains PcspCms object examples.

```
<?xml version="1.0" encoding="UTF-8"?>
<PcspCms xmlns="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:pcsp="http://www.cablelabs.com/Pcsp/I01/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance">
<!--
```

The following text shows a sample call management system (CMS) object.

Not much defined yet...just its key.

The CMS object serves as a collection for services and endpoints.

```
-->
<CmsFqdn>cma01.cablelabs.com</CmsFqdn>

<!--
```

Here, we extend the data content of the main body of the CMS object.

```
-->
<Extension xmlns:ext="http://www.cablelabs.com/SampleExtension">
<ext:CmsId>CA146</ext:CmsId>
<ext:CmsClli>dallasrcdn7</ext:CmsClli>
<ext:CmsMgwMonitoringEnabled>Y</ext:CmsMgwMonitoringEnabled>
</Extension>
</PcspCms>
```

