Oracle® Communications Policy Management

Bandwidth on Demand Cable User's Guide

Release 11.5.2

E65489 Revision 01

August 2015

ORACLE®

Oracle® Communications Policy Management Bandwidth on Demand Cable User's Guide, Release 11.5.2

Table of Contents

List of Figures

List of Tables

Chapter 1

About This Guide

Topics:

This chapter describes the organization of the document and provides other information that could be useful to the reader.

Introduction

This guide describes how to implement the Bandwidth on Demand Application Manager (BoD AM).

Conventions

The following conventions are used throughout this guide:

- **Bold text** in procedures indicates icons, buttons, links, or menu items that you click on.
- *Italic text* indicates variables.
- `Monospace text` indicates text displayed on screen.
- **`Monospace bold text`** indicates text that you enter exactly as shown.

Scope and Audience

This guide is intended for the following trained and qualified service personnel who are responsible for operating Policy Management devices:

- System operators
- System administrators

How This Guide is Organized

The information in this guide is presented in the following order:

- *About This Guide* contains general information about this guide, the organization of this guide, and how to get technical assistance.
- *Introduction* contains an overview of the guide contents.
- *Using the BoD AM User Interface* describes how to use the BoD Graphical User Interface.
- *Configuring BoD AM Servers* describes how to configure BoD servers.
- *BoD Server Reports* describes how to view information about the BoD cluster system, individual blades, and protocol statistics.
- *Viewing and Modifying the BoD Trace Log* describes how to view the trace log for the BoD server and modify the trace log settings.
- *Viewing Active BoD Sessions* describes how to view active sessions within the BoD application.
- *Managing PCMM Services* describes how to manage the PCMM Service Management features within the BoD application.
- *BoD AM WSDL Definitions* describes how to use WSDL script files.
- *BoD Interface Error Codes* describes the BoD interface error codes displayed by the BoD application.

Documentation Admonishments

Admonishments are icons and text throughout this manual that alert the reader to assure personal safety, to minimize possible service interruptions, and to warn of the potential for equipment damage.

Table 1: Admonishments

Icon	Description
	Danger: (This icon and text indicate the possibility of *personal injury*.)
	Warning: (This icon and text indicate the possibility of *equipment damage*.)
	Caution: (This icon and text indicate the possibility of *service interruption*.)
	Topple: (This icon and text indicate the possibility of *personal injury* and *equipment damage*.)

Related Publications

For information about additional publications that are related to this document, refer to the *Related Publications Reference* document, which is published as a separate document on the Oracle Technology Network (OTN) site. See *Locate Product Documentation on the Oracle Technology Network Site* for more information.

Locate Product Documentation on the Oracle Technology Network Site

Oracle customer documentation is available on the web at the Oracle Technology Network (OTN) site, *http://docs.oracle.com*. You do not have to register to access these documents. Viewing these files requires Adobe Acrobat Reader, which can be downloaded at *http://www.adobe.com*.

1. Access the Oracle Technology Network site at *http://docs.oracle.com*.
2. Click **`Industries`**.

3. Under the Oracle Communications subheading, click the **Oracle Communications documentation** link.
   The Oracle Communications Documentation page appears with Tekelec shown near the top.
4. Click the **Oracle Communications Documentation for Tekelec Products** link.
5. Navigate to your Product and then the Release Number, and click the **View** link (the Download link will retrieve the entire documentation set).
   A list of the entire documentation set for the selected product and release appears.
6. To download a file to your location, right-click the **PDF** link, select **Save target as**, and save to a local folder.

Customer Training

Oracle University offers training for service providers and enterprises. Visit our web site to view, and register for, Oracle Communications training:

http://education.oracle.com/communication

To obtain contact phone numbers for countries or regions, visit the Oracle University Education web site:

www.oracle.com/education/contacts

My Oracle Support (MOS)

MOS (*https://support.oracle.com*) is your initial point of contact for all product support and training needs. A representative at Customer Access Support (CAS) can assist you with MOS registration.

Call the CAS main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at *http://www.oracle.com/us/support/contact/index.html*. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select **2** for New Service Request
2. Select **3** for Hardware, Networking and Solaris Operating System Support
3. Select one of the following options:
   - For Technical issues such as creating a new Service Request (SR), Select **1**
   - For Non-technical issues such as registration or assistance with MOS, Select **2**

You will be connected to a live agent who can assist you with MOS registration and opening a support ticket.

MOS is available 24 hours a day, 7 days a week, 365 days a year.

Emergency Response

In the event of a critical service situation, emergency response is offered by the Customer Access Support (CAS) main number at 1-800-223-1711 (toll-free in the US), or by calling the Oracle Support hotline for your local country from the list at *http://www.oracle.com/us/support/contact/index.html*. The emergency response provides immediate coverage, automatic escalation, and other features to ensure that the critical situation is resolved as rapidly as possible.

A critical situation is defined as a problem with the installed equipment that severely affects service, traffic, or maintenance capabilities, and requires immediate corrective action. Critical situations affect service and/or system operation resulting in one or several of these situations:

- A total system failure that results in loss of all transaction processing capability
- Significant reduction in system capacity or traffic handling capability
- Loss of the system's ability to perform automatic system reconfiguration
- Inability to restart a processor or the system
- Corruption of system databases that requires service affecting corrective actions
- Loss of access for maintenance or recovery operations
- Loss of the system ability to provide any required critical or major trouble notification

Any other problem severely affecting service, capacity/traffic, billing, and maintenance capabilities may be defined as critical by prior discussion and agreement with Oracle.

Chapter 2

Introduction

Topics:

This chapter provides an overview of the Bandwidth on Demand Application Manager (BoD AM) and the associated SOAP and HTTP Interface definitions.

Bandwidth on Demand Application Manager Overview

The Bandwidth on Demand Application Manager (BoD AM) allows applications to request the setup and teardown of dynamic Quality of Service (QoS) resources within a broadband network, providing the necessary bandwidth and priority to enhance the subscriber's experience.

The primary goal of the BoD AM is to provide a simplified and abstract interface for the purpose of creating dynamic service requests, allowing the application developer to integrate dynamic QoS resources into nearly any application. This is achieved by providing HTTP and Simple Object Access Protocol (SOAP) based interfaces that can easily be integrated into most application development environments.

Additionally, the BoD AM maintains and manages all of the state information that is associated with each request, allowing applications to be stateless in their operation.

The BoD AM presents a SOAP based RPC interface and a pure HTTP request interface. These interfaces provide similar functionality and are designed to allow application developers to use the interface that best suits their application.

For example, the HTTP interface allows a parameterized URL to be associated with the "onclick" action of a turbo-button, or simply allow any application to embed an HTTP POST message to dynamically adjust service. Alternatively, the SOAP interface provides easy session control through an RPC mechanism. The decision of HTTP vs. SOAP largely depends on the personal preferences of the developers of the calling application.

Within the BoD AM, the user can define a number of service names that translate into a particular traffic profile. For example, a generic service name "turboService" could be defined with an associated best effort upstream flow and a high-priority downstream flow. Additionally, a specific service name could be defined, such as "uploadService" that simply defines a high priority upstream flow.

Each of the interface bindings allows an application to create a new session, specifying a service name and also supplying a number of specialization parameters, such as bandwidth. For example, within a web portal, a number of links or buttons can be defined, all of which use the same "turboService" profile, each specifying a different upstream and downstream bandwidth. This can be used to vary the resulting QoS flows, either based on the application context or perhaps a subscriber tier.

The BoD AM also allows a calling of an application to specify the duration of QoS resource allocation. The application may choose to completely manage the lifecycle of the resources, in which case it is the responsibility of the application to free the resources at the appropriate time, either after a defined period, or once an application has completed its function. Alternatively, the application may simply tell the BoD AM to keep the resources active for a specified time, or until there is inactivity for a defined period.

BoD AM Georedundancy

BoD AM servers can support a georedundant configuration, consisting of a trio of servers (primary, standby, and spare) that form a georedundant cluster. This configuration allows the BoD AM to take advantage of High Availability by having backup servers in case of a server failure. The primary (active) BoD AM server replicates application state data to the standby and spare servers to provide continuity for BoD AM session management, including correlated PCMM Gate state information, during a catastrophic failure of the primary site or servers.

Georedundancy topology is managed through the Configuration Management Platform (CMP) as shown in *Figure 1: BoD AM Topology Overview*. For additional information on CMP systems, refer to the *Configuration Management Platform Cable User Guide*.

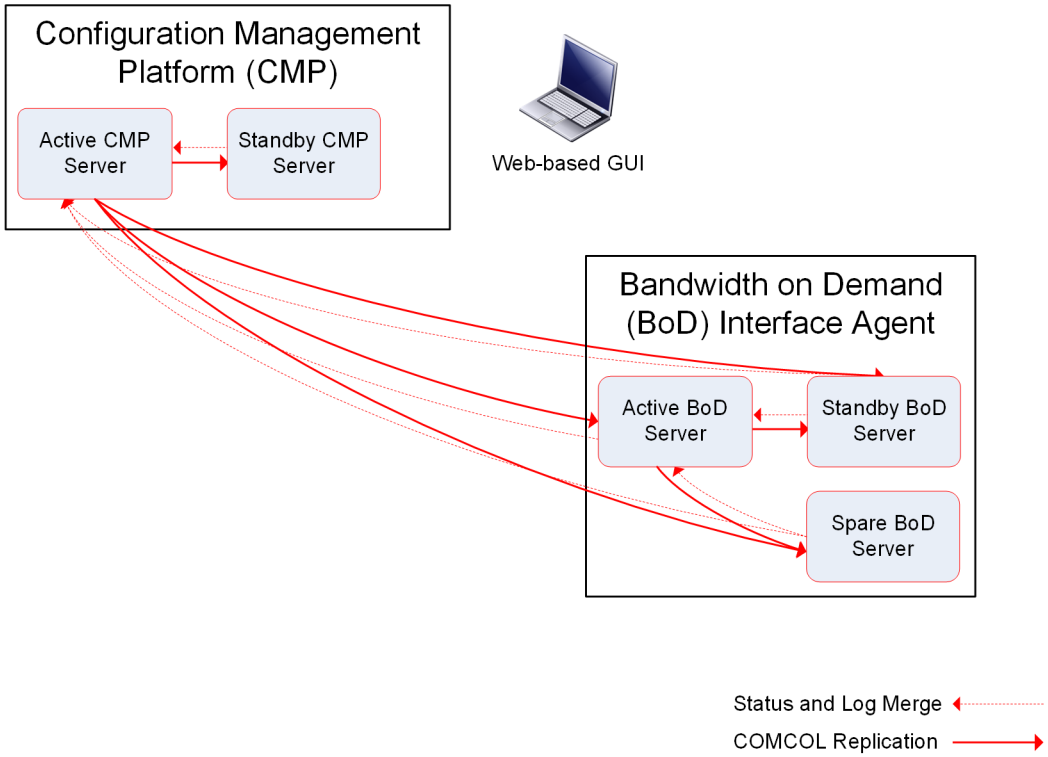


Figure 1: BoD AM Topology Overview

Within a georedundant cluster, the servers are connected through the OAM network. The servers work collaboratively as follows:

1. The active, standby, and spare servers communicate using several TCP connections over the OAM network to perform replication, monitor heartbeats, and merge events.
2. The servers share a virtual IP (VIP) cluster address to support automatic failover.
3. The COMCOL database runtime process constantly monitors the status of all servers in the cluster.
4. If the active server fails, it instructs the standby server to take over and become the active server.

The terms “active,” “standby,” and “spare” denote roles or states that the servers assume, and these roles or states can change, based on decisions made by the underlying COMCOL database, automatically and at any time. If both the active and standby servers become unavailable, the spare server automatically assumes the role or state of active server and continues to provide service.

The spare server can be located at a different site than the active and standby servers. If the two servers at one site become unavailable, the third server, located at another site, automatically continues to provide service.

Interface Overview

The BoD AM provides two distinct styles (Session and Traffic Classifier) of interface, each suited for different classes of applications. Additionally, Configuration Interfaces are available to set configuration settings for the BoD AM application.

The following figure shows the main system components and the protocols or interfaces used to communicate between them.

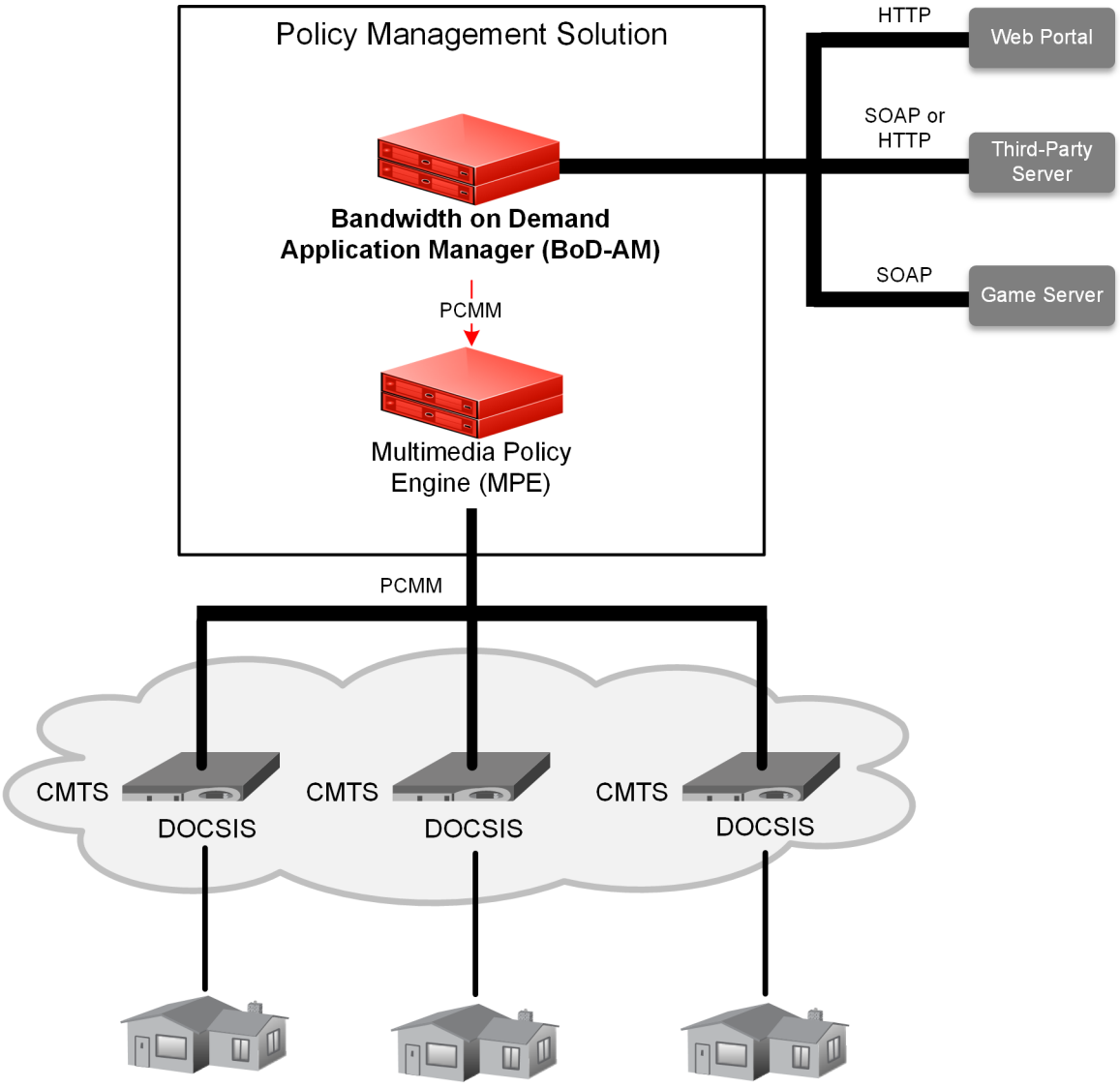


Figure 2: SOAP and HTTP Interface

Session Interface

For session-oriented applications, such as streaming media, file transfers, and bandwidth on demand, the Session Interface provides createSession, modifySession, and deleteSession interfaces. These interfaces allow the calling application to manage the lifecycle of the QoS session appropriate for the application. For example, a streaming media application may create a dedicated flow, specific for the bit rate and encoding of the content. When the content stream has finished or been terminated by the user, the application tears down the flow or allows the flow to be torn down automatically based on session inactivity. In addition, an MSO may wish to provide application specific "Turbo Buttons" on their web portal. When the subscriber presses the Turbo Button a new session is created based on parameters configured by the MSO. This session can again be torn down by re-clicking on the button or based on a session duration timer or based on session inactivity.

Traffic Classifier Interface

The Traffic Classifier Interface is used by applications that classify traffic, and want to allow the operator to assign this traffic to a particular Class of Service. This interface puts no requirements on the calling application to manage sessions, but instead allows the MSO to define the session parameters and the BoD AM to manage session setup and teardown based on the subscriber. For example, an application that detects worm and virus activity simply issues addTrafficClassifier calls when a worm or virus is detected and a deleteTrafficClassifier when the situation is resolved. In this situation, the MSO may define a WORM profile with reduced bandwidth and/or priority to carry the traffic based on the identified classification.

SOAP Interface Definitions

The following sections describe the SOAP interface supported by the BoD AM. Refer to *BoD AM WSDL Definitions* for a Web Services Description Language (WSDL) definition that can be used to generate client side code to call these functions.

SOAP Session Interfaces

For session-oriented applications, such as streaming media, file transfers, and bandwidth on demand, the SOAP Session Interface provides createSession and deleteSession interfaces that allow the calling application to manage the lifecycle of the QoS session appropriate for the application.

createSession Interface

The createSession interface allows an application to create a new dynamic PCMM session. The parameters of this interface are defined as follows:

```
public createSession(String pcmmServiceName, String groupName, String sessionId,
 int upBwKbps,
int upBwMaxKbps, int downBwKbps, int downBwMaxKbps, int durInMins, String subIp,
 String subMac,
int subPort,String serverIp, int serverPort, String subName, int upBwMinKbps,
int downBwMinKbps,
boolean disableRecreateIfRecoverableErrors, long vollimitKbps, long
```

```
upRequiredAttrMask,
long upForbiddenAttrMask, long upAttrAggrRuleMask, long upPeakTrafficRate, long
 downRequiredAttrMask,
long downForbiddenAttrMask, long downAttrAggrRuleMask, long downPeakTrafficRate,
 string cmtsIPAddress)
```

Note: The required parameters depend on the service definition (which parameters are tagged as "passed-in").

Where:

pcmmServiceName — specifies the service name, which defines the upstream and downstream traffic profiles. See *Creating a Service* for additional information. Note that the **pcmmServiceName** is not the PCMM Service Class Name defined within the DOCSIS and PacketCable specifications.

groupName — specifies the caller-supplied group name, assigned to the session, for reporting purposes.

sessionId — specifies the caller-supplied session identifier associated to the session, or null if no caller supplied identifier is needed.

upBwKbps — specifies the minimum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed. The valid values include 0 to 2^22-1 (4194303). Note that 2^32 bps = 2^22 kbps = 2^12 Mbps = 4096 Mbps.

upBwMaxKbps — specifies the upstream maximum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed. The valid values include 0 to 2^22-1 (4194303). Note that 2^32 bps = 2^22 kbps = 2^12 Mbps = 4096 Mbps.

downBwKbps — specifies the minimum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed. The valid values include 0 to 2^22-1 (4194303). Note that 2^32 bps = 2^22 kbps = 2^12 Mbps = 4096 Mbps.

downBwMaxKbps — specifies the downstream maximum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed. The valid values include 0 to 2^22-1 (4194303). Note that 2^32 bps = 2^22 kbps = 2^12 Mbps = 4096 Mbps.

durInMins — specifies the duration for the session, or 0 for a session of indefinite duration (where the BoD AM is responsible for teardown).

subPort — specifies the subscriber port. A value of 0 represents a wildcard value, and a value of -1 indicates that this argument is not needed.

destIp — specifies the destination IPv4 address.

destPort — specifies the destination port. A value of 0 represents a wildcard value, and a -1 value indicates that this argument is not needed.

disableRecreateIfRecoverableError — disables the BoD AM retry logic when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to true, when BoD AM encounters one of these errors, the retry is not attempted. If this parameter is omitted, BoD always tries to recover from any recoverable errors.

upRequiredAttrMask -- specifies the upstream required attribute mask.

upForbiddenAttrMask -- specifies the upstream forbidden attribute mask.

upAttrAggrRuleMask -- specifies the upstream required attribute aggregation rule mask.

downRequiredAttrMask -- specifies the downstream required attribute mask.

downForbiddenAttrMask -- specifies the downstream forbidden attribute mask.

downAttrAggrRuleMask -- specifies the downstream required attribute aggregation rule mask.

downPeakTrafficRate -- specifies the downstream peak traffic rate.

upPeakTrafficRate -- specifies the upstream peak traffic rate.

cmtsIPAddress -- specifies the IP address of the CMTS.

When this operation is successful, a newly created session id is returned (the caller-supplied session identifier if provided, or a system-generated session identifier if a caller supplied session id is not). If this operation is unsuccessful, an error result code is returned in the following format:

```
Error: <error code>
```

For a complete list of error codes, see *BoD Interface Error Codes*.

createSessionIPv6 Interface

The createSessionIPv6 interface allows an application to create a new dynamic PCMM session when an IPv6 classifier is used. The parameters of this interface are defined as follows:

```
public String createSessionIPv6 (String serviceName, String groupName,
    String sessionId, int upBwKbps, int upBwMaxKbps, int downBwKbps,
    int downBwMaxKbps, int durInMins, long vollimitKbps, String subIP,
    String subMac, String extSubIP, short subIpPrefixLength,
    int extSubPortStart, int extSubPortEnd, String extDestIP,
    short destIpPrefixLength, int extDestPortStart, int extDestPortEnd,
    String subName, int upBwMinKbps, int downBwMinKbps,
    boolean disableRecreateIfRecoverableErrors, long upRequiredAttrMask,
    long upForbiddenAttrMask, long upAttrAggrRuleMask,
    long downRequiredAttrMask, long downForbiddenAttrMask,
   long downAttrAggrRuleMask, long upPeakTrafficRate, long downPeakTrafficRate)
 ;
```

where:

serviceName — specifies the service name, which defines the upstream and downstream traffic profiles.

groupName — specifies the caller-supplied group name, assigned to the session, for reporting purposes.

sessionID — specifies the caller-supplied session identifier associated to the session, or null if no caller supplied identifier is needed.

upBwKbps — specifies the minimum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed. The valid values include 0 to 2^22-1 (4194303). Note that 2^32 bps = 2^22 kbps =2^12 Mbps = 4096 Mbps.

upBwMaxKbps — specifies the upstream maximum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed. The valid values include 0 to 2^22-1 (4194303). Note that 2^32 bps = 2^22kbps = 2^12 Mbps = 4096 Mbps.

downBwKbps — specifies the minimum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed. The valid values include 0 to 2^22-1 (4194303). Note that 2^32 bps = 2^22 kbps =2^12 Mbps = 4096 Mbps.

downBwMaxKbps — specifies the downstream maximum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed. The valid values include 0 to 2^22-1 (4194303). Note that 2^32bps = 2^22 kbps = 2^12 Mbps = 4096 Mbps.

durInMins — specifies the duration for the session, or 0 for a session of indefinite duration (where the BoD AM is responsible for teardown).

vollimitKbps — specifies the volume limit for the session.

subIP — specifies the subscriber IPv6 address. This identifies the subscriber requesting the QoS service.

extSubIP — specifies the source IPv6 address.

subIpPrefixLength — specifies how many high order bits in the corresponding IPv6 source Address to consider in determining a match. The valid values are from 0 - 128.

extSubPortStart — specifies the low-end TCP/UDP source port value.

extSubPortEnd — specifies the high-end TCP/UDP source port value.

extDestIP — specifies the destination IPv6 address.

destIpPrefixLength — specifies how many high order bits in the corresponding IPv6 destination Address to consider in determining a match. The valid values are from 0 - 128.

extDestPortStart — specifies the low-end TCP/UDP destination port value.

extDestPortEnd — specifies the high-end TCP/UDP destination port value.

subName — specifies the username string for this subscriber for recording purposes.

upBwMinKbps — specifies the upstream maximum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed.

downBwMinKbps — specifies the downstream maximum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed.

disableRecreateIfRecoverableErrors — allows the BoD AM retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to true, means that when a BoD AM encounters one of these errors, the retry is not attempted. If this parameter is omitted, BoD AM always tries to recover from any recoverable errors.

upRequiredAttrMask — specifies the upstream required attribute mask.

upForbiddenAttrMask — specifies the upstream forbidden attribute mask.

upAttrAggrRuleMask — specifies the upstream required attribute aggregation rule mask.

downRequiredAttrMask — specifies the downstream required attribute mask.

downForbiddenAttrMask — specifies the downstream forbidden attribute mask.

downAttrAggrRuleMask — specifies the downstream required attribute aggregation rule mask.

downPeakTrafficRate — specifies the downstream peak traffic rate.

upPeakTrafficRate — specifies the upstream peak traffic rate.

modifySession Interface

The modifySession interface allows the modification of the bandwidth and classifier parameters for standard SOAP types.

modifySessionExtension Interface

The modifySessionExtension interface allows the modification of the bandwidth and classifier parameters for extended SOAP types.

modifySessionIPv6 Interface

The modifySessionIPv6 interface allows the modification of the bandwidth and classifier parameters for IPv6 SOAP types.

deleteSession Interface

The deleteSession interface allows an application to delete the QoS resources previously allocated. The parameters of this interface are defined as follows:

```
public String deleteSession(String sessionId) ;
throws RemoteException
```

Where the **`sessionId`** specifies the unique session identifier for the session.

When this operation is successful, a string containing the passed-in sessionId is returned. If this operation is not successful, an error result code is returned in the following format:

```
Error: <error code>
```

For a complete list of error codes, see *BoD Interface Error Codes*.

SOAP Session Extended Classifiers

The parameters of this interface are defined as follows:

```
public String createSessionExtension(String pcmmServiceName, String groupName,

    String sessionId, int upBwKbps, int upBwMaxKbps, int downBwKbps,
    int downBwMaxKbps, int durInMins, long vollimitKbps, String subIP,
    String subMac, String extSubIP, String extSubIpMask, int extSubPortStart,
    int extSubPortEnd, String extDestIP, String extDestIpMask,
    int extDestPortStart, int extDestPortEnd, String subName, int upBwMinKbps,
    int downBwMinKbps, boolean disableRecreateIfRecoverableErrors,
    long upRequiredAttrMask, long upForbiddenAttrMask, long upAttrAggrRuleMask,

    long downRequiredAttrMask, long downForbiddenAttrMask,
   long downAttrAggrRuleMask, long upPeakTrafficRate, long downPeakTrafficRate,
 String cmtsIPAddress) ;
```

Note: The required parameters depend on the service definition (which parameters are tagged as "passed-in").

Where:

pcmmServiceName — specifies the service name, which defines the upstream and downstream traffic profiles. See *Creating a Service* for additional information. Note that the pcmmServiceName is not the PCMM Service Class Name defined within the DOCSIS and PacketCable specifications.

groupName — specifies the caller supplied group name, assigned to the session, for reporting purposes.

sessionId — specifies the caller supplied session identifier associated to the session, or null if no caller supplied identifier is needed.

upBwKbps — specifies the minimum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed.

upBwMaxKbps — specifies the upstream maximum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed.

downBwKbps — specifies the minimum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed.

downBwMaxKbps — specifies the downstream maximum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed.

durInMins — specifies the duration for the session, or 0 for a session of indefinite duration (where the BoD AM is responsible for teardown).

volLimitInKb — specifies the volume limit for the session.

subIp: specifies the subscriber IPv4 address. This identifies the subscriber requesting the QoS service.

subMac -- specifies the subscriber MAC address. The passed in MAC address is translated to an IP address when MAC address translation is enabled. This identifies the subscriber requesting the QoS service.

extSubIp — specifies the source IPv4 subnet address. It can also specify any specific IP address provided "extSubIpMask" value is set to 32.

extSubIpMask — specifies the source subnet mask.

extSubPortStart — specifies the start port no. of the source port range.

extSubPortEnd — specifies the end port no. of the source port range.

extDestIp — specifies the destination IPv4 subnet address. It can also specify any specific ip address provided "extDestIpMask" value is set to 32.

extDestIpMask — specifies the destination subnet mask.

extDestPortStart — specifies the start port no. of the destination port range.

extDestPortEnd — specifies the end port no. of the destination port range.

subName — specifies the username string for this subscriber for recording purposes.

upBwMinKbps — specifies the upstream maximum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed.

downBwMinKbps — specifies the downstream maximum reserved traffic rate, in kbps. A value of -1 indicates that this value is not needed.

disableRecreateIfRecoverableErrors — allows the BoD AM retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to true, when BoD AM encounters one

of these errors, the retry is not attempted. Note, if this parameter is omitted, BoD AM always tries to recover from any recoverable errors.

upRequiredAttrMask — specifies the required attribute mask.

upForbiddenAttrMask — specifies the forbidden attribute mask.

upAttrAggrRuleMask — specifies the required attribute aggregation rule mask.

downRequiredAttrMask — specifies the required attribute mask.

downForbiddenAttrMask — specifies the forbidden attribute mask.

downAttrAggrRuleMask — specifies the attribute aggregation mask.

downPeakTrafficRate — specifies the downstream peak traffic rate.

upPeakTrafficRate — specifies the upstream peak traffic rate.

cmtsIPAddress -- specifies the IP address of CMTS

When this operation is successful, a newly created session id is returned (the caller supplied session identifier if provided, or a system generated session identifier if a caller supplied session id is not provided). If this operation is unsuccessful, an error result code is returned in the following format:

```
Error: <error code>
```

For a complete list of error codes, refer to *BoD Interface Error Codes*.

SOAP Traffic Classifier Interface

The Traffic Classifier Interface is used to add or delete classifiers for a service flow.

addTrafficClassifier interface

The addTrafficClassifier allows an application to add a traffic stream to a flow. The parameters of this interface are defined as follows:

```
public String addTrafficClassifier(String pcmmServiceName, String groupName,
    String subIp, String int subPort, String destIP, int destPort, String
cmtsIPAddress) ;
throws RemoteException
```

Where:

pcmmServiceName — specifies the PCMM service name, which defines the upstream and downstream traffic profiles. See *Creating a Service* for additional information.

groupName — specifies a caller-supplied group name, assigned to the session, for reporting purposes.

subIp — specifies the subscriber IPv4 address. This identifies the user requesting the QoS service.

subPort — specifies the subscriber port. A value of 0 represents a wildcard value, and a -1 value indicates that this argument is not needed.

destIp — specifies the destination IPv4 address.

destPort — specifies the destination port. A value of 0 represents a wildcard value, and a -1 value indicates that this argument is not needed.

cmtsIPAddress -- specifies the IP address of CMTS.

When this operation is successful, the pcmmServiceName is returned. If this operation is unsuccessful, an error result code is returned in the following format:

```
Error: <error code>
```

For a complete list of error codes, see *BoD Interface Error Codes*.

deleteTrafficClassifier Interface

The deleteTrafficClassifier allows an application to de-classify and therefore remove a traffic stream from a QoS flow already in existence. The parameters of this interface are defined as follows:

```
public String deleteTrafficClassifier(String pcmmServiceName, String subIp,
    String int subPort, String destIp, int destPort, String cmtsIPAdress);
throws RemoteException
```

Where:

pcmmServiceName — specifies the PCMM service name, which defines the upstream and downstream traffic profiles.

subIp — specifies the subscriber IPv4 address. This identifies the user requesting the QoS service.

subPort — specifies the subscriber port. A value of 0 represents a wildcard value, and a -1 value indicates that this argument is not needed.

destIp — specifies the destination IPv4 address.

destPort — specifies the destination port. A value of 0 represents a wildcard value, and a -1 value indicates that this argument is not needed.

cmtsIPAddress -- specifies the IP address of CMTS.

When this operation is successful, the pcmmServiceName is returned. If this operation is unsuccessful, an error result code is returned in the following format:

```
Error: <error code>
```

For a complete list of error codes, refer to *BoD Interface Error Codes*.

addTrafficClassifierWithExtClassifier interface

The addTrafficClassifierWithExtClassifier allows an application to add a traffic stream to a flow with additional classifiers. The parameters of this interface are defined as follows:

```
public String addTrafficClassifierWithExtClassifier (String pcmmServiceName,
    String groupName, String subIp, String int extSubPortStart,
    String extDestIp, int extDestPortStart,
    boolean disableRecreateIfRecoverableErrors, String extSubIp,
    String extSubIpMask, String extDestIpMask, int extSubPortEnd,
    int extDestPortEnd, String cmtsIPAddress) ;
```

Where:

pcmmServiceName — specifies the PCMM service name, which defines the upstream and downstream traffic profiles. See *Creating a Service* for additional information.

groupName — specifies a caller-supplied group name, assigned to the session, for reporting purposes.

subIp — specifies the subscriber IPv4 address. This identifies the user requesting the QoS service.

extSubIp — specifies the source IPv4 subnet address.

extSubIpMask — specifies the source IP mask.

extDestIp — specifies the destination IPv4 subnet address.

extDestIpMask — specifies the destination IP mask.

extSubPortStart — specifies the start port address for source port range.

extSubPortEnd — specifies the end port address for source port range.

extDestPortStart — specifies the start port address for destination port range.

extDestPortEnd — specifies the end port address for destination port range.

cmtsIPAddress -- specifies the IP Address of CMTS.

When this operation is successful, the pcmmServiceName is returned. If this operation is unsuccessful, an error result code is returned in the following format:

```
Error: <error code>
```

disableRecreateIfRecoverableErrors — disables the BoD retry logic when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to true, when a BoD AM system encounters one of these errors, the retry is not attempted. If this parameter is omitted, a BoD AM system always tries to recover from any recoverable errors.

For a complete list of error codes, see *BoD Interface Error Codes*.

addTrafficClassifierWithIPv6Classifier interface

The addTrafficClassifierWithIPv6Classifier allows an application to add a traffic stream to a flow when IPv6 classifiers are used. The parameters of this interface are defined as follows:

```
public String addTrafficClassifierWithIPv6Classifier (String serviceName,
    String groupName, String subIP, int extSubIP,
    String extDestIP, int extDestPortStart, boolean
disableRecreateIfRecoverableErrors,
    String extSubIP, short subIpPrefixLength, short destIpPrefixLength,
    int extSubPortEnd, int extDestPortEnd);
```

Where:

serviceName — specifies the start port address for destination port range.

groupName — specifies the caller-supplied group name, assigned to the session, for reporting purposes.

subIP — specifies the subscriber IPv6 address. This identifies the subscriber requesting the QoS service.

extSubPortStart — specifies the low-end TCP/UDP source port value.

extDestIP — specifies the destination IPv6 address.

extDestPortStart — specifies the low-end TCP/UDP destination port value.

disableRecreateIfRecoverableErrors — allows the BoD AM retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to true, when the BoD AM encounters one of these errors, the retry is not attempted. If this parameter is omitted, the BoD AM always tries to recover from any recoverable errors.

extSubIP — specifies the source IPv6 address.

subIpPrefixLength — specifies how many high order bits in the corresponding IPv6 source Address to consider in determining a match. The valid values are from 0 – 128.

destIpPrefixLength — specifies how many high order bits in the corresponding IPv6 destination Address to consider in determining a match. The valid values are from 0 – 128.

extSubPortEnd — specifies the high-end TCP/UDP source port value.

extDestPortEnd — specifies the high-end TCP/UDP destination port value.

deleteTrafficClassifierWithExtClassifer Interface

The deleteTrafficClassifierWithExtClassifier allows an application to de-classify and therefore remove a traffic stream from a QoS flow already in existence. The parameters of this interface are defined as follows:

```
public String deleteTrafficClassifierWithExtClassifier (String pcmmServiceName,

    String subIp, int extSubPortStart, String extDestIp,
    int extDestPortStart, String extSubIp, String extSubIpMask,
    String extDestIpMask, int extSubPortEnd, int extDestPortEnd, String
cmtsIPAddress) ;
```

Where:

pcmmServiceName — specifies the PCMM service name, which defines the upstream and downstream traffic profiles.

subIp — specifies the subscriber IPv4 address. This identifies the user requesting the QoS service.

extSubIp — specifies the source IPv4 subnet address.

extSubIpMask — specifies the source IP mask.

extDestIp — specifies the destination IPv4 subnet address.

extDestIpMask — specifies the destination IP mask.

extSubPortStart — specifies the start port address for source port range.

extSubPortEnd — specifies the end port address for source port range.

extDestPortStart — specifies the start port address for destination port range.

extDestPortEnd — specifies the end port address for destination port range.

cmtsIPAddress -- specifies the IP address of CMTS.

When this operation is successful, the pcmmServiceName is returned. If this operation is unsuccessful, an error result code is returned in the following format:

```
Error: <error code>
```

For a complete list of error codes, see *BoD Interface Error Codes*.

deleteTrafficClassifierWithIPv6Classifier Interface

The deleteTrafficClassifierWithIPv6Classifier interface allows an application to add a traffic stream to a flow when IPv6 classifiers are used. The parameters of this interface are defined as follows:

```
public String deleteTrafficClassifierWithIPv6Classifier (String serviceName,
    String subIP, int extSubPortStart, String extDestIP,
    int extDestPortStart, String extSubIP, short subIpPrefixLength,
    short destIpPrefixLength, int extSubPortEnd, int extDestPortEnd) ;
```

where:

`serviceName` — specifies the start port address for destination port range.

`subIP` — specifies the subscriber IPv6 address. This identifies the user requesting the QoS service.

`extSubPortStart` — specifies the low-end TCP/UDP source port value.

`extDestIP` — specifies the destination IPv6 address.

`extDestPortStart` — specifies the low-end TCP/UDP destination port value.

`extSubIP` — specifies the source IPv6 address.

`subIpPrefixLength` — specifies how many high order bits in the corresponding IPv6 source Address to consider in determining a match. The valid values are from 0 - 128.

`destIpPrefixLength` — specifies how many high order bits in the corresponding IPv6 destination Address to consider in determining a match. The valid values are from 0 - 128.

`extSubPortEnd` — specifies the high-end TCP/UDP source port value.

`exDestPortEnd` — specifies the high-end TCP/UDP destination port value.

SOAP Traffic Multiple Classifier Interface

The Traffic Classifier Interface is used to create SOAP traffic with multiple classifiers for a service flow. To create a SOAP session with multiple classifiers, use this request:

createSessionWithMultipleClassifier interface

The createSessionWithMultipleClassifier allows an application to add a traffic stream to a flow. The parameters of this interface are defined as follows:

```
<xs:element name="createSessionWithMultiClassifiers">
<xs:complexType>
<xs:sequence>
<xs:element name="String_1" type="xs:string"/>  //serviceName
<xs:element name="String_2" type="xs:string"/>  //groupName
<xs:element name="String_3" type="xs:string"/>  //sessionId
<xs:element name="int_4" type="xs:int"/>//upBwKbps
<xs:element name="int_5" type="xs:int"/>//upBwMaxKbps
<xs:element name="int_6" type="xs:int"/>//downBwKbps
<xs:element name="int_7" type="xs:int"/>//downBwMaxKbps
<xs:element name="int_8" type="xs:int"/>//durInMins
<xs:element name="String_9" type="xs:string"/>//subIp
<xs:element name="String_10" type="xs:string"/>//subMac
<xs:element name="String_11" type="xs:string"/>//subPort
<xs:element name="String_12" type="xs:string"/>//serverIp
<xs:element name="String_13" type="xs:string"/>//serverPort
```

```
<xs:element name="String_14" type="xs:string"/>//extSubIP
<xs:element name="String_15" type="xs:string"/>//extSubIpMask
<xs:element name="String_16" type="xs:string"/>//extSubPortStart
<xs:element name="String_17" type="xs:string"/>//extSubPortEnd
<xs:element name="String_18" type="xs:string"/>//extDestIP
<xs:element name="String_19" type="xs:string"/>//extDestIpMask
<xs:element name="String_20" type="xs:string"/>//extDestPortStart
<xs:element name="String_21" type="xs:string"/>//extDestPortEnd
<xs:element name="String_22" type="xs:string"/>//IPv6SubIP
<xs:element name="String_23" type="xs:string"/>//IPv6subIpPrefixLength
<xs:element name="String_24" type="xs:string"/>//IPv6SubPortStart
<xs:element name="String_25" type="xs:string"/>//IPv6SubPortEnd
<xs:element name="String_26" type="xs:string"/>//IPv6DestIP
<xs:element name="String_27" type="xs:string"/>//ipv6DestIpPrefixLength
<xs:element name="String_28" type="xs:string"/>//IPv6DestPortStart
<xs:element name="String_29" type="xs:string"/>//IPv6DestPortEnd
<xs:element name="String_30" type="xs:string"/>//subName
<xs:element name="int_31" type="xs:int"/>//upBwMinKbps
<xs:element name="int_32" type="xs:int"/>//downBwMinKbps
<xs:element name="boolean_33"
type="xs:boolean"/>//disableRecreateIfRecoverableErrors
<xs:element name="long_34" type="xs:long"/>//vollimitKbps
<xs:element name="long_35" type="xs:long"/>//upRequiredAttrMask
<xs:element name="long_36" type="xs:long"/>//upForbiddenAttrMask
<xs:element name="long_37" type="xs:long"/>//upAttrAggrRuleMask
<xs:element name="long_38" type="xs:long"/>//upPeakTrafficRate
<xs:element name="long_39" type="xs:long"/>//downRequiredAttrMask
<xs:element name="long_40" type="xs:long"/>//downForbiddenAttrMask
<xs:element name="long_41" type="xs:long"/>//downAttrAggrRuleMask
<xs:element name="long_42" type="xs:long"/>//downPeakTrafficRate
 <xs:element name="cmtsIPAddress" type="xs:string" maxOccurs="1" minOccurs="0"
/
</xs:sequence>
</xs:complexType>
</xs:element>
```

Where:

`pcmmServiceName` — specifies the PCMM service name, which defines the upstream and downstream traffic profiles. See *Creating a Service* for additional information.

`groupName` — specifies a caller-supplied group name, assigned to the session, for reporting purposes.

`subIp` — specifies the subscriber IPv4 address. This identifies the user requesting the QoS service.

`subPort` — specifies the subscriber port. A value of 0 represents a wildcard value, and a -1 value indicates that this argument is not needed.

`destIp` — specifies the destination IPv4 address.

`destPort` — specifies the destination port. A value of 0 represents a wildcard value, and a -1 value indicates that this argument is not needed.

`cmtsIPAddress` -- specifies the IP address of the CMTS.

When this operation is successful, the pcmmServiceName is returned. If this operation is unsuccessful, an error result code is returned in the following format:

```
Error: <error code>
```

For a complete list of error codes, see *BoD Interface Error Codes*.

HTTP Interface Definitions

The following describes the HTTP interfaces supported by the BoD AM.

HTTP Configuration Interfaces

The HTTP Configuration Interfaces are used to set configuration settings for the BoD application.

setPolicyServerIP.do Interface

The setPolicyServerIP.do interface allows an application to specify the IP address of the Policy Server that receives PCMM requests. Use the following syntax when using this interface.

```
http://<serverip>/bod/setPolicyServerIP.do?PSIP=x.x.x.x
```

Where:

`serverip` — specifies the IP address of the BoD AM.

x.x.x.x — specifies the IP Address of Policy Server.

When this operation is successful the following displays:

```
<BODY>
   <STATUS>success</STATUS> or <STATUS>failed</STATUS>
</BODY>
```

The returned status field displays the success or failure of the requested operation.

The setAmId.do interface allows the Application Manager ID to be set for the BoD application. Note that this may be overridden by an AM ID specified for a particular PCMM service definition. Use the following syntax when using this interface:

```
http://<serverip>/bod/setAmId.do?AMID=x
```

Where:

`serverip` — specifies the IP address of the BoD AM.

x — specifies the new application manager ID to use for the BoD AM.

When this operation is successful the following displays:

```
<BODY>
   <STATUS>success</STATUS> or <STATUS>failed</STATUS>
</BODY>
```

The returned status field displays the success or failure of the requested operation.

HTTP Session Interfaces

For session-oriented applications, such as streaming media, file transfers, and bandwidth on demand, the HTTP Session Interface provides createSession and deleteSession interfaces that allow the calling application to manage the lifecycle of the QoS session appropriate for the application.

Note: HTTP session interfaces can also have multiple classifiers. See *HTTP Session Interfaces with Multiple Classifiers*.

createSession.do Interface

The createSession.do interface allows an application to create a session based on QoS parameters. Use the following syntax when using this interface:

```
http://<serverip>/bod/createSession.do?SERVICENAME=servicename
&GROUPNAME=groupname&SUBIP=x.x.x.x&SUBPORT=n
&DESTIP=y.y.y.y&DESTPORT=m&UPBW=u&UPBWMAX=umax&DOWNBW=d&DOWNBWMAX=dmax&DUR=t
&SUBNAME=user&SSID=s&DISABLERETRYONERROR=<true|false>
&UPRAMASK=uram&UPFAMASK=ufam&UPAARMASK=uaarm &DOWNRAMASK=dram
&DOWNFAMASK=dfam&DOWNAARMASK=daarm&DOWNPTR=dptr&UPPTR=uptr&VOLLIMIT=v&CMTSIP=cmtsip
```

Where:

`serverip` — specifies the IP Address of the BoD AM.

`servicename` — specifies the PCMM service name, which defines the upstream and downstream traffic profiles. Refer to *Creating a Service* for additional information.

`groupname` (optional) — specifies a caller-supplied group name assigned to the session for reporting purposes.

`n` (optional) — specifies the subscriber port to classify traffic on.

`y.y.y.y` — allows the operator to limit the traffic that receives this quality to and from a particular server IPv4 address and is used for the classifier.

`m` (optional) — specifies the destination port to classify traffic on.

`u` (optional) — specifies the upstream bandwidth in kbps.

`umax` (optional) — specifies the upstream maximum reserved bandwidth in kbps.

`d` (optional) — specifies the downstream bandwidth in kbps.

`dmax` (optional) — specifies the downstream maximum reserved bandwidth in kbps.

`t` (optional) — specifies the duration the session should be active. This session is automatically torn down after **`t`** minutes.

`user` (optional) — specifies the username string for this subscriber for recording purposes.

`s` (optional) — specifies the application-supplied session ID. This is automatically generated if not supplied.

`true|false` — allows the BoD retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to **`true`**, when the BoD AM encounters one of these errors, the retry is not attempted. If this parameter is set to **`false`** or omitted, the BoD AM tries to recover from any recoverable errors.

uram -- specifies the specifies the Upstream Required Attribute Mask.

ufam -- specifies the Upstream Forbidden Attribute Mask.

uaarm -- specifies the Upstream Attribute Aggregation Rule Mask.

dram -- specifies the Downstream Required Attribute Mask.

dfam -- specifies the Downstream Forbidden Attribute Mask.

daarm -- specifies the Downstream Attribute AggregationRule Mask.

dptr -- specifies the Downstream Peak Traffic Rate.

uptr -- specifies the Upstream Peak Traffic Rate.

v -- specifies the volume limit for the session.

CMTSIP -- specifies the IP address of the CMTS.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
   <SSID>ssid</SSID>
</BODY>
```

The SSID is used to identify this session is a subsequent deleteSession request.

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
</BODY>
```

The returned status field returns the success or failure of the requested operation. A failed status may be returned if the resource is not available within the access network or if the MSO has created and deployed a policy to reject the request.

modifySession.do Interface

The modifySession.do interface allows an application to modify the session bandwidth and classifier parameters for an active BoD session. Use the following syntax when using this interface:

```
http://<serverip>/BoD/modifySession.do?SSID=sessionid&UPBW=U&UPBWMAX=Umax&
DOWNBW=D&DOWNBWMAX=Dmax&SUBIP=Y.Y.Y.Y&SUBPORT=subport&DESTIP=Z.Z.Z.Z&
DESTPORT=destport
```

Where:

serverip — specifies the IP Address of the BoD server.

sessionid — specifies the unique active session id.

U (optional) — specifies the upstream bandwidth in kbps.

D (optional) — specifies the downstream bandwidth in kbps.

Umax (optional) — specifies the upstream maximum reserved bandwidth in kbps.

Dmax (optional) — specifies the downstream maximum reserved bandwidth in kbps.

`Y.Y.Y.Y` — specifies the source subnet address and is used for the classifier.

`Z.Z.Z.Z` — specifies the destination subnet address and is used for the classifier.

`subport` — specifies the source port number.

`destport` — specifies the destination port number.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
<STATUS>success</STATUS>
<SSID>ssid</SSID>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
<STATUS>failure</STATUS>
<FAILUREREASON>description of the failure reason</FAILUREREASON>
</BODY>
```

Important: If the BoD session service parameters have been set on the CMP GUI, the configured parameters will not be changed even after sending a session modification request.

deleteSession.do Interface

The deleteSession.do interface allows an application to remove the QoS resources for a session previously created with createSession.do. Use the following syntax when using this interface:

```
http://<serverip>/bod/deleteSession.do?SSID=s
```

Where:

`serverip` — specifies the IP Address of the BoD AM.

`s` — specifies the SSID returned from create request.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
 </BODY>
```

A failure status is displayed if the session could not be torn down.

HTTP Session Interfaces with Multiple Classifiers

For session-oriented applications, such as streaming media, file transfers, and bandwidth on demand, the HTTP Session Interface provides createSession and deleteSession interfaces that allow the calling application to manage the life cycle of the QoS session appropriate for the application.

Note: Parameter values used for multiple Classifiers are formatted and interpreted similarly to the single classifier createSession.do. However, parameters related to the Classifier(s) for the service will include multiple values, separated by a pipe ("|") character, and ordered corresponding to the Index values in the Service Definition. If a given parameter is not required for one Classifier in a multiple-Classifier Service, then the corresponding parameter should be omitted but the pipe character is still required. Or the corresponding parameter should be populated as -1.

createSession.do Interface for multiple standard classifiers

The createSession.do interface allows for an application to use a mixed type of interface with multi-Standard Classifiers to create a session based on QoS parameters. Use the following syntax when using this interface:

```
http://bod_addr/bod/createSession.do?http://bod_addr/bod/createSession.do?SERVICENAME=
MultiStandardService&DUR=120&SUBIP=10.0.8.11|10.0.8.12&DESTIP=20.0.8.11|20.0.8.12&SUBPORT=10241|10242&
DESTPORT=20241|20242&CMTSIP=cmtsip
```

Note: In addition to the basic syntax and parameters used for multiple standard classifiers, these parameters can be used:

```
UPBW=u&UPBWMAX=umax&DOWNBW=d&DOWNBWMAX=dmax&DUR=t
&SUBNAME=user&SSID=s&DISABLERETRYONERROR=<true|false>
&UPRAMASK=uram&UPFAMASK=ufam&UPAARMASK=uaarm &DOWNRAMASK=dram
&DOWNFAMASK=dfam&DOWNAARMASK=daarm&DOWNPTR=dptr&UPPTR=uptr&VOLLIMIT=v
```

Where:

`servicename` — (multiStandard) specifies the PCMM service name for multiple standard classifiers, which defines the upstream and downstream traffic profiles. Refer to *Creating a Service* for additional information.

`1.1.1.1` and **`2.2.2.2`** — 2 will be as sourceip parameters to be set to Classifiers and only 1.1.1.1 (Subip[0])will be as SUBip in BodSession. Subip,Destip,Subport and Destport are configured as pass-in parameters in Standard Classifier.

`umax` (optional) — specifies the upstream maximum reserved bandwidth in kbps.

`d` (optional) — specifies the downstream bandwidth in kbps.

`dmax` (optional) — specifies the downstream maximum reserved bandwidth in kbps.

`t` (optional) — specifies the duration the session should be active. This session is automatically torn down after **`t`** minutes.

`uaarm` -- specifies the Upstream Attribute Aggregation Rule Mask.

`user` (optional) — specifies the username string for this subscriber for recording purposes.

`s` (optional) — specifies the application-supplied session ID. This is automatically generated if not supplied.

`true|false` — allows the BoD retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to **`true`**, when the BoD AM encounters one of these errors, the retry is not attempted. If this parameter is set to **`false`** or omitted, the BoD AM tries to recover from any recoverable errors.

`uram` -- specifies the specifies the Upstream Required Attribute Mask.

`ufam` -- specifies the Upstream Forbidden Attribute Mask.

`dram` -- specifies the Downstream Required Attribute Mask.

`dfam` -- specifies the Downstream Forbidden Attribute Mask.

`daarm` -- specifies the Downstream Attribute AggregationRule Mask.

`dptr` -- specifies the Downstream Peak Traffic Rate.

`uptr` -- specifies the Upstream Peak Traffic Rate.

`v` -- specifies the volume limit for the session.

`cmtsip` -- specifies the IP address of CMTS.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
   <SSID>ssid</SSID>
</BODY>
```

The SSID is used to identify this session is a subsequent deleteSession request.

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
</BODY>
```

The returned status field returns the success or failure of the requested operation. A failed status may be returned if the resource is not available within the access network or if the MSO has created and deployed a policy to reject the request.

modifySession.do Interface

The modifySession.do interface allows an application to modify the session bandwidth and classifier parameters for an active BoD session.

Note: This function does not support multiple classifiers.

Use the following syntax when using this interface:

```
http://<serverip>/BoD/modifySession.do?SSID=sessionid&UPBW=U&UPBWMAX=Umax&
DOWNBW=D&DOWNBWMAX=Dmax&SUBIP=Y.Y.Y.Y&SUBPORT=subport&DESTIP=Z.Z.Z.Z&
DESTPORT=destport
```

Where:

`serverip` — specifies the IP Address of the BoD server.

`sessionid` — specifies the unique active session id.

U (optional) — specifies the upstream bandwidth in kbps.

D (optional) — specifies the downstream bandwidth in kbps.

Umax (optional) — specifies the upstream maximum reserved bandwidth in kbps.

Dmax (optional) — specifies the downstream maximum reserved bandwidth in kbps.

Y.Y.Y.Y — specifies the source subnet address and is used for the classifier.

Z.Z.Z.Z — specifies the destination subnet address and is used for the classifier.

subport — specifies the source port number.

destport — specifies the destination port number.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
<STATUS>success</STATUS>
<SSID>ssid</SSID>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
<STATUS>failure</STATUS>
<FAILUREREASON>description of the failure reason</FAILUREREASON>
</BODY>
```

Important: If the BoD session service parameters have been set on the CMP GUI, the configured parameters will not be changed even after sending a session modification request.

deleteSession.do Interface

The deleteSession.do interface allows an application to remove the QoS resources for a session previously created with createSession.do. Use the following syntax when using this interface:

```
http://<serverip>/bod/deleteSession.do?SSID=s
```

Where:

serverip — specifies the IP Address of the BoD AM.

s — specifies the SSID returned from create request.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
 </BODY>
```

A failure status is displayed if the session could not be torn down.

HTTP Session Extended Classifiers

createSession.do Extended Interface

Use the following syntax when using this interface:

```
http://<serverip>/bod/createSession.do?SERVICENAME=servicename
&GROUPNAME=groupname&SUBIP=x.x.x.x&EXTSUBIP=y.y.y.y
&EXTSUBMASK=submask&EXTSUBPORTSTART=spstart&EXTSUBPORTEND=spend&EXTDESTIP=z.z.z.z
&EXTDESTMASK=destmask&EXTDESTPORTSTART=dpstart&EXTDESTPORTEND=dpend&UPBW=u
&UPBWMAX=umax&DOWNBW=d&DOWNBWMAX=dmax&DUR=t&SUBNAME=user&SSID=s&VOLLIMIT=v
&DISABLERETRYONERROR=<true|false>&UPRAMASK=uram&UPFAMASK=ufam&UPAARMASK=uaarm
&DOWNPTR=dptr&DOWNRAMASK=dram&DOWNFAMASK=dfam&DOWNAARMASK=daarm&UPPTR=upptr&CMTSIP=cmtsip
```

Note: The required parameters depend on the service definition (which parameters are tagged as "passed-in").

Where:

`serverip` — specifies the IP address of the BoD AM.

`servicename` — specifies the PCMM service name, which defines the upstream and downstream traffic profiles. See *Creating a Service* for additional information.

`groupname` (optional) — specifies a caller-supplied group name assigned to the session for reporting purposes.

`y.y.y.y` — specifies the source subnet address and is used for the classifier. It can also specify any specific IPv4 address provided "subMask" value is set to 32.

`submask` — specifies the IP mask used with source subnet.

`spstart` — specifies the start port number for the source port range.

`spend` — specifies the end port number for the source port range.

`z.z.z.z` — specifies destination subnet address and is used for the classifier. It can also specify any specific IPv4 address provided "destmask" value is set to 32.

`destmask` — specifies the IP mask used with destination subnet.

`dpstart` — specifies the start port number for the destination port range.

`dpend` — specifies the end port number for the destination port range.

`u` (optional) — specifies the upstream bandwidth in kbps.

`umax` (optional) — specifies the upstream maximum reserved bandwidth in kbps.

`d` (optional) — specifies the downstream bandwidth in kbps.

`dmax` (optional) — specifies the downstream maximum reserved bandwidth in kbps.

`t` (optional) — specifies the duration the session should be active. This session is automatically torn down after T minutes.

`user` (optional) — specifies the username string for this subscriber for recording purposes.

s (optional) — specifies the application supplied session ID. This is automatically generated if not supplied.

v (optional) — specifies the volume limit for the session. (This is not supported yet and should be supported in next release.)

true|false — allows the BoD retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a pcmm error code 127 and subcode 211). If this value is set to **true**, when the BoD AM encounters one of these errors, the retry is not attempted. If this parameter is set to **false** or omitted, the BoD AM tries to recover from any recoverable errors.

uram — specifies the Upstream Required Attribute Mask.

ufam — specifies the Upstream Forbidden Attribute Mask.

uaarm — specifies the Upstream Attribute Aggregation Rule Mask.

dptr — specifies the Downstream Peak Traffic Rate.

dram — specifies the Downstream Required Attribute Mask.

dfam — specifies the Downstream Forbidden Attribute Mask.

daarm — specifies the Downstream Attribute Aggregation Rule Mask.

upptr — specifies the Upstream Peak Data Rate.

cmtsip -- specifies the IP address of CMTS.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
   <SSID>ssid</SSID>
</BODY>
```

The SSID is used to identify this session is a subsequent deleteSession request.

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
</BODY>
```

The returned status field returns the success or failure of the requested operation. A failed status may be returned if the resource is not available within the access network or if the MSO has created and deployed a policy to reject the request.

createSession.do Extended Interface with Multiple Classifiers

Use the following syntax when using this interface:

```
http://<serverip>/bod/createSession.do?SERVICENAME=multiExtended&GROUPNAME=groupname&
SUBIP=X.X.X.X&EXTSUBIP=extsubip1|extsubip2&EXTSUBIPMASK=subMask1|subMask2&EXTSUBPORTSTART
=SPStart1|SPStart2&EXTSUBPORTEND=SPEnd1|
SPEnd2&EXTDESTIP=extdestip1|extdestip2&EXTDESTIPMASK=destMask1|
destMask2&EXTDESTPORTSTART=DPStart1|DPStart2&EXTDESTPORTEND=DPEnd1|DPEnd2&UPBW=U&UPBWMAX
=Umax&DOWNBW=D&DOWNBWMAX=Dmax&DUR=50&CMTSIP=cmtsip
```

Where:

X.X.X.X is needed as subscribe ip in bodsession while Y.Y.Y.Y and Y1.Y1.Y1.Y1 are as sourceip parameters set in Classifiers. So do otheCr parameters.

modifySession.do Extended Interface

The modifySession.do interface allows an application to modify the extended classifier parameters for an active BoD session. Use the following syntax when using this interface:

```
http://<serverip>/BoD/modifySession.do?SSID=sessionid&EXTSUBIP=Y.Y.Y.Y&EXTSUBMASK=
subMask&EXTSUBPORTSTART=SPStart&EXTSUBPORTEND=SPEnd&EXTDESTIP=Z.Z.Z.Z&EXTDESTMASK=
destMask&EXTDESTPORTSTART=DPStart&EXTDESTPORTEND=DPEnd&UPBW=U&UPBWMAX=Umax&DOWNBW=D&DOWNBWMAX=Dmax
```

Where:

serverip — specifies the IP Address of the BoD AM.

sessionid — specifies the unique active session ID.

Y.Y.Y.Y — specifies the source subnet address and is used for the classifier. It can also specify any specific IP address if the **subMask** value is set to 32.

subMask — specifies the IP mask used with source subnet.

Z.Z.Z.Z — specifies destination subnet address and is used for the classifier. It can also specify any specific IP address if the **destMask** value is set to 32.

destMask — specifies the IP mask used with destination subnet.

SPStart — specifies the start port number for the source port range.

SPEnd — specifies the end port number for the source port range.

DPStart — specifies the start port number for the destination port range.

DPEnd — specifies the end port number for the destination port range.

U (optional) — specifies the upstream bandwidth in kbps.

D (optional) — specifies the downstream bandwidth in kbps.

Umax (optional) — specifies the upstream maximum reserved bandwidth in kbps.

Dmax (optional) — specifies the downstream maximum reserved bandwidth in kbps.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
<STATUS>success</STATUS>
<SSID>ssid</SSID>
</BODY>
```

When this option is not successful the following is returned as the HTTP response:

```
<BODY>
<STATUS>failure</STATUS>
<FAILUREREASON>description of the failure reason</FAILUREREASON>
</BODY>
```

Important: If the BoD session service bandwidth and classifier parameters have been set on the CMP GUI, the configured parameters will not be changed even after sending a session modification request.

deleteSession.do Interface

The deleteSession.do interface allows an application to remove the QoS resources for a session previously created with createSession.do. Use the following syntax when using this interface:

```
http://<serverip>/bod/deleteSession.do?SSID=s
```

Where:

`serverip` — specifies the IP Address of the BoD AM.

`s` — specifies the SSID returned from create request.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
 </BODY>
```

A failure status is displayed if the session could not be torn down.

HTTP Session with Multiple Extended Classifiers

createSession.do Multiple Extended Interface

Use the following syntax when using this interface:

Note: Parameter values used for multiple Classifiers are formatted and interpreted similarly to the single classifier createSession.do. However, parameters related to the Classfier(s) for the service will include multiple values, separated by a pipe ("|") character, and ordered corresponding to the Index values in the Service Definition. If a given parameter is not required for one Classifier in a multiple-Classifier Service, then the corresponding parameter should be omitted but the pipe character is still required. If a given parameter is not required for one Classifier in a multiple-Classifier Service, then the corresponding parameter should be omitted but the pipe character is still required. Or the corresponding parameter should be populated as -1.

```
http://<serverip>/bod/createSession.do?SERVICENAME=multiExtended&GROUPNAME=groupname
&SUBIP=X.X.X.X&EXTSUBIP=Y.Y.Y.Y|Y1.Y1.Y1.Y1&EXTSUBMASK=subMask1|subMask2&EXTSUBPORTSTART
=SPStart1|SPStart2&EXTSUBPORTEND=SPEnd1|SPEnd2&EXTDESTIP=Z.Z.Z.Z|Z1.Z1.Z1.Z1&EXTDESTMASK=destMask1|

destMask2&EXTDESTPORTSTART=DPStart1|DPStart2&EXTDESTPORTEND=DPEnd1|DPEnd2&UPBW=U&UPBWMAX=Umax&DOWNBW=
D&DOWNBWMAX=Dmax&DUR=50&CMTSIP=cmtsip
```

Note: In addition to the basic syntax and parameters used for multiple standard classifiers, these parameters can be used:

```
UPBW=u&UPBWMAX=umax&DOWNBW=d&DOWNBWMAX=dmax&DUR=t
&SUBNAME=user&SSID=s&DISABLERETRYONERROR=<true|false>
&UPRAMASK=uram&UPFAMASK=ufam&UPAARMASK=uaarm &DOWNRAMASK=dram
&DOWNFAMASK=dfam&DOWNAARMASK=daarm&DOWNPTR=dptr&UPPTR=uptr&VOLLIMIT=v
```

Where:

`x:x:x:x` — is needed as the subscribe ip in bodsession while .

`y:y:y:y` — is used as the sourceip parameter set in the Classifiers.

`y1:y1:y1:y1` — is used as the sourceip parameter set in the Classifiers.

`umax` (optional) — specifies the upstream maximum reserved bandwidth in kbps.

`d` (optional) — specifies the downstream bandwidth in kbps.

`dmax` (optional) — specifies the downstream maximum reserved bandwidth in kbps.

`t` (optional) — specifies the duration the session should be active. This session is automatically torn down after **`t`** minutes.

`uaarm` -- specifies the Upstream Attribute Aggregation Rule Mask.

`user` (optional) — specifies the username string for this subscriber for recording purposes.

`s` (optional) — specifies the application-supplied session ID. This is automatically generated if not supplied.

`true|false` — allows the BoD retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to **`true`**, when the BoD AM encounters one of these errors, the retry is not attempted. If this parameter is set to **`false`** or omitted, the BoD AM tries to recover from any recoverable errors.

`uram` -- specifies the specifies the Upstream Required Attribute Mask.

`ufam` -- specifies the Upstream Forbidden Attribute Mask.

`dram` -- specifies the Downstream Required Attribute Mask.

`dfam` -- specifies the Downstream Forbidden Attribute Mask.

`daarm` -- specifies the Downstream Attribute AggregationRule Mask.

`dptr` -- specifies the Downstream Peak Traffic Rate.

`uptr` -- specifies the Upstream Peak Traffic Rate.

`v` -- specifies the volume limit for the session.

`cmtsip` -- specifies the IP address of CMTS.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
   <SSID>ssid</SSID>
</BODY>
```

The SSID is used to identify this session is a subsequent deleteSession request.

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
</BODY>
```

The returned status field returns the success or failure of the requested operation. A failed status may be returned if the resource is not available within the access network or if the MSO has created and deployed a policy to reject the request.

createSession.do IPv6 Interface with multiple classifiers

Use the following syntax when using this interface:

```
http://<serverip>/bod/createSession.do?SERVICENAME=multiIPv6&GROUPNAME=groupname&
SUBIP=X.X.X.X&IPV6SUBIP=ipv6subip1|ipv6subip2&IPV6SUBPREFIXLENGTH=subprefixlength1|
subprefixlength2& IPV6SUBPORTSTART =ipv6subportstart1|
ipv6subportstart2&IPV6SUBPORTEND=
ipv6subportend1|ipv6subportend2&IPV6DESTIP=ipv6destip1|ipv6destip2&IPV6DESTPREFIXLENGTH=
destprefixlength1|destprefixlength2&IPV6DESTPORTSTART=destportstart1|kkkkk
destportstart2k&IPV6DESTPORTEND= destportend1| destportend2
```

Where:

X.X.X.X is needed as subscribe ip in bodsession while ipv6subip1 and ipv6subip2 are as sourceip parameters which set in ipv6 Classifiers. So do other parameters.

modifySession.do IPv6 Interface

The modifySession.do interface allows an application to modify the session IPv6 classifier parameters.

Note: This function does not support multiple classifiers.

Use the following syntax when using this interface:

```
http://<serverip>/BoD/modifySession.do?SSID=sessionid&IPV6SUBIP=Y.Y.Y.Y&
IPV6SUBPREFIXLENGTH=SUBPREFIXLENGTH&IPV6SUBPORTSTART=SPStart&
IPV6SUBPORTEND=SPEnd&IPV6DESTIP=Z.Z.Z.Z&IPV6DESTPREFIXLENGTH=
DESTPREFIXLENGTH&IPV6DESTPORTSTART=DPStart&IPV6DESTPORTEND=DPEnd
&UPBW=U&UPBWMAX=Umax&DOWNBW=D&DOWNBWMAX=Dmax
```

Where:

`serverip` — specifies the IP Address of the BoD server.

`sessionid` — specifies the unique active session id.

`Y.Y.Y.Y` — specifies the IPV6 source subnet address and is used for the classifier.

`SUBPREFIXLENGTH` — specifies the prefix length of the IPV6 source subnet.

`Z.Z.Z.Z` — specifies IPV6 destination subnet address and is used for the classifier.

`DESTPREFIXLENGTH` — specifies the prefixlength of IPV6 destination subnet.

`SPStart` — specifies the start port number for the IPV6 source port range.

`SPEnd` — specifies the end port number for the IPV6 source port range.

DPStart — specifies the start port number for the IPV6 destination port range.

DPEnd — specifies the end port number for the IPV6 destination port range.

U (optional) — specifies the upstream bandwidth in kbps.

D (optional) — specifies the downstream bandwidth in kbps.

Umax (optional) — specifies the upstream maximum reserved bandwidth in kbps.

Dmax (optional) — specifies the downstream maximum reserved bandwidth in kbps.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
<STATUS>success</STATUS>
<SSID>ssid</SSID>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
<STATUS>failure</STATUS>
<FAILUREREASON>description of the failure reason</FAILUREREASON>
</BODY>
```

Important: If the BoD session service bandwidth and classifier parameters have been set on the CMP GUI, then the configured parameters will not be changed even after sending a session modification request.

deleteSession.do Interface

The deleteSession.do interface allows an application to remove the QoS resources for a session previously created with createSession.do. Use the following syntax when using this interface:

```
http://<serverip>/bod/deleteSession.do?SSID=s
```

Where:

serverip — specifies the IP Address of the BoD AM.

s — specifies the SSID returned from the create request.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
 </BODY>
```

A failure status is displayed if the session could not be torn down.

HTTP Session with Multiple Extended and IPv6 Classifiers

createSession.do IPv6 Interface

Use the following syntax when using this interface:

Note: Parameter values used for multiple Classifiers are formatted and interpreted similarly to the single classifier createSession.do. However, parameters related to the Classfier(s) for the service will include multiple values, separated by a pipe ("|") character, and ordered corresponding to the Index values in the Service Definition. If a given parameter is not required for one Classifier in a multiple-Classifier Service, then the corresponding parameter should be omitted but the pipe character is still required. If a given parameter is not required for one Classifier in a multiple-Classifier Service, then the corresponding parameter should be omitted but the pipe character is still required. Or the corresponding parameter should be populated as -1.

```
http://<serverip>/bod/createSession.do?SERVICENAME=multiExtendedIPv6&GROUPNAME=groupname
&SUBIP=X.X.X.X&IPV6SUBIP=ipv6subip1|ipv6subip2||&IPV6SUBPREFIXLENGTH=prfixlength1|

prfixlength2||& IPV6SUBPORTSTART
=ipv6subport1|ipv6subport2||&IPV6SUBPORTEND=ipv6subportend1|
ipv6subportend2||&IPV6DESTIP=ipv6destip1|ipv6destip2||&IPV6DESTPREFIXLENGTH=destprefixlength1|
destprefixlength2||&IPV6DESTPORTSTART=ipv6destportstart1|ipv6destportstart2||&IPV6DESTPORTEND=
ipv6destportend1|ipv6destportend2||&EXTSUBIP=||extsubip3|extsubip4&EXTSUBMASK=||subMask3|
subMask4&EXTSUBPORTSTART=||SPStart3|SPStart4&EXTSUBPORTEND=||SPEnd3|SPEnd4&EXTDESTIP=||extdestip3|
extdestip4&EXTDESTMASK=||destMask3|destMask4&EXTDESTPORTSTART=||DPStart3|DPStart4&EXTDESTPORTEND=||
DPEnd3| DPEnd4&CMTSIP=cmtsip
```

Note: In addition to the basic syntax and parameters used for multiple standard classifiers, these parameters can be used:

```
UPBW=u&UPBWMAX=umax&DOWNBW=d&DOWNBWMAX=dmax&DUR=t
&SUBNAME=user&SSID=s&DISABLERETRYONERROR=<true|false>
&UPRAMASK=uram&UPFAMASK=ufam&UPAARMASK=uaarm &DOWNRAMASK=dram
&DOWNFAMASK=dfam&DOWNAARMASK=daarm&DOWNPTR=dptr&UPPTR=uptr&VOLLIMIT=v
```

Where:

The service multiExtendedIPv6 has four Classifiers including two ipv6 Classifiers and two extended Classifiers. The first and second Classifiers are IPv6 Classifiers while the third and the fourth are extended Classifiers.

```
IPV6SUBIP= ipv6subip1|ipv6subip2||
EXTSUBIP=|| extsubip3| extsubip4
```

The IPV6SUBIP field has four parameters, ipv6subip1 andipv6subip2will be validated and set to Classifiers according to the first and second IPv6 Classifiers' required vars. The last two parameters are empty strings "".

The EXTSUBIP field also four parameters, the first and second parameters are all "", the last two parameters extsubip3and extsubip4 will be validated and set to Classifiers according to the last two Extended Classifiers' required vars.

Note: The sequence of each field should match the Classifiers' sequence. For example, if a PCMMService has 4 different Standard Classifiers, each of field should like **`Subip`** field: SUBIP= 10.60.2.2| 10.5.5.5| 10.20.2.2| 2.2.2.2

10.60.2.2 will be validated according to the 1st Classifier's required parameter and set to the first Classifier.

10.5.5.5 will be validated according to the 2nd Classifier 's required parameter and set to the second Classifier.

10.20.2.2 will be validated according to the 3rd Classifier's required parameter and set to the third Classifier.

2.2.2.2 will be validated according to the 4th Classifier's required parameter and set to the fourth Classifier.

Of course, the last two parameters can be set as other values and will not take effect.

`serverip` — specifies the IP address of the BoD AM.

`servicename` — specifies the PCMM service name, which defines the upstream and downstream traffic profiles. See *Creating a Service* for additional information.

`groupname` (optional) — specifies a caller-supplied group name assigned to the session for reporting purposes.

`x:x:x:x:x:x:x:x` — specifies the subscriber IPv6 address, which is mutually exclusive with SUBMAC.

`zz:zz:zz:zz:zz:zz` — specifies the subscriber MAC address. The passed-in MAC address is translated to an IP address when MAC address translation has been enabled. This identifies the user requesting the QoS service. If MAC address translation is not enabled, the subIp parameter should be specified. See *HTTP Configuration Interfaces* for further information.

`u` (optional) — specifies the upstream bandwidth in kbps.

`umax` (optional) — specifies the upstream maximum reserved bandwidth in kbps.

`d` (optional) — specifies the downstream bandwidth in kbps.

`dmax` (optional) — specifies the downstream maximum reserved bandwidth in kbps.

`t` (optional) — specifies the duration the session should be active. This session is automatically torn down after T minutes.

`user` (optional) — specifies the username string for this subscriber for recording purposes.

`s` (optional) — specifies the application supplied session ID. This is automatically generated if not supplied.

`v` (optional) — specifies the volume limit for the session.

`true|false` — allows the BoD retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to **`true`**, when the BoD AM encounters one of these errors, the retry is not attempted. If this parameter is set to **`false`** or omitted, the BoD AM tries to recover from any recoverable errors.

`y:y:y:y:y:y:y:y` — specifies the destination IPv6 address.

`ss6` — specifies the low-end TCP/UDP source port value.

`se6` — specifies the high-end TCP/UDP source port value.

`ds6` — specifies the low-end TCP/UDP destination port value.

`de6` — specifies the high-end TCP/UDP destination port value.

sp6 — specifies how many high-order bits in the corresponding IPv6 subscriber address to consider in determining a match. The valid values are from 0 – 128.

dp6 — specifies how many high order bits in the corresponding IPv6 destinationAddress to consider in determining a match. The valid values are from 0 – 128.

uram — specifies the Upstream Required Attribute Mask.

ufam — specifies the Upstream Forbidden Attribute Mask.

uaarm — specifies the Upstream Attribute Aggregation Rule Mask.

dptr — specifies the Downstream Peak Traffic Rate.

dram — specifies the Downstream Required Attribute Mask.

dfam — specifies the Downstream Forbidden Attribute Mask.

daarm — specifies the Downstream Attribute Aggregation Rule Mask.

upptr — specifies the Upstream Peak Data Rate.

cmstip -- specifies the IP address of CMTS.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
   <SSID>ssid</SSID>
</BODY>
```

The SSID is used to identify this session is a subsequent deleteSession request.

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
</BODY>
```

The returned status field returns the success or failure of the requested operation. A failed status may be returned if the resource is not available within the access network or if the MSO has created and deployed a policy to reject the request.

createSession.do IPv6 Interface with multiple classifiers

Use the following syntax when using this interface:

```
http://<serverip>/bod/createSession.do?SERVICENAME=multiIPv6&GROUPNAME=groupname&
SUBIP=X.X.X.X&IPV6SUBIP=ipv6subip1|ipv6subip2&IPV6SUBPREFIXLENGTH=subprefixlength1|
subprefixlength2& IPV6SUBPORTSTART =ipv6subportstart1|
ipv6subportstart2&IPV6SUBPORTEND=
ipv6subportend1|ipv6subportend2&IPV6DESTIP=ipv6destip1|ipv6destip2&IPV6DESTPREFIXLENGTH=
destprefixlength1|destprefixlength2&IPV6DESTPORTSTART=destportstart1|kkkkk
destportstart2k&IPV6DESTPORTEND= destportend1| destportend2&CMTSIP=cmtsip
```

Where:

X.X.X.X is needed as subscribe ip in bodsession while ipv6subip1 and ipv6subip2 are as sourceip parameters which set in ipv6 Classifiers. So do other parameters.

modifySession.do IPv6 Interface

The modifySession.do interface allows an application to modify the session IPv6 classifier parameters.

Note: This function does not support multiple classifiers.

Use the following syntax when using this interface:

```
http://<serverip>/BoD/modifySession.do?SSID=sessionid&IPV6SUBIP=Y.Y.Y.Y&
IPV6SUBPREFIXLENGTH=SUBPREFIXLENGTH&IPV6SUBPORTSTART=SPStart&
IPV6SUBPORTEND=SPEnd&IPV6DESTIP=Z.Z.Z.Z&IPV6DESTPREFIXLENGTH=
DESTPREFIXLENGTH&IPV6DESTPORTSTART=DPStart&IPV6DESTPORTEND=DPEnd
&UPBW=U&UPBWMAX=Umax&DOWNBW=D&DOWNBWMAX=Dmax
```

Where:

`serverip` — specifies the IP Address of the BoD server.

`sessionid` — specifies the unique active session id.

`Y.Y.Y.Y` — specifies the IPV6 source subnet address and is used for the classifier.

`SUBPREFIXLENGTH` — specifies the prefix length of the IPV6 source subnet.

`Z.Z.Z.Z` — specifies IPV6 destination subnet address and is used for the classifier.

`DESTPREFIXLENGTH` — specifies the prefixlength of IPV6 destination subnet.

`SPStart` — specifies the start port number for the IPV6 source port range.

`SPEnd` — specifies the end port number for the IPV6 source port range.

`DPStart` — specifies the start port number for the IPV6 destination port range.

`DPEnd` — specifies the end port number for the IPV6 destination port range.

`U` (optional) — specifies the upstream bandwidth in kbps.

`D` (optional) — specifies the downstream bandwidth in kbps.

`Umax` (optional) — specifies the upstream maximum reserved bandwidth in kbps.

`Dmax` (optional) — specifies the downstream maximum reserved bandwidth in kbps.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
<STATUS>success</STATUS>
<SSID>ssid</SSID>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
<STATUS>failure</STATUS>
<FAILUREREASON>description of the failure reason</FAILUREREASON>
</BODY>
```

Important: If the BoD session service bandwidth and classifier parameters have been set on the CMP GUI, then the configured parameters will not be changed even after sending a session modification request.

deleteSession.do Interface

The deleteSession.do interface allows an application to remove the QoS resources for a session previously created with createSession.do. Use the following syntax when using this interface:

```
http://<serverip>/bod/deleteSession.do?SSID=s
```

Where:

serverip — specifies the IP Address of the BoD AM.

s — specifies the SSID returned from create request.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
 </BODY>
```

A failure status is displayed if the session could not be torn down.

HTTP Session IPv6 Classifiers

createSession.do IPv6 Interface

Use the following syntax when using this interface:

```
http://<serverip>/bod/createSession.do?SERVICENAME=servicename
&GROUPNAME=groupname&SUBIP=x:x:x:x:x:x:x:x&SUBMAC=zz:zz:zz:zz:zz:zz&UPBW=u
&UPBWMAX=umax&DOWNBW=d&DOWNBWMAX=dmax&DUR=t&SUBNAME=user&SSID=s&VOLLIMIT=v
&DISABLERETRYONERROR=<true|false>&IPV6SUBIP=x:x:x:x:x:x:x:x
&IPV6DESTIP=y:y:y:y:y:y:y:y&IPV6SUBPORTSTART=ss6&IPV6SUBPORTEND=se6
&IPV6DESTPORTSTART=ds6&IPV6DESTPORTEND=de6&IPV6SUBPREFIXLENGTH=sp6
&IPV6DESTPREFIXLENGTH=dp6&UPRAMASK=uram&UPFAMASK=ufam&UPAARMASK=uaarm
&DOWNPTR=dptr&DOWNRAMASK=dram&DOWNFAMASK=dfam&DOWNAARMASK=daarm&UPPTR=upptr"
```

Note: The required parameters depend on the service definition (which parameters are tagged as "passed-in").

Where:

serverip — specifies the IP address of the BoD AM.

servicename — specifies the PCMM service name, which defines the upstream and downstream traffic profiles. See *Creating a Service* for additional information.

groupname (optional) — specifies a caller-supplied group name assigned to the session for reporting purposes.

x:x:x:x:x:x:x:x — specifies the subscriber IPv6 address, which is mutually exclusive with SUBMAC.

zz:zz:zz:zz:zz:zz — specifies the subscriber MAC address. The passed-in MAC address is translated to an IP address when MAC address translation has been enabled. This identifies the user requesting the QoS service. If MAC address translation is not enabled, the subIp parameter should be specified. See *HTTP Configuration Interfaces* for further information.

u (optional) — specifies the upstream bandwidth in kbps.

umax (optional) — specifies the upstream maximum reserved bandwidth in kbps.

d (optional) — specifies the downstream bandwidth in kbps.

dmax (optional) — specifies the downstream maximum reserved bandwidth in kbps.

t (optional) — specifies the duration the session should be active. This session is automatically torn down after T minutes.

user (optional) — specifies the username string for this subscriber for recording purposes.

s (optional) — specifies the application supplied session ID. This is automatically generated if not supplied.

v (optional) — specifies the volume limit for the session. (This is not supported yet and should be supported in next release.)

true|false — allows the BoD retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127and subcode 211). If this value is set to **true**, when the BoD AM encounters one of these errors, the retry is not attempted. If this parameter is set to **false** or omitted, the BoD AM tries to recover from any recoverable errors.

y:y:y:y:y:y:y:y — specifies the destination IPv6 address.

ss6 — specifies the low-end TCP/UDP source port value.

se6 — specifies the high-end TCP/UDP source port value.

ds6 — specifies the low-end TCP/UDP destination port value.

de6 — specifies the high-end TCP/UDP destination port value.

sp6 — specifies how many high-order bits in the corresponding IPv6 subscriber address to consider in determining a match. The valid values are from 0 – 128.

dp6 — specifies how many high order bits in the corresponding IPv6 destinationAddress to consider in determining a match. The valid values are from 0 – 128.

uram — specifies the Upstream Required Attribute Mask.

ufam — specifies the Upstream Forbidden Attribute Mask.

uaarm — specifies the Upstream Attribute Aggregation Rule Mask.

dptr — specifies the Downstream Peak Traffic Rate.

dram — specifies the Downstream Required Attribute Mask.

dfam — specifies the Downstream Forbidden Attribute Mask.

daarm — specifies the Downstream Attribute Aggregation Rule Mask.

upptr — specifies the Upstream Peak Data Rate.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
   <SSID>ssid</SSID>
</BODY>
```

The SSID is used to identify this session is a subsequent deleteSession request.

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
</BODY>
```

The returned status field returns the success or failure of the requested operation. A failed status may be returned if the resource is not available within the access network or if the MSO has created and deployed a policy to reject the request.

createSession.do IPv6 Interface with multiple classifiers

Use the following syntax when using this interface:

```
http://<serverip>/bod/createSession.do?SERVICENAME=multiIPv6&GROUPNAME=groupname&
SUBIP=X.X.X.X&IPV6SUBIP=ipv6subip1|ipv6subip2&IPV6SUBPREFIXLENGTH=subprefixlength1|
subprefixlength2& IPV6SUBPORTSTART =ipv6subportstart1|
ipv6subportstart2&IPV6SUBPORTEND=
ipv6subportend1|ipv6subportend2&IPV6DESTIP=ipv6destip1|ipv6destip2&IPV6DESTPREFIXLENGTH=
destprefixlength1|destprefixlength2&IPV6DESTPORTSTART=destportstart1|kkkkk
destportstart2k&IPV6DESTPORTEND= destportend1| destportend2
```

Where:

X.X.X.X is needed as subscribe ip in bodsession while ipv6subip1 and ipv6subip2 are as sourceip parameters which set in ipv6 Classifiers. So do other parameters.

modifySession.do IPv6 Interface

The modifySession.do interface allows an application to modify the session IPv6 classifier parameters. Use the following syntax when using this interface:

```
http://<serverip>/BoD/modifySession.do?SSID=sessionid&IPV6SUBIP=Y.Y.Y.Y&
IPV6SUBPREFIXLENGTH=SUBPREFIXLENGTH&IPV6SUBPORTSTART=SPStart&
IPV6SUBPORTEND=SPEnd&IPV6DESTIP=Z.Z.Z.Z&IPV6DESTPREFIXLENGTH=
DESTPREFIXLENGTH&IPV6DESTPORTSTART=DPStart&IPV6DESTPORTEND=DPEnd
&UPBW=U&UPBWMAX=Umax&DOWNBW=D&DOWNBWMAX=Dmax
```

Where:

`serverip` — specifies the IP Address of the BoD server.

`sessionid` — specifies the unique active session id.

`Y.Y.Y.Y` — specifies the IPV6 source subnet address and is used for the classifier.

`SUBPREFIXLENGTH` — specifies the prefix length of the IPV6 source subnet.

`Z.Z.Z.Z` — specifies IPV6 destination subnet address and is used for the classifier.

DESTPREFIXLENGTH — specifies the prefixlength of IPV6 destination subnet.

SPStart — specifies the start port number for the IPV6 source port range.

SPEnd — specifies the end port number for the IPV6 source port range.

DPStart — specifies the start port number for the IPV6 destination port range.

DPEnd — specifies the end port number for the IPV6 destination port range.

U (optional) — specifies the upstream bandwidth in kbps.

D (optional) — specifies the downstream bandwidth in kbps.

Umax (optional) — specifies the upstream maximum reserved bandwidth in kbps.

Dmax (optional) — specifies the downstream maximum reserved bandwidth in kbps.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
<STATUS>success</STATUS>
<SSID>ssid</SSID>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
<STATUS>failure</STATUS>
<FAILUREREASON>description of the failure reason</FAILUREREASON>
</BODY>
```

Important: If the BoD session service bandwidth and classifier parameters have been set on the CMP GUI, then the configured parameters will not be changed even after sending a session modification request.

deleteSession.do Interface

The deleteSession.do interface allows an application to remove the QoS resources for a session previously created with createSession.do. Use the following syntax when using this interface:

```
http://<serverip>/bod/deleteSession.do?SSID=s
```

Where:

serverip — specifies the IP Address of the BoD AM.

s — specifies the SSID returned from create request.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
```

```
  <FAILUREREASON>description of the failure reason</FAILUREREASON>
</BODY>
```

A failure status is displayed if the session could not be torn down.

HTTP Session with Multiple IPv6 Classifiers

createSession.do IPv6 Interface with multiple classifiers

Use the following syntax when using this interface:

Note: Parameter values used for multiple Classifiers are formatted and interpreted similarly to the single classifier createSession.do. However, parameters related to the Classfier(s) for the service will include multiple values, separated by a pipe ("|") character, and ordered corresponding to the Index values in the Service Definition. If a given parameter is not required for one Classifier in a multiple-Classifier Service, then the corresponding parameter should be omitted but the pipe character is still required. If a given parameter is not required for one Classifier in a multiple-Classifier Service, then the corresponding parameter should be omitted but the pipe character is still required. Or the corresponding parameter should be populated as -1.

createSession.do IPv6 Interface with multiple classifiers

Use the following syntax when using this interface:

```
http://<serverip>/bod/createSession.do?SERVICENAME=multiIPv6&GROUPNAME=groupname&
SUBIP=X.X.X.X&IPV6SUBIP=ipv6subip1|ipv6subip2&IPV6SUBPREFIXLENGTH=subprefixlength1|
subprefixlength2& IPV6SUBPORTSTART
=ipv6subportstart1|ipv6subportstart2&IPV6SUBPORTEND=
ipv6subportend1|ipv6subportend2&IPV6DESTIP=ipv6destip1|ipv6destip2&IPV6DESTPREFIXLENGTH=
destprefixlength1|destprefixlength2&IPV6DESTPORTSTART=destportstart1|kkkkk
destportstart2k&IPV6DESTPORTEND=destportend1| destportend2&CMTSIP=cmtsip
```

Note: In addition to the basic syntax and parameters used for multiple standard classifiers, these parameters can be used:

```
UPBW=u&UPBWMAX=umax&DOWNBW=d&DOWNBWMAX=dmax&DUR=t
&SUBNAME=user&SSID=s&DISABLERETRYONERROR=<true|false>
&UPRAMASK=uram&UPFAMASK=ufam&UPAARMASK=uaarm &DOWNRAMASK=dram
&DOWNFAMASK=dfam&DOWNAARMASK=daarm&DOWNPTR=dptr&UPPTR=uptr&VOLLIMIT=v
```

Where:

X.X.X.X is needed as subscribe ip in bodsession while ipv6subip1 and ipv6subip2 are as **`sourceip`** parameters which set in ipv6 Classifiers. So do other parameters.

`umax` (optional) — specifies the upstream maximum reserved bandwidth in kbps.

`d` (optional) — specifies the downstream bandwidth in kbps.

`dmax` (optional) — specifies the downstream maximum reserved bandwidth in kbps.

`t` (optional) — specifies the duration the session should be active. This session is automatically torn down after **`t`** minutes.

`uaarm` -- specifies the Upstream Attribute Aggregation Rule Mask.

`user` (optional) — specifies the username string for this subscriber for recording purposes.

s (optional) — specifies the application-supplied session ID. This is automatically generated if not supplied.

true|false — allows the BoD retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to **true**, when the BoD AM encounters one of these errors, the retry is not attempted. If this parameter is set to **false** or omitted, the BoD AM tries to recover from any recoverable errors.

uram -- specifies the specifies the Upstream Required Attribute Mask.

ufam -- specifies the Upstream Forbidden Attribute Mask.

dram -- specifies the Downstream Required Attribute Mask.

dfam -- specifies the Downstream Forbidden Attribute Mask.

daarm -- specifies the Downstream Attribute AggregationRule Mask.

dptr -- specifies the Downstream Peak Traffic Rate.

uptr -- specifies the Upstream Peak Traffic Rate.

v -- specifies the volume limit for the session.

cmtsip -- specifies the IP address of CMTS.

modifySession.do IPv6 Interface

The modifySession.do interface allows an application to modify the session IPv6 classifier parameters.

Note: This function does not support multiple classifiers.

Use the following syntax when using this interface:

```
http://<serverip>/BoD/modifySession.do?SSID=sessionid&IPV6SUBIP=Y.Y.Y.Y&
IPV6SUBPREFIXLENGTH=SUBPREFIXLENGTH&IPV6SUBPORTSTART=SPStart&
IPV6SUBPORTEND=SPEnd&IPV6DESTIP=Z.Z.Z.Z&IPV6DESTPREFIXLENGTH=
DESTPREFIXLENGTH&IPV6DESTPORTSTART=DPStart&IPV6DESTPORTEND=DPEnd
&UPBW=U&UPBWMAX=Umax&DOWNBW=D&DOWNBWMAX=Dmax
```

Where:

serverip — specifies the IP Address of the BoD server.

sessionid — specifies the unique active session id.

Y.Y.Y.Y — specifies the IPV6 source subnet address and is used for the classifier.

SUBPREFIXLENGTH — specifies the prefix length of the IPV6 source subnet.

Z.Z.Z.Z — specifies IPV6 destination subnet address and is used for the classifier.

DESTPREFIXLENGTH — specifies the prefixlength of IPV6 destination subnet.

SPStart — specifies the start port number for the IPV6 source port range.

SPEnd — specifies the end port number for the IPV6 source port range.

DPStart — specifies the start port number for the IPV6 destination port range.

DPEnd — specifies the end port number for the IPV6 destination port range.

U (optional) — specifies the upstream bandwidth in kbps.

D (optional) — specifies the downstream bandwidth in kbps.

Umax (optional) — specifies the upstream maximum reserved bandwidth in kbps.

Dmax (optional) — specifies the downstream maximum reserved bandwidth in kbps.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
<STATUS>success</STATUS>
<SSID>ssid</SSID>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
<STATUS>failure</STATUS>
<FAILUREREASON>description of the failure reason</FAILUREREASON>
</BODY>
```

Important: If the BoD session service bandwidth and classifier parameters have been set on the CMP GUI, then the configured parameters will not be changed even after sending a session modification request.

deleteSession.do Interface

The deleteSession.do interface allows an application to remove the QoS resources for a session previously created with createSession.do. Use the following syntax when using this interface:

```
http://<serverip>/bod/deleteSession.do?SSID=s
```

Where:

serverip — specifies the IP Address of the BoD AM.

s — specifies the SSID returned from create request.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
 </BODY>
```

A failure status is displayed if the session could not be torn down.

HTTP Traffic Classifier Interfaces

The Traffic Classifier Interface is used for applications that classify traffic, which allows the caller to assign this traffic to a particular Class of Service and modify the classifier list for an existing service flow.

addTrafficClassifier.do Interface

The addTrafficClassifier.do interface allows an application to add a traffic stream to a flow. Use the following syntax when using this interface:

```
http://<serverip>/bod/addTrafficClassifier.do?SERVICENAME=servicename
&GROUPNAME=groupname&SUBIP=x.x.x.x&SUBPORT=n&DESTIP=y.y.y.y
&DESTPORT=m&DISABLERETRYONERROR=<true|false>&CMTSIP=cmtsip
```

Where:

`serverip` — specifies the IP address of the BoD AM.

`servicename` — specifies the PCMM service name, which defines the upstream and downstream traffic profiles. Refer to *Creating a Service* for additional information.

`groupname` (optional) — specifies a caller-supplied group name assigned to the session for reporting purposes.

`n` (optional) — specifies the subscriber port for the stream.

`y.y.y.y` (optional if subscriber port supplied) — specifies the destination application server IPv4 address.

`m` (optional) — specifies the destination port on the application server.

`true|false` — allows the BoD retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to **`true`**, when the BoD AM encounters one of these errors, the retry is not attempted. If this parameter is set to **`false`** or omitted, the BoD AM tries to recover from any recoverable errors.

CMTSIP -- specifies the IP address of the CMTS.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
   <SSID>N</SSID>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
 </BODY>
```

The returned status field displays the success or failure of the requested operation. A failed status may be displayed if the resource is not available within the access network or if the MSO has created and deployed a policy to reject the request. If the service is configured to be re-established on failure, a “success” message will always be returned, as long as the BoD AM successfully received the request.

deleteTrafficClassifier.do Interface

The deleteTrafficClassifier.do interface allows the caller to remove classifiers from an existing flow, removing specific traffic patterns from the flow treatment. Use the following syntax when using this interface:

```
http://<serverip>/bod/deleteTrafficClassifier.do?SERVICENAME=servicename
&SUBIP=x.x.x.x&DESTIP=y.y.y.y&DESTPORT=m
&DISABLERETRYONERROR=<true|false>&CMTSIP=cmtsip
```

Where:

`serverip` — specifies the IP address of the BoD AM.

`servicename` — specifies the PCMM service name, which defines the upstream and downstream traffic profiles. Refer to *Creating a Service* for additional information.

`n` (optional) — specifies the subscriber port for the stream.

`y.y.y.y` (optional if subscriber port supplied) — specifies the destination IPv4 address.

`m` (optional) — specifies the destination port.

`true|false` — allows the BoD retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to **`true`**, when the BoD AM encounters one of these errors, the retry is not attempted. If this parameter is set to **`false`** or omitted, the BoD AM tries to recover from any recoverable errors.

CMTSIP -- specifies the IP address of the CMTS.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
   <SSID>N</SSID>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
 </BODY>
```

A failure status is displayed if the supplied classifier could not be found for this subscriber.

HTTP Traffic Extended Classifier Interfaces

The Traffic Extended Classifier Interface is used for applications that classify traffic, which allows the caller to assign this traffic to a particular Class of Service and modify the classifier list for an existing service flow.

addTrafficClassifier.do Extended Interface

The addTrafficClassifier.do interface allows an application to add a traffic stream to a flow. Use the following syntax when using this interface:

```
http://<serverip>/bod/addTrafficClassifier.do?SERVICENAME=servicename
&GROUPNAME=groupname&SUBIP=x.x.x.x&EXTSUBIP=y.y.y.y
&EXTSUBMASK=submask&EXTSUBPORTSTART=spstart&EXTSUBPORTEND=spend
&EXTDESTIP=z.z.z.z&EXTDESTMASK=destmask&EXTDESTPORTSTART=dpstart
&EXTDESTPORTEND=dpend&DISABLERETRYONERROR=<true|false>&CMTSIP=cmtsip
```

Where:

serverip — specifies the IP address of the BoD AM.

servicename — specifies the PCMM service name, which defines the upstream and downstream traffic profiles. Refer to *Creating a Service* for additional information.

groupname (optional) — specifies a caller-supplied group name assigned to the session for reporting purposes.

y.y.y.y — specifies the source subnet address and is used for the classifier. It can also specify any specific IPv4 address provided "submask" value is set to 32.

submask — specifies the IP mask used with source subnet.

spstart — specifies the start port number for the source port range.

spend — specifies the end port number for the source port range.

z.z.z.z — specifies destination subnet address and is used for the classifier. It can also specify any specific IPv4 address provided "destmask" value is set to 32.

destmask — specifies the IP mask used with destination subnet.

dpstart — specifies the start port number for the destination port range.

dpend — specifies the end port number for the destination port range.

true|false — allows the BoD retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to **true**, means that when the BoD AM encounters one of these errors, the retry is not attempted. If this parameter is set to **false** or omitted, the BoD AM tries to recover from any recoverable errors.

cmtsip -- specifies the IP Address of CMTS.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
   <SSID>N</SSID>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
 </BODY>
```

The returned status field displays the success or failure of the requested operation. A failed status may be displayed if the resource is not available within the access network or if the MSO has created and deployed a policy to reject the request. If the service is configured to be re-established on failure, a "success" message will always be returned, as long as the BoD AM successfully received the request.

deleteTrafficClassifier.do Extended Interface

The deleteTrafficClassifier.do interface allows the caller to remove classifiers from an existing flow, removing specific traffic patterns from the flow treatment. Use the following syntax when using this interface:

```
http://<serverip>/bod/deleteTrafficClassifier.do?SERVICENAME=servicename
&GROUPNAME=groupname&SUBIP=x.x.x.x&EXTSUBIP=y.y.y.y
&EXTSUBMASK=submask&EXTSUBPORTSTART=SPStart&EXTSUBPORTEND=SPEnd
&EXTDESTIP=Z.Z.Z.Z&EXTDESTMASK=destMask&EXTDESTPORTSTART=DPStart
&EXTDESTPORTEND=DPEnd&DISABLERETRYONERROR=<true|false>&CMTSIP=cmtsip
```

Where:

`serverip` — specifies the IP address of the BoD AM.

`servicename` — specifies the PCMM service name, which defines the upstream and downstream traffic profiles. Refer to *Creating a Service* for additional information.

`groupname` (optional) — specifies a caller-supplied group name assigned to the session for reporting purposes.

`y.y.y.y` — specifies the source subnet address and is used for the classifier. It can also specify any specific IPv4 address provided "subMask" value is set to 32.

`submask` — specifies the IP mask used with source subnet.

`spstart` — specifies the start port number for the source port range.

`spend` — specifies the end port number for the source port range.

`z.z.z.z` — specifies destination subnet address and is used for the classifier. It can also specify any specific IPv4 address provided "destmask" value is set to 32.

`destmask` — specifies the IP mask used with destination subnet.

`dpstart` — specifies the start port number for the destination port range.

`dpend` — specifies the end port number for the destination port range.

`true|false` — allows the BoD retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to **`true`**, when the BoD AM encounters one of these errors, the retry is not attempted. If this parameter is set to **`false`** or omitted, the BoD AM tries to recover from any recoverable errors.

`cmtsip` -- specifies the IP Address of CMTS.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
   <SSID>N</SSID>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
 </BODY>
```

A failure status is displayed if the supplied classifier could not be found for this subscriber.

HTTP Traffic IPv6 Classifier Interfaces

The Traffic Classifier Interface is used for applications that classify traffic, which allows the caller to assign this traffic to a particular Class of Service and modify the classifier list for an existing service flow.

addTrafficClassifier.do IPv6 Interface

The addTrafficClassifier.do interface allows an application to add a traffic stream to a flow. Use the following syntax when using this interface:

```
http://<serverip>/bod/addTrafficClassifier.do?SERVICENAME=servicename
&GROUPNAME=groupname&SUBIP=x:x:x:x:x:x:x:x
&IPV6SUBIP=x:x:x:x:x:x:x:x&IPV6DESTIP=y.y.y.y.y.y.y.y&IPV6SUBPORTSTART=ss6
&IPV6SUBPORTEND=se6&IPV6DESTPORTSTART=ds6&IPV6DESTPORTEND=de6
&IPV6SUBPREFIXLENGTH=sp6&IPV6DESTPREFIXLENGTH=dp6&DISABLERETRYONERROR=<true|false>
```

Where:

`serverip` — specifies the IP address of the BoD AM.

`servicename` — specifies the PCMM service name, which defines the upstream and downstream traffic profiles. See *Creating a Service* for additional information.

`groupname` (optional) — specifies a caller-supplied group name assigned to the session for reporting purposes.

`true|false` — allows the BoD retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to **`true`**, when the BoD AM encounters one of these errors, the retry is not attempted. If this parameter is set to **`false`** or omitted, the BoD AM tries to recover from any recoverable errors.

`x:x:x:x:x:x:x:x` — specifies the source IPv6 address.

`y.y.y.y.y.y.y.y` — specifies the destination IPv6 address.

`ss6` — specifies the low-end TCP/UDP source port value.

`se6` — specifies the high-end TCP/UDP source port value.

`ds6` — specifies the low-end TCP/UDP destination port value.

`de6` — specifies the high-end TCP/UDP destination port value.

`sp6` — specifies how many high order bits in the corresponding IPv6 subscriber address to consider in determining a match. The valid values are from 0 – 128.

`dp6` — specifies how many high order bits in the corresponding IPv6 destination address to consider in determining a match. The valid values are from 0 – 128.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
   <SSID>N</SSID>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
 </BODY>
```

The returned status field displays the success or failure of the requested operation. A failed status may be displayed if the resource is not available within the access network or if the MSO has created and deployed a policy to reject the request. If the service is configured to be re-established on failure, a "success" message will always be returned, as long as the BoD AM successfully received the request.

deleteTrafficClassifier.do IPv6 Interface

The deleteTrafficClassifier.do interface allows the caller to remove classifiers from an existing flow, removing specific traffic patterns from the flow treatment. Use the following syntax when using this interface:

```
http://<serverip>/bod/deleteTrafficClassifier.do?SERVICENAME=servicename
&SUBIP=x.x.x.x&DESTIP=y.y.y.y&DESTPORT=m
&DISABLERETRYONERROR=<true|false>&IPV6SUBIP=x:x:x:x:x:x:x:x
&IPV6DESTIP=y:y:y:y:y:y:y:y&IPV6SUBPORTSTART=ss6&IPV6SUBPORTEND=se6
&IPV6DESTPORTSTART=ds6&IPV6DESTPORTEND=de6&IPV6SUBPREFIXLENGTH=sp6
&IPV6DESTPREFIXLENGTH=dp6
```

Where:

`serverip` — specifies the IP address of the BoD AM.

`servicename` — specifies the PCMM service name, which defines the upstream and downstream traffic profiles. See *Creating a Service* for additional information.

`n` (optional) — specifies the subscriber port for the stream.

`y.y.y.y` (optional if subscriber port supplied) — specifies the destination IPv4 address.

`m` (optional) — specifies the destination port.

`true|false` — allows the BoD retry logic to be disabled when a session create is attempted and a recoverable error is encountered (such as a PCMM error code 13, or a PCMM error code 127 and subcode 211). If this value is set to **`true`**, when the BoD AM encounters one of these errors, the retry is not attempted. If this parameter is set to **`false`** or omitted, the BoD AM tries to recover from any recoverable errors.

`x:x:x:x:x:x:x:x` — specifies the source IPv6 address.

`y:y:y:y:y:y:y:y` — specifies the destination IPv6 address.

`ss6` — specifies the low-end TCP/UDP source port value.

`se6` — specifies the high-end TCP/UDP source port value.

`ds6` — specifies the low-end TCP/UDP destination port value.

de6 — specifies the high-end TCP/UDP destination port value.

sp6 — specifies how many high order bits in the corresponding IPv6 subscriber address to consider in determining a match. The valid values are from 0 – 128.

dp6 — specifies how many high order bits in the corresponding IPv6 destination address to consider in determining a match. The valid values are from 0 – 128.

When this operation is successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>success</STATUS>
   <SSID>N</SSID>
</BODY>
```

When this operation is not successful, the following is returned as the HTTP response:

```
<BODY>
   <STATUS>failure</STATUS>
   <FAILUREREASON>description of the failure reason</FAILUREREASON>
 </BODY>
```

A failure status is displayed if the supplied classifier could not be found for this subscriber.

Chapter 3

Using the BoD AM User Interface

Topics:

This chapter describes how to open and use the Configuration Management Platform (CMP) User Interface (UI) to configure Bandwidth on Demand Application Manager (BoD AM) functionality.

Logging into the BoD AM

Logging into the CMP system provides access to the BoD GUI. Refer to the *Configuration Management Platform Cable User Guide* for information.

Navigation Pane

The BoD AM functionality is accessed from the CMP system GUI. Refer to the *Configuration Management Platform Cable User Guide* for a complete description of the GUI. *Figure 3: BoD GUI Overview Navigation Screen* shows the Navigation screen.

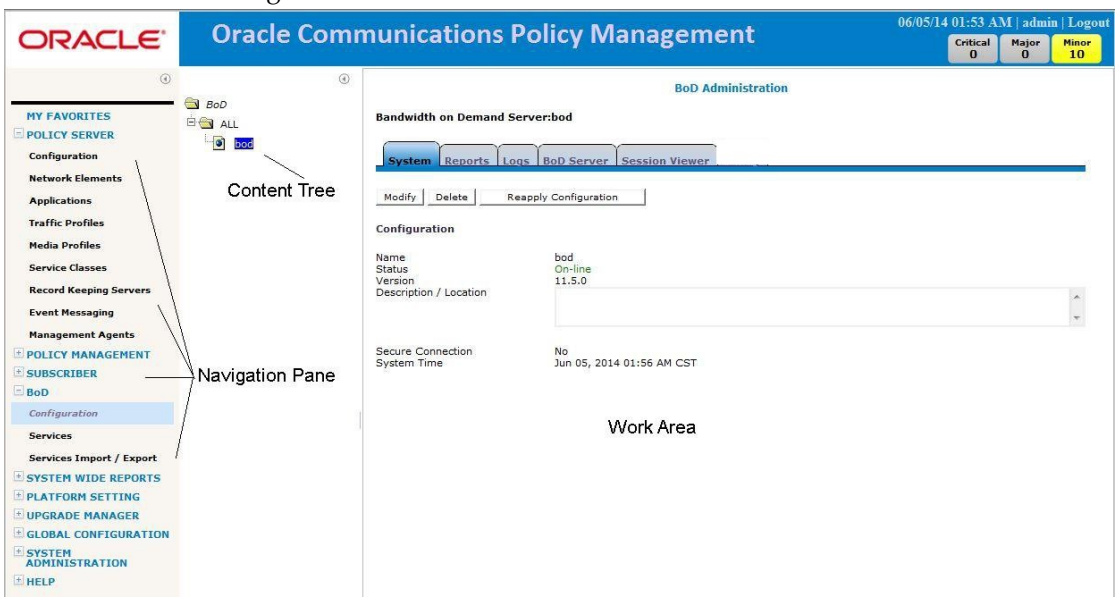


Figure 3: BoD GUI Overview Navigation Screen

Note: The CMP supports the MDF as an optional configuration mode. This mode must be configured before your CMP system will display MDF options. Refer to the *Configuration Management Platform Cable User's Guide* for a description of the Mode Settings page. Contact Customer Support to change an existing CMP system to support MDF functions.

You must have the correct privileges before you can view or make changes to the BoD settings. These privileges are assigned using the **User Management** option from the **System Administration** section of the Navigation pane. Refer to the *Configuration Management Platform Cable User's Guide* for information on user management.

Use the following options in the **BoD** section of the CMP navigation pane to access configurable settings for the BoD AM:

- **Configuration** — Use this option to create BoD servers and server groups, modify settings for an existing BoD server, and access reports, logs, sessions, and debug settings associated with that server.
- **Services** — Use this option to view, modify, create, and delete service information.

- **Services Import/Export** — Use this option to import and export services to a selected policy server.

BoD AM Audit Log

You can access the audit log for the BoD AM server using the **Audit Log** option in the **System Administration** section of the CMP GUI. Refer to the *Configuration Management Platform Cable User 's Guide* for additional information.

Chapter 4

Configuring BoD AM Servers

Topics:

This chapter describes how to create new Bandwidth on Demand Application Manager (BoD AM) servers and groups.

Creating a BoD Server

To create a BoD AM server:

1. From the **BoD** section of the navigation pane, select **Configuration**.
   The content tree displays a list of the Bandwidth on Demand servers.
2. Select the **ALL** folder.
   The **BoD Administration** page opens in the work area.

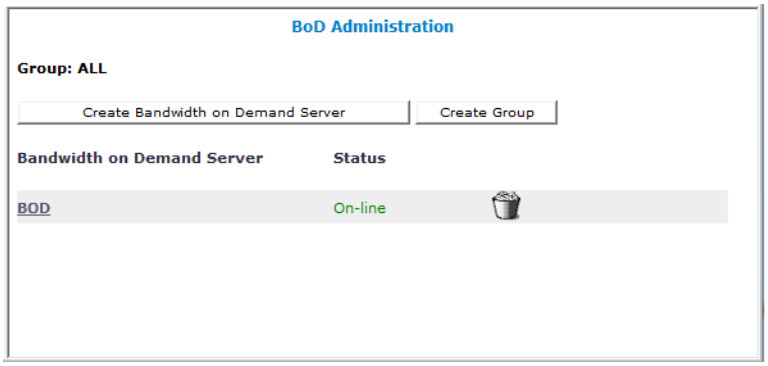


Figure 4: BoD Administration Page for the ALL Group

3. Click **Create Bandwidth on Demand Server**.
   The **New BoD** page opens in the work area.

   Edit the fields to create the server:

   - **Associated Cluster** — Select the name of the cluster where the BoD server associates.
   - **Name** — The name of the BoD server.
   - **Description/Location** — Descriptive text helping to identify the BoD server.
   - **Secure Connection** — Select to require a secure connection for the BoD server.

4. When you finish, click **Save**.

The BoD server is created and added to the list in the content tree.

Creating a BoD Server Group

You can create groups for BoD servers. Updates made to a group also apply to all servers in a group.

To create a BoD server group:

1. From the **BoD** section of the navigation pane, select **Configuration**.
   The content tree displays a list of the BoD servers.
2. Select the **ALL** folder.
   The **BoD Administration** page opens in the work area.
3. Click **Create Group**.

The **Create Group** page opens in the work area.

4. Enter the name of the group in the **Name** field.
5. When you finish, click **Save**.

A BoD server group is created.

Viewing and Modifying BoD Server Topology Information

The **System** tab on the **BoD Administration** page allows you to view and modify BoD server information. You can also use this tab to delete BoD servers from the CMP system and re-apply a configuration to a server.

Viewing BoD Topology Information

To view the server topology information:

1. From the **BoD** section of the navigation pane, select **Configuration**.
   The content tree displays a list of the servers.
2. From the content tree, select a server.
   The **BoD Administration** page opens in the work area and displays topology information for the selected server.

   Note: The **BoD Administration** page automatically opens to the **System** tab.

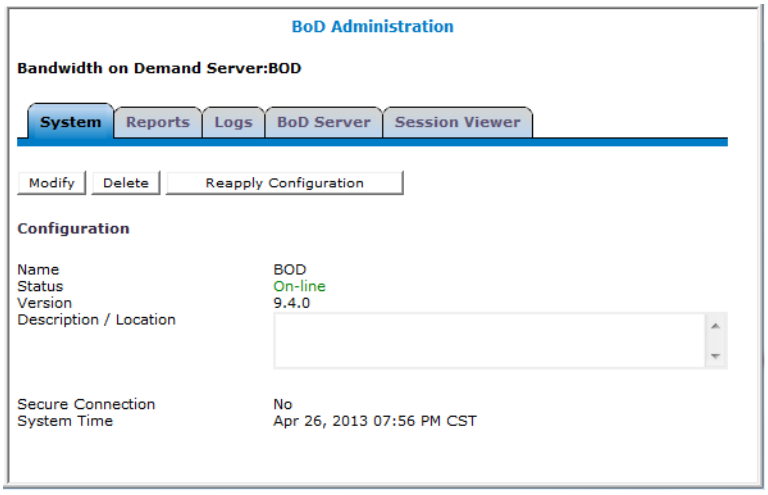


Figure 5: BoD Administration Page

The following settings are displayed:

- **Name** — The name of the server.
- **Status** — The status of the server.
- **Version** — The version of the software running on the server.
- **Description/Location** — Descriptive text for the server.

- **Secure Connection** — Whether the server has a secure connection.
- **System Time** — The current date and time.

Modifying BoD Topology Information

To modify BoD server topology information:

1. From the **BoD** section of the navigation pane, select **Configuration.**
   The content tree displays a list of the BoD servers.
2. Select a BoD server.
   The **BoD Administration** page opens in the work area and displays topology information about the server.
3. Click **Modify**.
   The **Modify System Settings** page opens in the work area.
4. Update the information. For a detailed description of each setting, refer to *Creating a BoD Server*.
5. When you finish, click **Save**.

Configuring a Server with Multiple Destination IP Addresses

To configure the BoD server topology information to send notifications to multiple pre-configured servers:

1. From the **BoD** section of the navigation pane, select **Configuration.**
   The content tree displays a list of the BoD servers.
2. From the content tree, select a BoD server.
   The **BoD Administration** page opens in the work area and displays topology information about the server.
3. Click **Modify**.
   The **Modify System Settings** page opens in the work area.
4. Select the server settings. See *Viewing BoD Server Settings* for details.
5. Select the HTTP method. See *Viewing BoD Server Settings* for details.
6. Configure the server information. Refer to *Viewing BoD Server Settings* for details.
7. Click **Add**.
   The information appears in the list field.
8. Repeat steps *Step 4* through *Step 7* to add more addresses.
9. When you finish, click **Save**.

Deleting a BoD Server

To delete a BoD server:

1. From the **BoD** section of the navigation pane, select **Configuration**.
   The content tree displays a list of the BoD servers.
2. Remove the BoD server using one of the following methods.

a) Select the **ALL** folder. A list of BoD servers appears in the work area. Click the (trash can) icon next the server that you want to delete.
b) Select a BoD server from the content tree. Click **Delete** on the **BoD Administration** page.

A confirmation message appears

3. Click **Ok** to delete the server.

The BoD server is deleted.

Reapplying the Configuration to a BoD Server

Note: Reapplying the configuration pushes the settings on the CMP system to the selected BoD server and overwrites the current settings stored on that server.

Caution: Reapplying the configuration pushes the settings on the CMP system to the selected BoD server and overwrites the current settings stored on that server.

To reapply the configuration to a BoD server:

1. From the **BoD** section of the navigation pane, select **Configuration**.
   The content tree displays a list of the BoD servers.
2. Select a BoD server.
   The **BoD Administration** page opens in the work area and displays information about the selected server.
3. Click **Reapply Configuration**.
   The configuration is reapplied.

Viewing and Modifying BoD Server Settings

The **BoD Server** tab on the BoD Administration page is used to configure customer-specific settings for the BoD server, such as PacketCable Multimedia (PCMM) settings and session status notification.

This information is used globally in all future BoD session activity.

Viewing BoD Server Settings

To view BoD server settings:

1. From the **BoD** section of the navigation pane, select **Configuration**.
   The content tree displays a list of the BoD servers.
2. Select a BoD server.
   The **BoD Administration** page opens in the work area.
3. Select the **BoD Server** tab.

   The **BoD Administration** page displays information for the selected BoD server.

   The configuration information includes the following:

- **MAC Translation URL** — The address of the server that provides the IP Address mapping for a request containing a MAC address. To support a design where the MAC address is delivered to the BoD application and a separate system provides the mapping of the MAC address to the corresponding IP address, configurable settings are available to look up these translations on a per-session basis. If this method is used there is a latency introduced for the lookup.
- **Translate MAC to IP in recreate** — If this parameter is enabled, then an additional MAC to IP translation is performed when the BoD server receives a gate report or learns that a gate has closed before the BoD server attempts to re-establish the gates. If the parameter is disabled, then the BoD server attempts to re-establish the gates utilizing the IP address received from the original MAC translation operation.
- **Session Status Notification** — This section of the **BoD Administration** page is used to configure whether the BoD server will notify a third-party calling application of events that change the session status.
  - **Notification server IP or FQDN** — The address of the BoD server that is used to notify a third-party calling application of events that change the session status.
  - **Server listening port** — The number of the port that is used to accept a specific request.

    Note: This option is not visible when Notification Strategy is disabled.
  - **Server pathname** — HTTP URL for the global or third-party calling application server.

    Note: This option is not visible when Notification Strategy is disabled.
  - **Notification strategy** — Determines whether the BoD server will notify a third-party calling application through HTTP requests of events that change the session status. You can select one of the following:
    - **Disabled** — Notification of session modification does not occur.
    - **Global server** — A global notification server is used to accept all HTTP requests. If this option is selected, then values must be entered in the **Notification server IP or FQDN**, **Server pathname** and **Server listening port** fields.

      Note: Use this option if multiple destination IP addresses are used for notifications.
    - **Calling Application Server** — Each notification is sent to the application server that created the session. If this option is selected, then a value between 0 - 65535 must be entered in the **Server listening port** field; and the value entered in the **Server pathname** field must be able to be resolved to a valid HTTP URL when combined with a valid server, port and schema.

      Note: Use this option if multiple destination IP addresses are used for notifications.
    - **HTTP GET** — The BoD sends the parameters through the URL using this format: http://NOTIF_IP/*PATH*?ssid=SESS_ID&reqIP=REQ_IP&state=S&reason=R&subcode=C
    - **HTTP POST** — the BoD sends the parameters in the body of HTTP message using this format: ssid=SESS_ID&reqIP=REQ_IP&state=S&reason=R&subcode=C

- **PCMM** — This section of the **BoD Administration** page is used to configure PacketCable Multimedia (PCMM) settings.
  - **PCMM Enabled** — Defines whether PacketCable Multimedia is enabled.
  - **PCMM Policy Server Configuration** — This section is used to add one or more policy servers (internal or external), allowing the BoD server to send out PCMM requests to an MPE device.

- **PCMM Application Manager Id** — The identifier for a BoD server. This value is an identifier within the protocol, and serves as a label for business rule enforcement at the policy server. It is also possible, within the specific service definitions, to override this value on a per service basis.
- **PCMM Gate Delete Retry Interval (seconds)** — The period of time to wait before attempting another gate delete.
- **Maximum gate delete retry (times)** — The maximum number of times a gate delete will be attempted.

Modifying BoD Server Information

To modify BoD server information:

1. From the **BoD** section of the navigation pane, select **Configuration**.
   The content tree displays a list of the BoD servers.
2. From the content tree, select a BoD server.
   The **BoD Administration** page opens in the work area.
3. Select the **BoD Server** tab.
   The **BoD Administration** page displays the server information.
4. Click **Modify**.
   The **BoD Administration** page displays options to allow you to modify the BoD server information.
5. Enter the information. For a detailed description of each field, refer to *Viewing BoD Server Settings*.
6. Use the PCMM area to add a new MPE device or clone/edit/delete the settings of an existing MPE device.
7. (Optional) Add, modify, or delete settings.
   - Cloning an entry in the table
     1. Select an entry in the table.
     2. Click **Clone**. The **Clone** window opens with the information for the entry.
     3. Make changes as required.
     4. When you finish, click **Save**. The entry is added to the table
   - Editing an entry in the table
     1. Select the entry in the table.
     2. Click **Edit**. The **Edit Response** window opens, displaying the information for the entry.
     3. Make changes as required.
     4. When you finish, click **Save**. The entry is updated in the table.
   - Deleting a value from the table
     1. Select the entry in the table.
     2. Click **Delete**. A confirmation message displays.
     3. Click **Delete** to remove the entry. The entry is removed from the table.
8. When you finish, from the **BoD Administration** page, click **Save**.

Configuring BoD Server Configuration Key Information

To view or modify BoD server configuration key information:

1. From the **BoD** section of the navigation pane, select **Configuration**.
   The content tree displays a list of the BoD servers.
2. From the content tree, select a BoD server.
   The **BoD Administration** page opens in the work area.
3. Select the **BoD Server** tab.
   The **BoD Administration** page displays the server information.
4. Click **Advanced**.
   The **Other Advanced Configuration Settings** page opens.

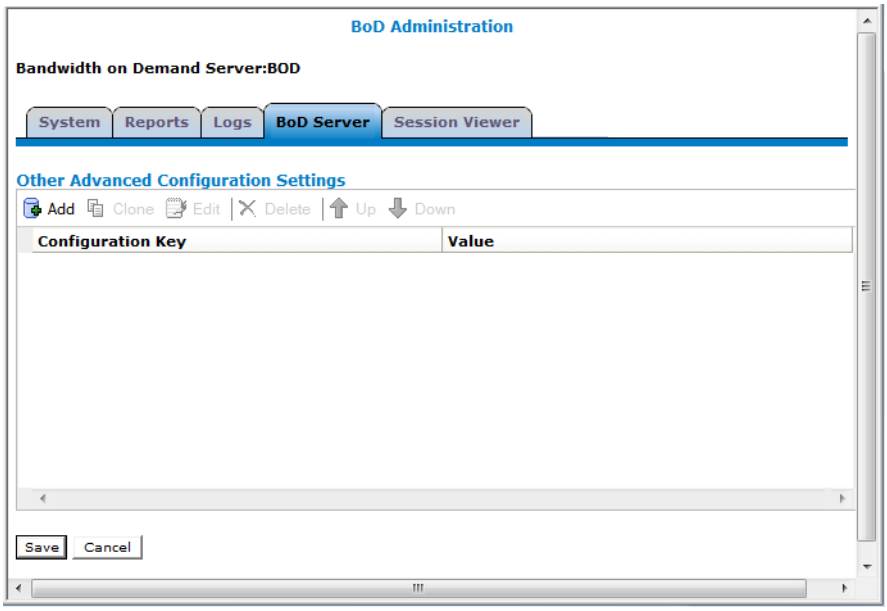


Figure 6: Other Advanced Configuration Settings Page

5. Click **Add** to add a configuration key.

   The **Add Configuration Key Value** pop-up appears.

6. When you finish, click **Save**.
7. (Optional) Add, modify, or delete keys.
   - Cloning an entry in the table
     1. Select an entry in the table.
     2. Click **Clone**. The **Clone** window opens with the information for the entry.
     3. Make changes as required.
     4. When you finish, click **Save**. The entry is added to the table
   - Editing an entry in the table
     1. Select the entry in the table.

2. Click **Edit**. The **Edit Response** window opens, displaying the information for the entry.
3. Make changes as required.
4. When you finish, click **Save**. The entry is updated in the table.

- Deleting a value from the table

1. Select the entry in the table.
2. Click **Delete**. A confirmation message displays.
3. Click **Delete** to remove the entry. The entry is removed from the table.

8. When you finish, from the **BoD Administration** page, click **Save**.

Chapter 5

BoD Server Reports

Topics:

This chapter describes the Reports tab on the BoD Administration page. This tab displays current information about the Bandwidth on Demand (BoD) cluster system, each blade of the cluster, and the protocol statistics.

Viewing BoD Reports

To view BoD reports:

1. From the **BoD** section of the navigation pane, select **Configuration**.
   The content tree displays a list of the BoD servers.
2. From the content tree, select a BoD server.
   The BoD Administration page opens in the work area.
3. On the BoD Administration page, select the **Reports** tab.
   The BoD Administration page displays report information for the selected server.

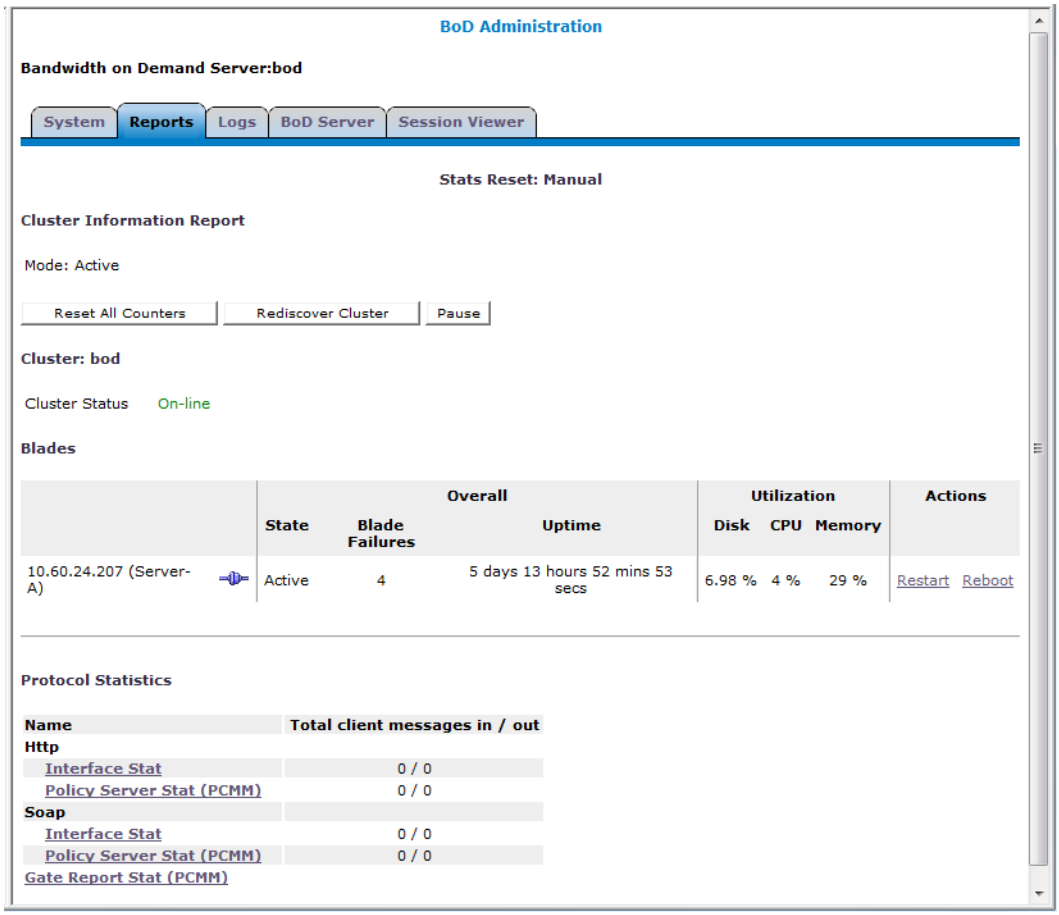


Figure 7: BoD Reports

The report is divided into 3 sections:

- Cluster Information Report
- Blades
- Protocol Statistics

Viewing the Cluster Information Report

The Cluster Information Report is located on the BoD Administration page Reports tab. See *Viewing BoD Reports* for information on accessing the Reports tab.

The Cluster Information report consists of the following:

- **Cluster Status** — Current state of the cluster. Supported values are:
  - **On-Line** — The cluster is healthy and fully redundant
  - **Non-Service Affecting Failure** — The cluster is fully functional, but cabling is not fully connected.
  - **Failed** — The cluster is not available.
  - **Degraded** — Service is currently not affected but a failure has occurred.
- **Failures** — The number of cluster failures since the last time the counter was cleared.
- **Uptime** — The total amount of time this cluster has been in service since the last failure.

From the Cluster Information Report, you can perform one of the following:

- Click **Reset All Counters** to reset all of the counters to zero.
- Click **Rediscover Cluster** to refresh the page manually and obtain the latest reports from the remote server.
- Click **Pause** to stop the page refreshing automatically.

Viewing Blade Information

Blade information is displayed on the BoD Administration page Reports tab. See *Viewing BoD Reports* for information on accessing the Reports tab.

Blade information includes:

- **State** — The state of the blade (active,)
- **Blade Failures** — The number of times the blade has failed.
- **Uptime** — The amount of time the blade has been active.
- **Disk** —The amount of disk space used by the blade.
- **CPU** — The amount of processor space used by the blade.
- **Memory** — The amount of memory used by the blade.

You can perform the following tasks from the Actions section of the Blades report:

- Click **Restart** to restart a blade.
- Click **Reboot** to reboot a blade.

Viewing Protocol Statistics

HTTP, SOAP, and Gate Information statistics are available from the BoD Administration page Reports tab. See *Viewing BoD Reports* for information on accessing the Reports tab.

From the Protocol Statistics section of the report, select the desired statistics.

- HTTP — Interface or Policy Server statistics for the HTTP protocol.

  - Interface — Click **Interface Stat**. The HTTP Interface Statistics page opens. This page displays information on the Total, Success, and Failure Requests, Query Session, and Last stats session reset time.
  - Policy Server — Click **Policy Server Stat (PCMM)**. The Policy Server Statistics page opens. This page displays information on the Total, Successful, Failed, and Timed Out Messages, and Total Sessions, and Last stats session reset time.
- SOAP — Interface or Policy Server statistics for the SOAP protocol.
  - Interface — Click **Interface Stat**. The SOAP Interface Statistics page opens. From this page, you can view information on the Total, Success, and Failure Requests, and Last stats session reset time.
  - Policy Server— Click **Policy Server Stat (PCMM)**. The Policy Server Statistics page opens. From this page, you can view information on the Total, Successful, Failed, and Timed Out Messages, Total Sessions, and Last stats session reset time.
- Gate — Gate Report Statistics.

  Click **Gate Report Stat (PCMM)**. The Gate Report Statistics page opens. This page displays information on Total, Invalid, Ignored, and Processed gate reports.

On all of the Statistics pages, you can perform the following:

- **Reset Counters** — Resets all counters to zero.
- **Show Deltas** — Shows the change in counters between the current and previous reports. After clicking **Show Deltas**, the button changes to **Show Absolute**, which shows the total count.
- **Pause** — Forces the GUI to stop refreshing automatically.
- **Cancel** — Returns you to the BoD Administration page.

Chapter 6

Viewing and Modifying the BoD Trace Log

Topics:

This chapter describes the BoD trace log. You can view the trace log for the Bandwidth on Demand (BoD) server and modify the trace log settings from the Logs tab on the BoD Administration page.

BoD AM Trace Log Forwarding

The CMP system provides log forwarding configuration for all products that have trace logs: MPE, MRA, MA, BoD, and the CMP itself. Refer to *Configuring Log Settings* in the *Configuration Management Platform Cable User 's Guide* for additional information.

Viewing BoD Trace Log Configuration

To view the BoD Trace log configuration information:

1. From the **BoD** section of the navigation pane, select **Configuration**.
   The content tree displays a list of the BoD servers.
2. From the content tree, select a **BoD** server.
   The BoD Administration page opens in the work area.
3. On the BoD Administration page, select the **Logs** tab.
   The BoD Administration page displays the Trace log level, which describes the level of error that is logged.

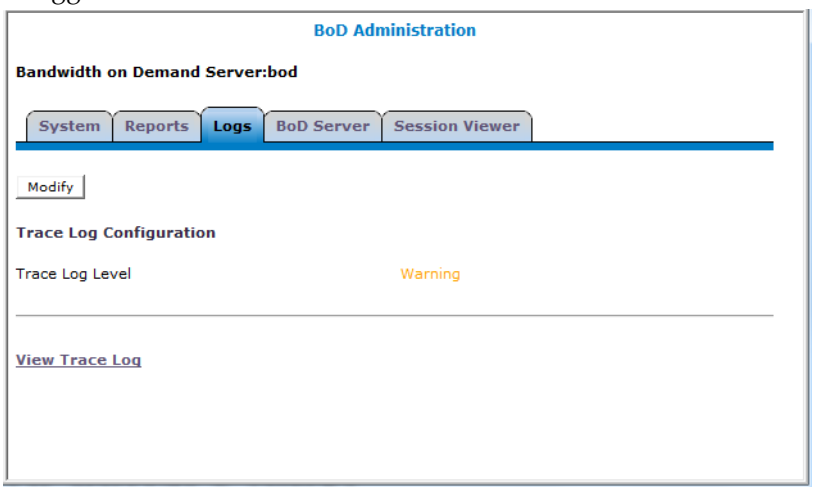


Figure 8: Trace Log Configuration Page

Modifying BoD Trace Log Configuration

To modify the BoD Trace log configuration information:

1. From the **BoD** section of the navigation pane, click **Configuration**.
   The content tree displays a list of the BoD servers.
2. From the content tree, select a **BoD** server.

The **BoD Administration** page opens in the work area.

3. On the BoD Administration page, select the **Logs** tab.
   The **BoD Administration** page displays Trace log information.
4. Click **Modify**.
   The **Modify Trace Log Settings** page opens in the work area.
5. Select the desired level of error from the pull-down menu.
6. When you finish, click **Save** (or **Cancel** to discard your changes).

Viewing and Modifying the Trace Log

The Trace log displays the events logged by the system during the timeframe specified by the user. Events logged are based on the BoD Trace log setting.

To view or modify the BoD Trace log:

1. From the **BoD** section of the navigation pane, click **Configuration**.
   The content tree displays a list of the BoD servers.
2. From the content tree, select a BoD server.
   The BoD Administration page opens in the work area.
3. On the BoD Administration page, select the **Logs** tab.
   The BoD Administration page displays Trace log configuration information.
4. Click **View Trace Log**.
   The Trace Log Viewer opens in the work area.

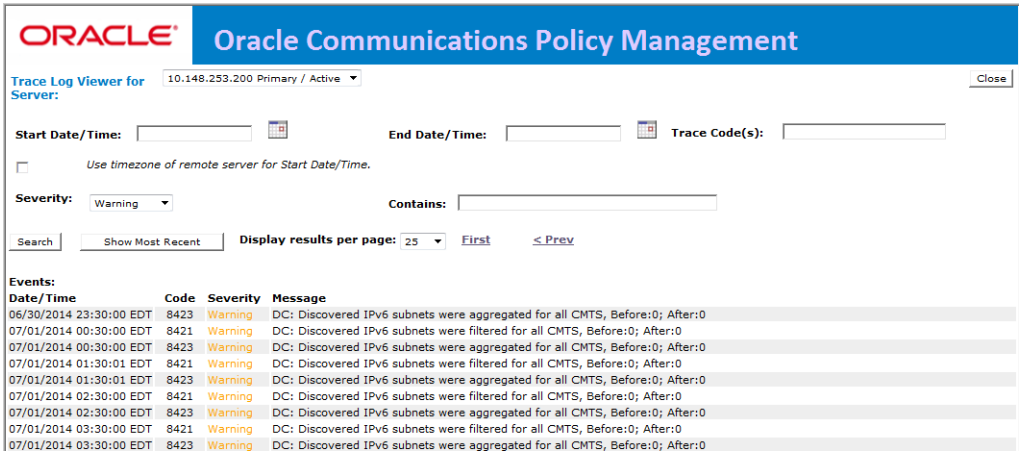


Figure 9: Trace Log Viewer

The following information is displayed for each event:

- **Date/Time** — Date and time when the event occurred. Time is relative to the server time.
- **Code** — Event code. Refer to *BoD Interface Error Codes* for additional information on event codes.
- **Severity** — Severity level of the event.
- **Message** — Message associated with the event. If there is additional information available, the event entry shows as a link. Click on the link to see additional detail in the frame below.

5. You can modify the filter settings on the Trace Log Viewer to display certain records.
   a) To change the start date and time or end date and time, click the calendar next to the **Start Date/Time** or **End Date/Time** field, respectively. The calendar and clock displays. Set the desired date and time and click **Enter** to change the settings.
   b) Click **Use timezone of remote server for Start Date/Time** to filter by the timezone of the remote server.
   c) Enter the desired Trace Codes in the **Trace Code(s)** field.
   d) Select a severity level from the **Severity** drop-down list. Enter the text to match on in the **Contains** field.
   e) Select the number of results to display on the page from the **Display Results per Page** drop-down list.
6. After entering the filtering information click one of the following to update the Trace Log Viewer:
   - **Search** — Locates the events matching the desired filtering criteria.
   - **Show Most Recent** — Shows the most current events.
7. To exit the Trace Log viewer, click **Close** or the red **X** in the upper right-hand corner of the viewer.

Enabling BoD Trace Log Forwarding

You can set BoD to foward trace logs from specified servers. Complete these steps to enable trace log forwarding.

1. From the **BoD** section of the navigation pane, select **Configuration**.
   The content tree displays a list of the BoD servers.
2. From the content tree, select a **BoD** server.
   The BoD Administration page opens in the work area.
3. On the BoD Administration page, select the **Logs** tab.
   The BoD Administration page displays the Trace log level, which describes the level of error that is logged.

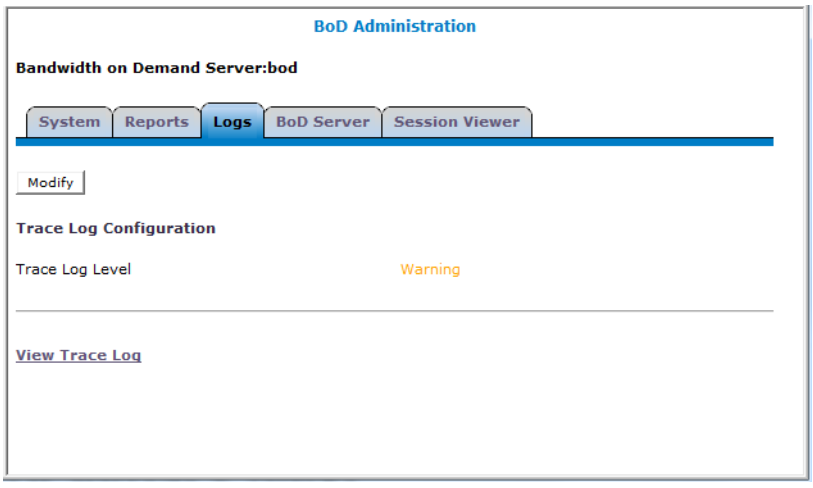


Figure 10: Trace Log Configuration Page

4. Click **Modify** to open the **Modify Trace Log Settings** screen.

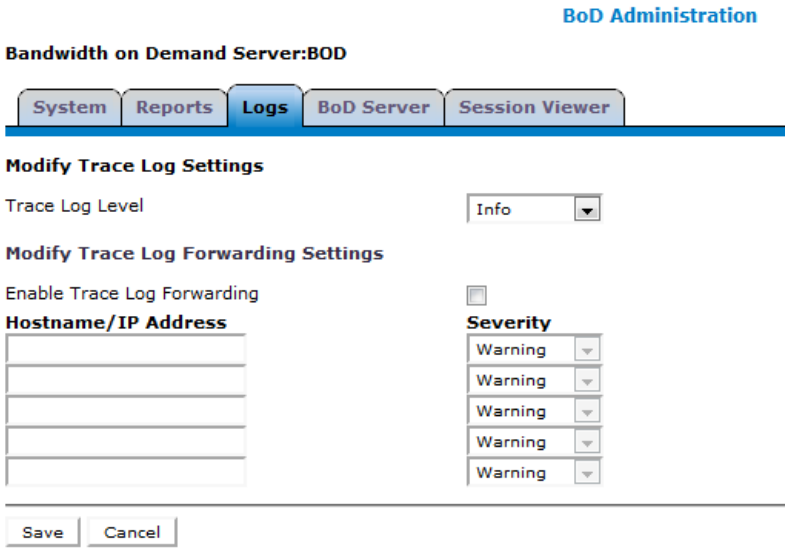


Figure 11: Modify Trace Log Settings Screen

5. Select the needed **Trace Log Level** from the list.
6. Select **Enable Trace Log Forwarding**.

   Note: Hostname/IP Address fields cannot be filled in or edited if this feature is not enabled.

7. Enter the **Hostname/IP Address** for each server to be used.

   Note: Both IPv4 and IPv6 formats can be used.

   Note: If a duplicate IP address is used, you are prompted to use another IP address.

8. Set the **Severity** of the Trace Log to be forwarded from the list.
9. Click **Save** to save the changes to the system.

Chapter 7

Viewing Active BoD Sessions

Topics:

This chapter describes the Session Viewer tab of the BoD Administration page. The Session Viewer tab displays sessions that are currently active in the Bandwidth on Demand (BoD) application. A session represents a Quality of Service (QoS) enhancement to a subscriber's broadband experience that is managed by the BoD application.

About BoD Sessions

Each session can have up to two PCMM gates. These gates are used to characterize traffic for a subscriber's upstream and downstream traffic.

When a session is created, the BoD application initiates PCMM requests to a Multimedia Policy Engine (MPE) device. If the MPE device allows the upgrade requests to occur (based on its current policy rules in place), it interacts with a cable modem termination system (CMTS), attempting to actually reserve the necessary resources that carry out the service enhancement for the subscriber. The exact QoS characteristics of the service enhancement are defined in a service profile. If this request is successfully carried out, or under certain error conditions might be carried out in the future, a session is considered to be in an active state.

Viewing General Active Session Information

To view general session information:

1. From the **BoD** section of the navigation pane, select **Configuration.**
   The content tree displays a list of BoD servers.
2. From the content tree, select a **BoD server**.
   The **BoD Administration** page opens in the work area.
3. On the **BoD Administration** page, select the **Session Viewer** tab.
   The View Active Sessions page opens.

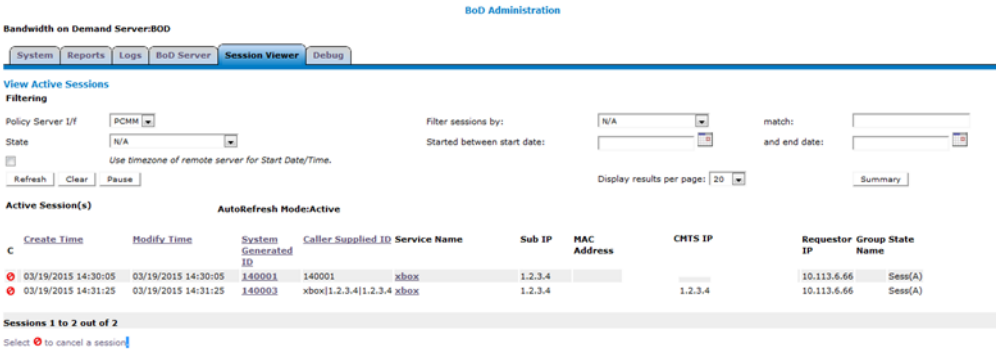


Figure 12: Session Viewer Page

This page contains the following information:

- **Create Time** — Date and time the session was created.
- **Modify Time** — Date and time the session was modified.
- **System Generated ID** — BoD generated identifier used as a unique identifier assigned to the session. This ID can be used to display detailed information about the session.
- **Caller Supplied ID** — Caller-supplied ID which allows the caller to assign a unique identifier to this BoD session for reporting and tracking purposes.
- **Service Name** — Name of the service profile associated with this session.

- **Sub IP** — Subscriber's IP address
- **MAC Address** — Subscriber's MAC address (if provided in the request).
- **CMTS IP** -- (If system is set to CMTS IP enabled) - Shows the IP Address for the CMTS.
- **Requestor IP** — Requestor's IP address
- **Group Name** — Caller-supplied group name, which allows the caller to group sessions based on a unique group name for reporting and tracking purposes.
- **State** — Current state of the session.

4. Use filtering to determine how the sessions are displayed on the page.
   a) Select the type of filtering, based on text matches of the options in the **Filter sessions by** pull-down list. You can select the following:
      - Subscriber IP
      - Caller Supplied ID
      - System Generated ID
      - Requestor IP
      - Group Name
      - MAC address
      - Upgate ID
      - Downgate ID (N/A)
   b) Enter the text to match in the **match** field.
   c) Select the desired state from the **State** pull-down list. You can select the following:
      - Processing req. received
      - Resolving MAC Address
      - Establishing Session
      - Re-establishing Session
      - Active Sessions (N/A)
   d) To change the start and end dates and times, click on the calendar next to the start and end date fields. The calendar and clock appear. Select the dates to start and end the session interval, enter the desired time, and click **Enter**.
   e) Click **Use timezone of remote server for Start Date/Time** to filter by the timezone of the remote server.
   f) Select the number of results to display on the page from the **Display Results per Page** drop-down list.
   g) After entering the filtering criteria, click **Refresh** to update the page with the new criteria.
      Click **Clear** to re-set the filtering criteria to the default values. Click **Resume** to start the auto-refresh functionality.

Viewing Active Session Summary Information

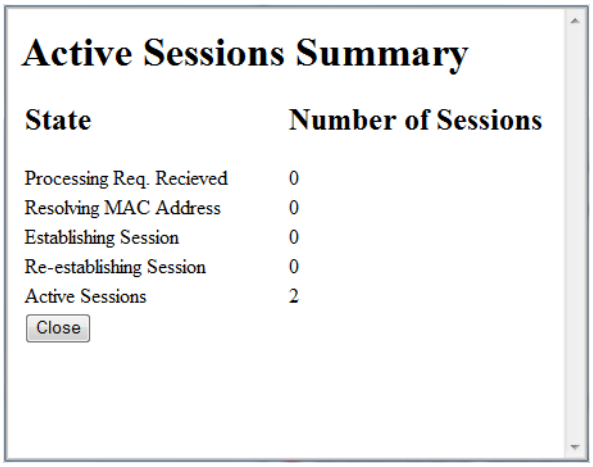
To display active session summary information:

1. From the **BoD** section of the navigation pane, select **Configuration**.
   The content tree displays a list of the BoD servers.

2. From the content tree, select the **BoD** server to be viewed.
   The **BoD Administration** page opens in the work area.
3. On the **BoD Administration** page, select the **Session Viewer** tab.
   The BoD Administration page displays active sessions page information.
4. Click **Summary**.

   The Active Sessions Summary window opens.

   Figure 13: Active Sessions Summary Window



   Where:

   - **Processing Req. Received** — Total number of sessions the BoD system has received (HTTP or SOAP requests) but no further processing has been performed.
   - **Resolving MAC Address** — Total number of sessions where the BoD system is attempting to resolve MAC address translation.
   - **Establishing Session** — Total number of sessions where the BoD application is attempting to establish a first time session.
   - **Re-Establishing Session** — Total number of sessions where the BoD is attempting to re-establish a session after an interruption.
   - **Active Sessions** — Total number of sessions were all associated flows are successfully established.

5. Click **Close**.

Cancelling an Active Session

Canceling a session causes the BoD application to attempt to remove the service enhancement for the subscriber. If the attempt is successful, any PCMM gates created on the CMTS on behalf of the subscriber are removed. If removal fails, the BoD application retries the removal in the future, and leaves the session in the active state. The BoD application eventually stops trying to remove the session after a certain number of attempts.

Note: You must have the appropriate privilege to delete a session. Refer to the *Configuration Management Platform Cable User Guide* for assistance.

To cancel an active session:

1. From the **BoD** section of the navigation pane, select **Configuration.**
   The content tree displays a list of the BoD servers.
2. From the content tree, select a BoD server.
   The BoD Administration page opens in the work area.
3. On the BoD Administration page, select the **Session Viewer** tab.
   The BoD Administration page displays active session information.
4. Select the red cancel icon , located next to the desired session to cancel.
   You are asked 'Are you sure you want to cancel this session?'. Click **OK** to cancel the session and remove it from the list of active sessions (or click **Cancel** to cancel your request).

Viewing Detailed Session Information

To view detailed session information:

1. From the **BoD** section of the navigation pane, select **Configuration.**
   The content tree displays a list of the BoD servers.
2. From the content tree, select a BoD server.
   The **BoD Administration** page opens in the work area.
3. On the BoD Administration page, select the **Session Viewer** tab.
   The **BoD Administration** page displays active session information.
4. Click the **System Generated ID** of the desired session.
   The Session Details page opens.

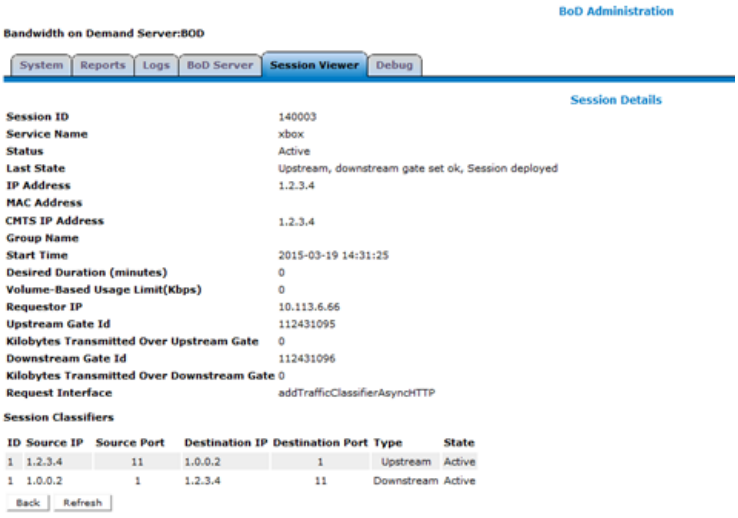


Figure 14: Session Details Page

The Session Details page is divided into the following sections:

- Session Details — Displays general information regarding the session.

- Scheduled Activity for Session (only displayed for active sessions) — Displays the next scheduled activity to occur for the session (for example, a gate info), and when the activity occurs.
- Session Classifiers — Displays the current classifiers associated with this session across its upstream and/or downstream gates, along with the state of the classifiers.

The Session Details page contains the following fields:

- **Session Id** — An ID for the session, created by the user or generated by the system.
- **Service Name** — The name of the service profile used to create the session.
- **Status** — The overall condition of a session, indicating if the session is active, or completed.
- **Last State** — BoD maintains an internal state machine for each active session. This machine manages session set-up, session tear-down, gate info requests, and gate recreation when gates are missing. This field indicates the current state of the session in the state machine.
- **IP Address** — The IP address of the subscriber to which the service enhancement applies.
- **MAC Address** — The MAC address of the subscriber (if provided in the request).
- **CMTS IP Address** (Only if CMTS IP is enabled. See Global Configuration Settings in the *CMP Cable User's Guide)* -- Specifies the IP address of the CMTS.
- **Group Name** — The group name associated with this session.
- **Start Time** — The start date and time of the session.
- **Desired Duration (minutes)** — The desired duration for the session (how long the QoS adjustment should be activated for the subscriber). This is a configurable setting in the service profile, where an explicit duration can be specified, or it could be a parameterized value.
- **Volume-Based Usage Limit (Kbps)** — The amount of data that can be transmitted over the Gate before meeting a volume threshold.
- **Requestor IP** — The IP address of the caller who initiated the request for the QoS upgrade for this subscriber.
- **Upstream Gate Id** — The PCMM gate ID for the upstream gate created for the subscriber. This ID is returned from the CMTS device as a response to a Gate Set request and is blank if the service profile associated with this session does not have an upstream profile configured.
- **Kilobytes Transmitted Over Upstream Gate** — The number of kilobytes transmitted over the upstream gate during the course of this session being active. BoD periodically performs a PCMM GateInfo of gates created for a subscriber. This field displays a running total of the kilobytes transmitted. If gates have been recreated for a subscriber during the course of a session, this field will reflect kilobytes transmitted across ALL upstream gates that have been created for this subscriber during the course of this session.
- **Downstream Gate Id** — The PCMM gate ID for the downstream gate created for the subscriber. This ID is returned from the CMTS device as a response to a Gate Set request and is blank if the service profile associated with this session does not have a downstream profile configured.
- **Kilobytes Transmitted Over Downstream Gate** — The number of kilobytes transmitted over the downstream gate during the course of this session being active. BoD periodically performs a PCMM GateInfo of gates created for a subscriber. This field displays a running total of the kilobytes transmitted. If gates have been recreated for a subscriber during the course of a session, this field will reflect kilobytes transmitted across ALL downstream gates that have been created for this subscriber during the course of this session.
- **Request Interface** — The interface this session was created through. Sessions can be created through the HTTP interface or the SOAP interface.

Viewing Scheduled Activity for Session

The Scheduled Activity for Session section displays only when a session is in an active state. The BoD server maintains a state machine for all active sessions. This section provides an indication of the next activity that is taken on behalf of this session by the BoD server. This section indicates what type of task and the date and time that this task is to be executed.

Viewing Session Classifiers

The Session Classifier section displays the current classifiers associated to this session across its upstream and/or downstream gates, along with the state of the classifiers. A session may have multiple sets of classifiers that may have been activated and deactivated during the sessions time of activity. Refer to *Creating a Service* for further information on traffic classifiers.

Chapter 8

Managing PCMM Services

Topics:

The PCMM Services management features allow you to view, create, modify, and delete services.

Note: The services displayed in the examples in this chapter are preconfigured. The services shown in your application may vary.

About PCMM Services

Services are the Quality of Service (QoS) definitions applied to the controls of service flows within a CMTS. These services define the QoS and IP flow information for the Gates setup for a subscriber connection on a cable modem termination system (CMTS), where the PacketCable Multimedia (PCMM) gate is a logical representation of a policy decision installed on the CMTS. The PCMM gate is used to control access by a single IP flow to enhanced QoS.

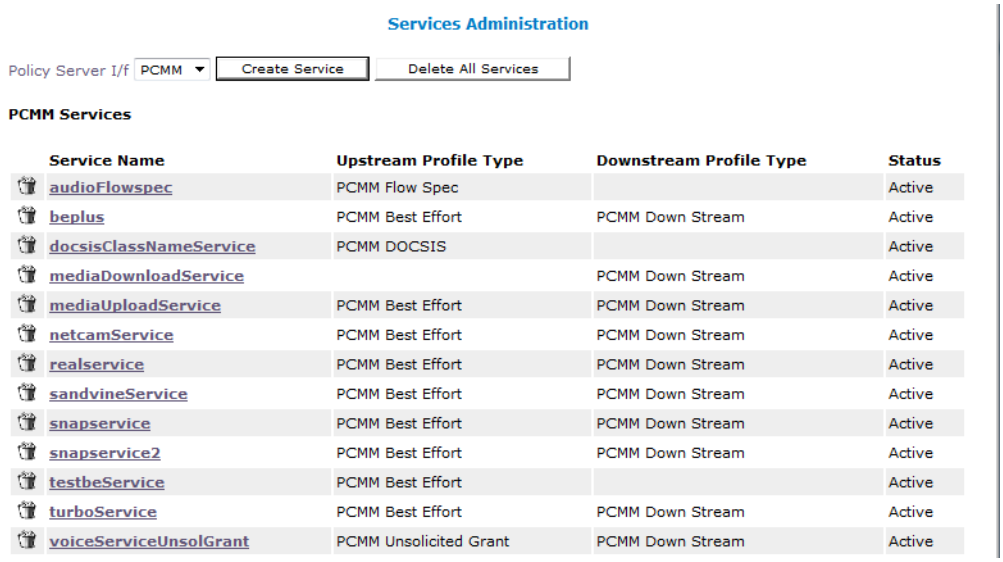
Viewing Services

The **Services Administration** page displays the names and information for the PCMM services.

From this page, you can modify the parameters contained in a service, copy the parameters to a new service, delete the service from the BoD application, or deactivate the service.

To view currently configured services:

1. From the **BoD** section of the navigation pane, select **Services**.
   The **Services Administration** page opens in the work area.



Service Name	Upstream Profile Type	Downstream Profile Type	Status
audioFlowspec	PCMM Flow Spec		Active
beplus	PCMM Best Effort	PCMM Down Stream	Active
docsisClassNameService	PCMM DOCSIS		Active
mediaDownloadService		PCMM Down Stream	Active
mediaUploadService	PCMM Best Effort	PCMM Down Stream	Active
netcamService	PCMM Best Effort	PCMM Down Stream	Active
realservice	PCMM Best Effort	PCMM Down Stream	Active
sandvineService	PCMM Best Effort	PCMM Down Stream	Active
snapservice	PCMM Best Effort	PCMM Down Stream	Active
snapservice2	PCMM Best Effort	PCMM Down Stream	Active
testbeService	PCMM Best Effort		Active
turboService	PCMM Best Effort	PCMM Down Stream	Active
voiceServiceUnsolGrant	PCMM Unsolicited Grant	PCMM Down Stream	Active

Figure 15: Services Administration Page

The Services Administration page displays the following fields:

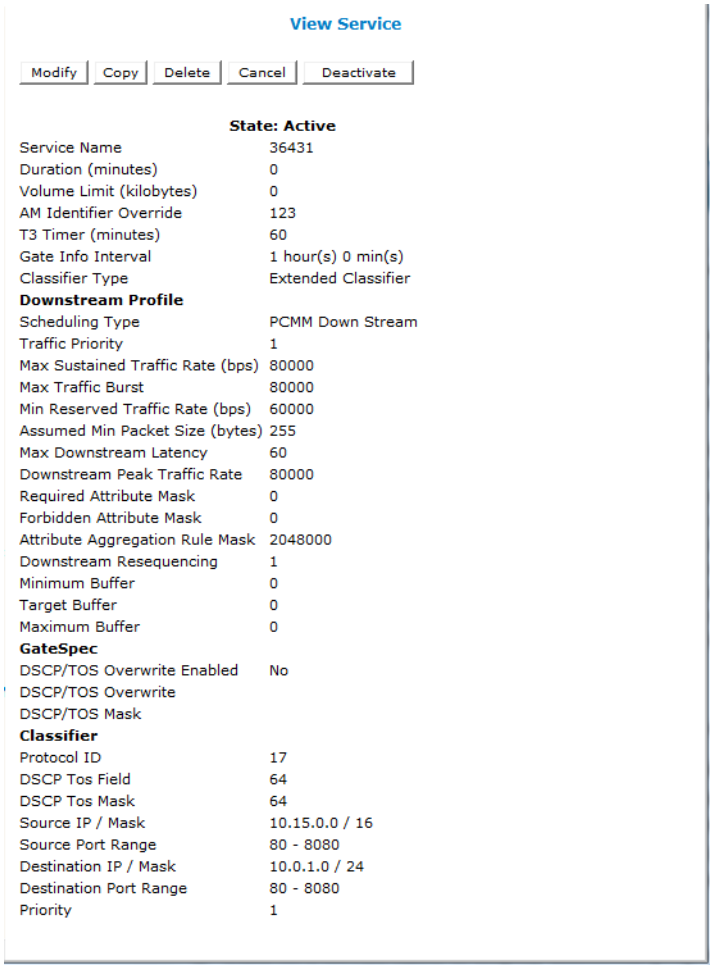
- **Service Name** — User-supplied unique identifiable name for the service.
- **Upstream Profile Type** — Traffic profile associated with a gate through a FlowSpec, DOCSIS Service Class Name, or a DOCSIS specific parameterization scheme.
- **Downstream Profile Type** — Traffic profile associated with a gate through a FlowSpec, DOCSIS Service Class Name, or a DOCSIS specific parameterization scheme.
- **Status** — Status of the service.

2. Click on a service name to display additional properties associated with the service.

   The **View Service** page opens in the work area.

   Note: The fields shown on the **View Service** page depend on the values selected for the **Classifier Type** and **Scheduling Type** fields. The fields in your service may differ from this example. See *Table 4: Service Field Definitions and Ranges* for a description of each field.

   Figure 16: View Service Page



Creating a Service

You can create a service using the **Create Service** button or by copying an existing service.

In addition, you can create (or modify) a PCMMService that contains multiple Classifiers at the same time. A PCMMService can contain multiple Classifiers which means that you can create sessions with multiple standard /extended/ipv6 Classifiers.

Note: Only extended/ipv6 types can be mixed.

You can also configure Services with multiple Classifiers in each direction (Upstream or Downstream) and add or remove Classifiers from Services.

Creating a Service

To create a service:

1. From the **BoD** section of the navigation pane, select **Services**.
   The **Services Administration** page opens in the work area.
2. Click **Create Service**.
   The **Create Service** page opens in the work area.

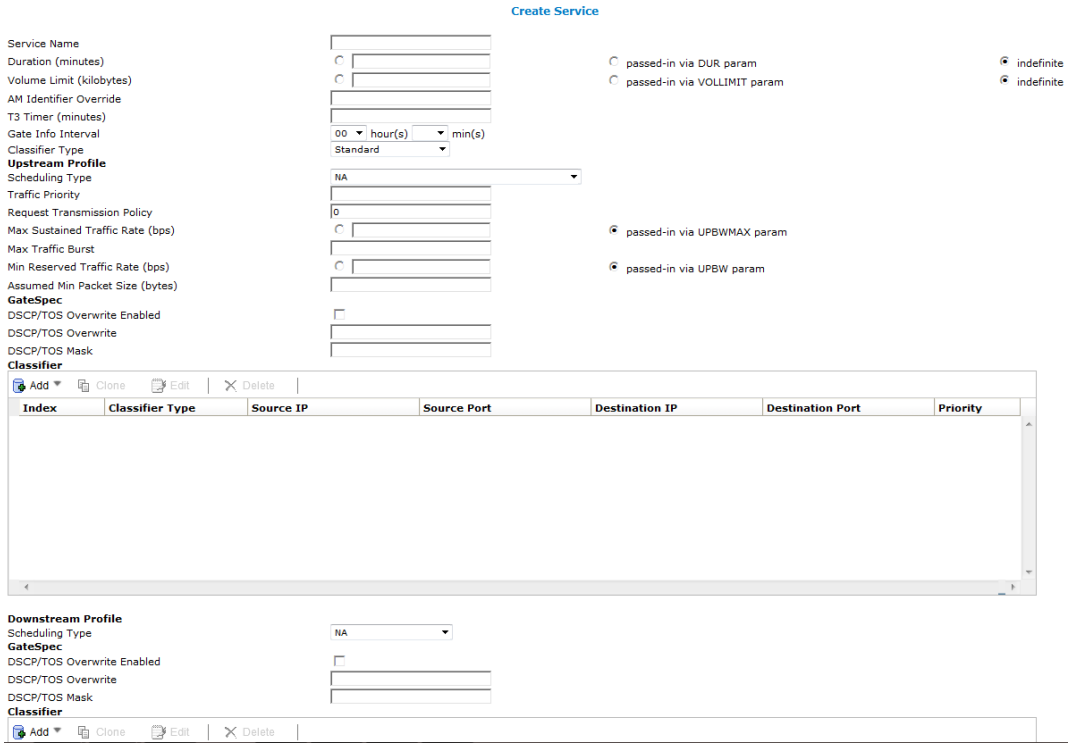


Figure 17: Create Service Page

3. Enter a **Service Name**.
4. Configure the service. See *Table 4: Service Field Definitions and Ranges*
5. Configure an **Upstream Profile**. See *Adding Upstream and Downstream Profiles*.
6. Configure the **GateSpecs**. See *Adding Gate Specifications*.
7. Add a **Classifier**. See *Adding Classifiers*.
8. Create the **Upstream Profile**. See *Table 4: Service Field Definitions and Ranges*.
9. When you finish, click **Save**.

Adding Upstream and Downstream Profiles

To create upstream and downstream profiles for a new service:

Note: The parameters stated here are for both upstream and downstream profiles.

1. From the Create Service screen scroll to the **Upstream Profile** section.

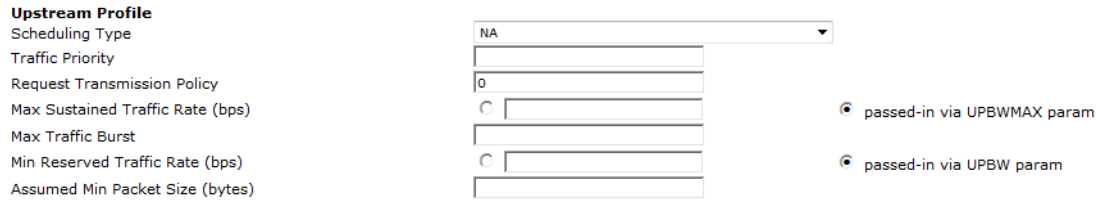


Figure 18: Upstream Profile

Note: For information on entering values, see *Table 4: Service Field Definitions and Ranges*.

2. Select the **Scheduling Type** from the pull-down menu.
3. Type in a **Traffic Priority**.
4. Set the: **Request Transmission Policy**.
5. Set the **Max Sustained Traffic Rate**.
6. Set the **Max Traffic Burst**.
7. Set the **Min Reserved Traffic Rate**.
8. Set the **Assumed Min Packet Size**.
9. Associate the stream with a **Classifier**.

Adding Gate Specifications

To set the gate specifications for a service:

1. From the **Create Service** screen, scroll down to the **GateSpec** section.

   Note: The parameters for the gate specifications are the same for both upstream and downstream profiles.

   For specific information on each field for a gate, see *Table 4: Service Field Definitions and Ranges*.

Figure 19: GateSpec Section

2. (Optional) Select if the **DSCP/TOS Overwrite Enabled** is active or not.
3. Enter the **DSCP/TOS Overwrite** period.
4. Enter the **DSCP/TOS Mask** period.
5. Associate the **GateSpec** with a classifier(s).

Adding Classifiers

To add one or more classifiers:

1. From the **Create Service** page click the **Add** button on the Classifier tool bar.
   The **Classifier** drop-down menu opens.

Note: This process is for adding both upstream and downstream classifiers.

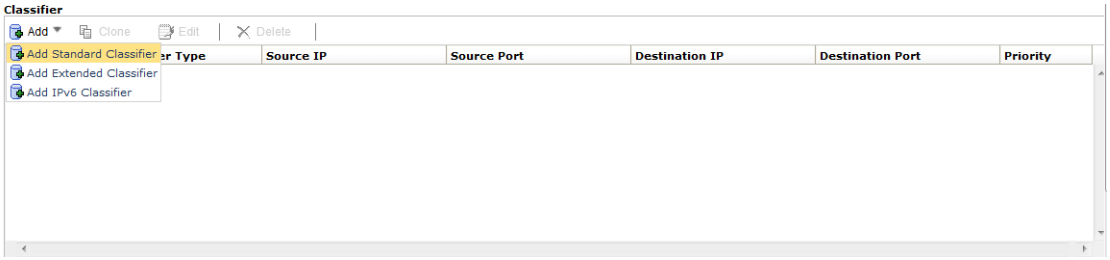


Figure 20: Classifier Table with Drop-down Menu

2. Select the **Classifier**.
   The **Add Classifier** window opens.

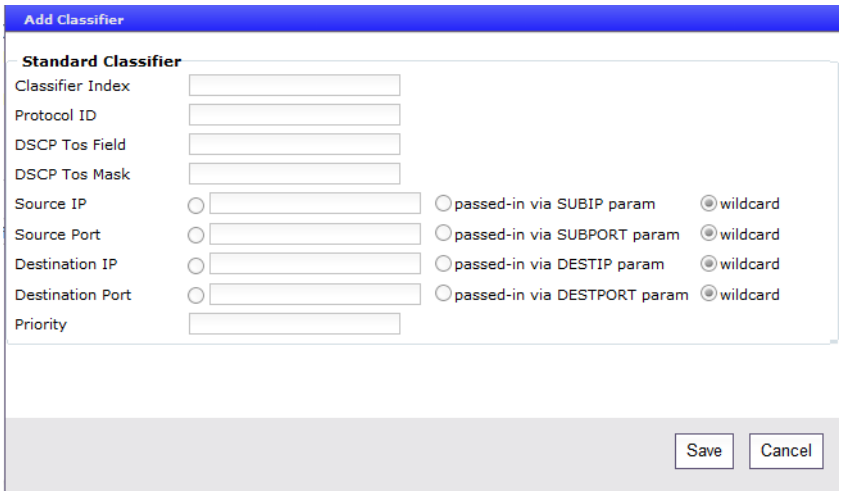


Figure 21: Add Classifier Screen (Standard Classifier selected)

3. Type in the appropriate values in each of the fields, See *Table 4: Service Field Definitions and Ranges*.
4. Click the **Save** button to save the settings.
   You are returned to the **Create Service** screen.
5. Repeat steps *Step 1* through *Step 4* to add additional classifiers.

   Note: For classifier compatibility. See *Compatibility Matrix for Creating Multiple Classifiers*.

Correct Parameter and Classifier Order

The parameter/classifier order should match the guidelines for the PCMMService. For example, if two standard Classifiers are pre-defined in the PCMMService profile illustrated in this example:

```
http:bodurl/bod/createsession.do?Servicename=123&Subip=x1.x1.x1.x1|x2.x2.x2.x2&Destip=
d1.d1.d1.d1|d2.d2.d2.d2&subport=subport1|subport2&destport=destport1|destport2
```

Then the correct parameter order should follow the Classifier order.

Table 2: Correct Parameter - Classifier Order

	Classifier index1	Classifier index2

Subip	x1.x1.x1.x1	x 2. x 2. x 2. x 2
Destip	d1 . d1 . d1 . d1	d2 . d2 . d2 . d2
Subport	subport1	subport2
Destport	destport1	destport2

Stated in a different way:

The parameters from Subip_list[0], Destip[0],subport[0] and destport[0] belong to the Classifier1.

The parameters from Subip_list[1], Destip[1],subport[1] and destport[1] belong to the Classifier2.

Compatibility Matrix for Creating Multiple Classifiers

This table provides the compatibility rules for creating session interfaces with multiple classifiers to the PCMMService:

Note: Parameter values used for multiple Classifiers are formatted and interpreted similarly to the single classifier createSession.do. However, parameters related to the Classfier(s) for the service will include multiple values, separated by a pipe ("|") character, and ordered corresponding to the Index values in the Service Definition. If a given parameter is not required for one Classifier in a multiple-Classifier Service, then the corresponding parameter should be omitted but the pipe character is still required.

Table 3: Multiple Classifier Compatibility Matrix

Classifier Type	Classifier Type	Classifier Compatibility
Standard	Standard	Yes
Standard	Extended	No
Standard	Ipv6	No
Extended	Extended	Yes
Ipv6	Ipv6	Yes
Extended	Ipv6	Yes

Creating a Service from an Existing Service

To create a service from an existing service:

1. From the **BoD** section of the navigation pane, select **Services**.
   The **Service Administration** page opens in the work area.
2. Select a service.
   The **View Service** page opens in the work area.
3. Click **Copy**.
   The **Copy Service** page opens in the work area.

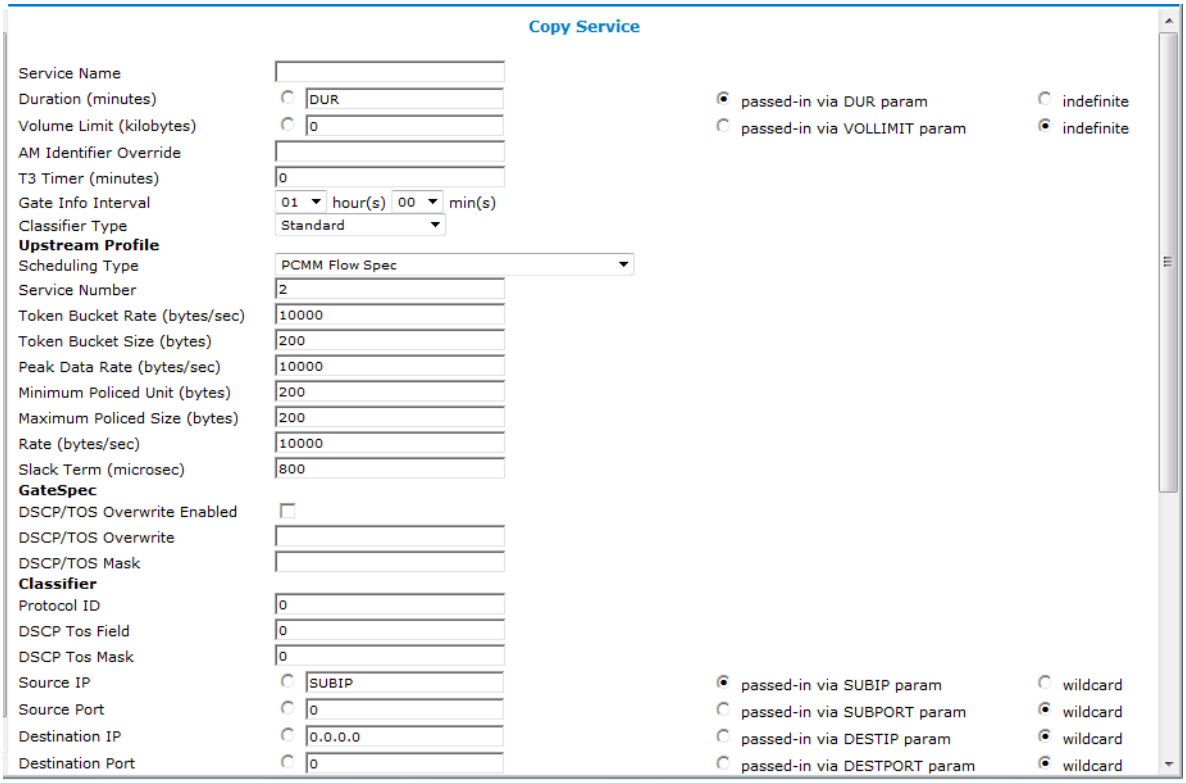


Figure 22: Copy Service Page

4. Enter a new service name for the service, and edit the fields. For a description of each field and its associated values, see the *Table 4: Service Field Definitions and Ranges*.
5. When you finish, click **Save**.

Modifying a Service

To modify a service:

1. From the **BoD** section of the navigation page, select **Services**.
   The **Services Administration** page opens in the work area.
2. Select a service.
   The **View Service** page opens in the work area.
3. Click **Modify**.
   The **Edit Service** page opens in the work area.
4. Edit the fields.
   See *Table 4: Service Field Definitions and Ranges* for descriptions of the fields used to edit a PCMM service. The fields available depend on the values selected for the **Classifier Type** and **Scheduling Type** fields.

Table 4: Service Field Definitions and Ranges

Field	Description	Valid Value
Service Name	A unique name for the PCMM service.	A string of up to 32 characters.
Duration (minutes)	The duration this service is deployed. Enter a value or select **passed-in via DUR param** or **indefinite**.	Integer (0 - 35791394)
Volume Limit (kilobytes)	The volume limit. Enter a value or select **passed-in via VOLLIMIT param** or **indefinite**.	Long (0 - 9223372036854775807)
AM Identifier Override	An override for the BoD AM ID. This value overrides the value set in the **Configure Settings AM Identifier** field.	Long (0 - 4294967295)
T3 Timer (minutes)	The T3 inactivity timer value.	Integer (1 - 1092.25)
Gate Info Interval	The gate information interval, in hours and minutes.	4-byte unsigned integer
Classifier Type	Classifier type. Depending on what you select here, other fields may become available on the page.	• Standard • Extended Classifier • IPv6 Classifier • Mixed Classifier
Upstream Profile		
Scheduling Type	The traffic profile type. Depending on what you select here, other fields may become available on the page.	• NA • PCMM Best Effort • PCMM Real-Time Polling • PCMM Non-Real-Time Polling • PCMM Unsolicited Grant • PCMM Unsolicited Grant with Activity Detection • PCMM Upstream Drop • PCMM Flow Spec • PCMM DOCSIS
Traffic Priority	The relative priority assigned to the service flow in comparison with other flows.	1 byte (0 - 255)
Request Transmission Policy	Specifies which IUC opportunities the CM uses for upstream transmission requests	4-byte (0 - 4294967295)

Field	Description	Valid Value
	and packet transmissions for this service flow.	
Max Sustained Traffic Rate (bps)	The rate parameter for a token-bucket-based rate limit for this service flow. Enter a value or select **passed-in via UPBWMAX param**.	4-byte (0 - 4294967295)
Max Traffic Burst	The token bucket size, in bytes, for a token-bucket-based rate limit for this service flow.	4-byte (0 - 4294967295)
Min Reserved Traffic Rate (bps)	The minimum rate reserved for this service flow. Enter a value or select **passed-in via UPBW param**.	4-byte (0 - 4294967295)
Assumed Min Packet Size (bytes)	The assumed minimum packet size for which the Minimum Reserved Traffic rate is provided for this service flow.	2 bytes (0 - 65535). Enter 0 if a specific Assumed Minimum Reserved Traffic Rate Packet size is not required. Upon receipt of a value of 0, the CMTS must utilize its implementation-specific default size for this parameter, not 0 bytes.
Maximum Concatenated Burst	The maximum concatenated burst, in bytes, that a service flow is allowed.	2-byte (0 - 65535)
Service Class Name	The DOCSIS Service Class to be used to describe QoS attributes.	32 characters
Service Number	The service number. A controlled load service must contain only the TSpec token bucket parameters, and not the RSpec. A guaranteed service must contain both the TSpec and the RSpec.	Short (0 - 255) 5 - controlled load 2 - guaranteed
Token Bucket Rate (bytes/sec)	Defines how traffic is injected into the network by the sending application.	Float (0.0 - 3.4028234663852886E38)
Token Bucket Size (bytes)	Controls the maximum amount of data that the flow can send at the peak rate.	Float (0.0 - 3.4028234663852886E38)
Peak Data Rate (bytes/sec)	The peak data rate.	Float (0.0 - 3.4028234663852886E38)

Field	Description	Valid Value
Minimum Policed Unit (bytes)	The minimum size of a packet that can be subject to policing.	Long (0 - 2147483647)
Maximum Policed Size (bytes)	The maximum size of a burst of data that can exceed the given bandwidth limit.	Long (0 - 2147483647)
Rate (bytes/sec)	The rate.	Float (0.0 - 3.4028234663852886E38)
Slack Term (microsec)	The slack term, corresponding to latency or jitter depending on the service.	Long (0 - 2147483647)
Envelope	The envelope types (i.e. Authorized, Reserved, and Committed) that are present in the object.	1 byte (0 - 255) A value of 1 indicates that the envelope type is present in the Traffic Profile.
Unsolicited Grant Size (bytes)	The grant size.	2-byte (0 - 65535) There is no default value.
Grants Per Interval	The number of grants per Nominal Grant Interval.	1 byte (0 - 255) There is no default value. A value of 1 is recommended.
Nominal Grant Interval	The nominal time, in microseconds, between successive data grant opportunities for a service flow.	4-byte (0 - 4294967295)
Tolerated Grant Jitter (microsec)	The maximum amount of time that transmission opportunities can be delayed from the nominal periodic schedule.	4-byte (0 - 4294967295)
Nominal Polling Interval (microsec)	The nominal interval between successive unicast request opportunities for this service flow on the upstream channel.	4-byte (0 - 4294967295)
Tolerated Poll Jitter (microsec)	The maximum amount of time that a polling request can be delayed.	Long (0 - 4294967295)
Required Attribute Mask	Limits the set of channels. Enter a value or select **passed-in via UPRAMASK param**.	4-byte (0 - 4294967295)
Forbidden Attribute Mask	Limits the set of channels and bonding groups to which the CMTS assigns the service flow	4-byte (0 - 4294967295)

Field	Description	Valid Value
	by forbidding certain attributes. Enter a value or select **passed-in via UPFAMASK param**.	
Attribute Aggregation Rule Mask	Guides the CMTS on how it can use the attribute masks of individual channels to construct a dynamic bonding group for this service flow. Enter a value or select **passed-in via UPAAMASK param**.	4-byte (0 - 4294967295)
Upstream Peak Data Rate	The peak traffic rate, in bits per second, that is allowed for a service flow. Enter a value or select **passed-in via UPPTR param**.	4-byte (0 - 4294967295)
Minimum Buffer	The lower limit for the size of the buffer to be provided for a service flow.	4-byte (0 - 4294967295)
Target Buffer	The desired value for the size of the buffer to be provided for a service flow.	4-byte (0 – 4294967295)
Maximum Buffer	The upper limit for the size of the buffer to be provided for a service flow.	4-byte (0 – 4294967295)
GateSpec		
DSCP/TOS Overwrite Enabled	Enables the DSCP/TOS Overwrite functionality. If this field is set, then the CMTS must mark the packets traversing the CMTS DSCP/TOS value. If the field is cleared, then the CMTS must not perform any marking.	Enabled/Disabled
DSCP/TOS Overwrite	Used to overwrite the DSCP/TOS field of packets associated with the DOCSIS Service Flow that corresponds to the Gate	1 byte (0 - 255)
DSCP/TOS Mask	The bit mask used to identify particular bits within the DSCP/TOS field	1 byte (0 - 255)
Classifier		
Protocol ID	The protocol identifier.	0 - 257

Field	Description	Valid Value
DSCP/Tos Field	The Differentiated Services Code Point (DSCP) or IP Precedence.	1 byte (0 - 255)
DSCP/Tos Mask	The bit mask used to select relevant bits from the accompanying DSCP/Tos field value.	1 byte (0 - 255)
Next Header Type	The desired next header type value for any header or extension header associated with the packet.	A value of 256 matches traffic with any IPv6 next header type value. A value of 257 matches both TCP and UDP traffic.
Traffic Class Mask	A mask defining which of the 8 bits should be used for matching a traffic class value.	1 byte (0 - 255).
Traffic Class Range	Enter a lower and upper value to match a range of traffic class values.	1 byte (0 - 255).
Flow Label	Contains valid data for comparison with the IPv6 Flow Label.	4 bit (0 - 15) This flag must be set to 1 if data is needed. When comparison of the IPv6 Flow Label for this entry is irrelevant then the flag cannot be set (value = 0). When the Flow Label flag is set to 0, the CMTS cannot include the IPV6 Flow Label field in the classifier. All other flags must be set to zero
Source IP (Standard classifier) Source IP / Mask (Extended classifier) Source IP / Prefix Length (IPv6 classifier)	The source IP address used to classify traffic. Enter a value, select that the values are passed in by parameters, or select **wildcard**.	4-octet IPv4 address (Standard classifier) 4-octet IPv4 mask (Extended classifier) 16-octet IPv6 address (IPv6 classifier) Prefix Length: Short (0 - 128)
Source Port (Standard classifier) Source Port Range (Extended and IPv6 classifiers)	The source port range used to classify traffic. Enter a value, select that the values are passed in by parameters, or select **wildcard**.	2 bytes 0 - 32768 (Standard classifier) 0 - 65535 (Extended classifier)

Field	Description	Valid Value
Destination IP (Standard classifier) Destination IP / Mask (Extended classifier) Destination IP / Prefix Length (IPv6 classifier)	The destination IP address used to classify traffic. Enter a value, select that the values are passed in by parameters, or select **wildcard**.	4-octet IPv4 address (Standard classifier) 4-octet IPv4 mask (Extended classifier) 16-octet IPv6 address (IPv6 classifier) Prefix Length: Short (0 - 128)
Destination Port (Standard classifier) Destination Port Range (Extended and IPv6 classifiers)	The destination port used to classify traffic. Enter a value, select that the values are passed in by parameters, or select **wildcard**.	2 bytes 0 - 32768 (Standard classifier) (0 - 65535) Extended classifier
Priority	The priority level for this classifier.	Current DOCSIS supported values are 64 - 191.
Downstream Profile		
Scheduling Type	The downstream profile type. Depending on what you select here, other fields may become available on the page.	• NA • PCMM Down Stream • PCMM DOCSIS • PCMM Flow Spec
Traffic Priority	The relative priority assigned to the service flow in comparison with other flows.	1 byte (0 - 255)
Max Sustained Traffic Rate (bps)	The rate parameter for a token-bucket-based rate limit for this service flow. Enter a value or select **passed-in via DOWNBWMAX param**.	4-byte (0 - 4294967295)
Max Traffic Burst	The token bucket size, in bytes, for a token-bucket-based rate limit for this service flow.	4 bytes (0 - 4294967295)
Downstream Peak Traffic Rate	The rate parameter of a token-bucket-based peak rate limiter for packets of a downstream service flow. Enter a value or select **passed-in via DOWNPTR param**.	4-byte (0 - 4294967295)
Min Reserved Traffic Rate (bps)	The minimum rate reserved for this service flow. Enter a value or select **passed-in via DOWNBW param**.	4-byte (0 - 4294967295)

Field	Description	Valid Value
Assumed Min Packet Size (bytes)	The assumed minimum packet size for which the Minimum Reserved Traffic rate is provided for this service flow.	2-byte (0 - 65535). Enter 0 if a specific Assumed Minimum Reserved Traffic Rate Packet size is not required. Upon receipt of a value of 0, the CMTS must utilize its implementation-specific default size for this parameter, not 0 bytes.
Max Downstream Latency	The maximum latency between the receptions of a packet on the CMTS's NSI and the forwarding of the packet on its RF interface.	4 byte (0 - 4294967295)
Service Class Name	The pre-configured service class name associated with a gate.	32 characters
Service Number	The service number. A controlled load service must contain only the TSpec token bucket parameters, and not the RSpec. A guaranteed service must contain both the TSpec and the RSpec.	Short (0 - 255) 5 - controlled load 2 - guaranteed
Token Bucket Rate (bytes/sec)	Defines how traffic is injected into the network by the sending application.	Float (0.0 - 3.4028234663852886E38)
Token Bucket Size (bytes)	Controls the maximum amount of data that the flow can send at the peak rate.	Float (0.0 - 3.4028234663852886E38)
Peak Data Rate (bytes/sec)	The peak data rate.	Float (0.0 - 3.4028234663852886E38)
Minimum Policed Unit (bytes)	The minimum size of a packet that can be subject to policing.	Long (0 - 2147483647)
Maximum Policed Size (bytes)	The maximum size of a burst of data that can exceed the given bandwidth limit.	Long (0 - 2147483647)
Rate (bytes/sec)	The rate.	Float (0.0 - 3.4028234663852886E38)
Slack Term (microsec)	The slack term, corresponding to latency or jitter, depending on the service.	Long (0 - 2147483647)

Field	Description	Valid Value
Required Attribute Mask	Limits the set of channels. Enter a value or select **passed-in via DOWNRAMASK param**.	4-byte (0 - 4294967295)
Forbidden Attribute Mask	Limits the set of channels. Enter a value or select **passed-in via DOWNRAMASK param**.	4-byte (0 - 4294967295)
Attribute Aggregation Rule Mask	Guides the CMTS on how it can use the attribute masks of individual channels to construct a dynamic bonding group for this service. Enter a value or select **passed-in via DOWNRAMASK param**.	4-byte (0 - 4294967295)
Downstream Resequencing	Specifies the use of sequence numbers in downstream DOCSIS 3.0 service flows.	1 byte (0 - 255)
Minimum Buffer	The lower limit for the size of the buffer to be provided for a service flow.	4-byte (0 - 4294967295)
Target Buffer	The desired value for the size of the buffer to be provided for a service flow.	4-byte (0 - 4294967295)
Maximum Buffer	The upper limit for the size of the buffer to be provided for a service flow.	4-byte (0 - 4294967295)

Note: The**passed-in** radio button indicates the field obtains its value from a passed-in HTTP parameter from the create session or add traffic classifier request. The use of the**Wildcard** radio button indicates to accept or match all and on a duration field indicates an indefinite duration. Refer to the PCMM specification for details for each wildcard value.

5. When you finish, click **Save**.

Deactivating a Service

Deactivating a service prevents any new requests for this service from establishing QoS. Any existing sessions will not be affected.

To deactivate a service:

1. From the **BoD** section of the navigation pane, select **Services**.
   The Services Administration page opens in the work area.
2. Click on the desired service.
   The View Service page opens in the work area.

3. Click **Deactivate**.

   The service is deactivated and the **Deactivate** button is replaced by an **Activate** button, which is used to reactivate the service.

Deleting a Service

Deleting a service prevents any new requests for this service from establishing QoS. Any existing sessions will not be affected.

To delete a service:

1. From the **BoD** section of the navigation pane, select **Services**.

   The **Services Administration** page opens in the work area.

2. Select a service.

   The **View Service** page opens in the work area.

3. Click **Delete**.

   A confirmation message appears.

4. Click **OK** to delete the service.

The service is deleted.

Importing a Service

The Import Services page allows an XML file containing services definitions to be imported into the BoD application. By clicking **Browse**, a file chooser dialog box is opened. The operator can then choose a file to import into the BoD application. If the imported file passes validation, all the Service definitions currently defined in the BoD application are replaced with the Service definitions defined in the import file. This feature is typically used in conjunction with the Export Services feature.

Note: Attributes upRequiredAttrMask, upForbiddenAttrMask, upAttrAggrRuleMask, downRequiredAttrMask, downForbiddenAttrMask, downAttrAggrRuleMask, and downPeakTrafficRate are expected in a SOAP or HTTP request for gates. If these attributes are not included, the request will fail. Therefore, after importing an XML file from previous versions of the BoD application, ensure that the service has included the new attributes. To configure these attributes in the BoD application, see *Modifying a Service*.

To import a service:

1. From the **BoD** section of the navigation pane, select **Services Import/Export**.

   The **Import/Export** page opens in the work area.

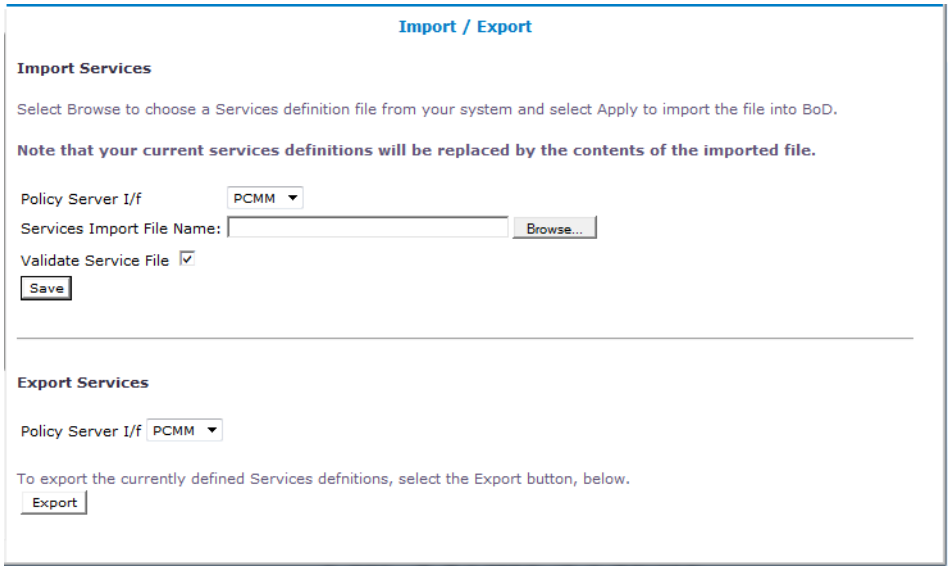


Figure 23: Services Import/Export Page

2. Enter the filename to import in the **Services Import File Name** field or click **Browse** to locate the desired file.
3. To have the file validated before import, click **Validate Service File**.
4. After you have entered the desired information, click **Save** to import the file.

Exporting a Service

The Export Services page invokes a secondary browser window containing an XML document that reflects the currently defined service profile definitions configured within the BoD AM application. This content can be saved by choosing the **File** > **Save As...** menu item in this secondary browser window. This feature can also be used to save snapshots of service definitions, which can be later imported back into the BoD AM application using the Import Services feature.

To export a Service:

1. From the **BoD** section of the navigation screen, select **Services Import/Export**.
   The Services Import/Export page opens in the work area.
2. Click **Export**.
3. You are prompted to open or save the file.
4. Click **Open**. An XML file appears.
5. Select **Save As ...** from the **File** pull-down menu.
6. Enter the location to export the file and click **Save**.
   The file exports to the entered location.

Appendix A

BoD AM WSDL Definitions

Topics:

This appendix shows the Bandwidth on Demand Application Manager (BoD AM) WSDL script files.

BoD AM WSDL Definitions

The following WSDL script defines the BoD AM SOAP API.

```
<?xml version='1.0' encoding='UTF-8'?><wsdl:definitions
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://www.camiant.com/wsdl"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:ns1="http://schemas.xmlsoap.org/soap/http"
name="BodSessionManagementService" targetNamespace="http://www.camiant.com/wsdl">

    <wsdl:types>
<xs:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://www.camiant.com/wsdl"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:ns1="http://schemas.xmlsoap.org/soap/http"
targetNamespace="http://www.camiant.com/wsdl" version="1.0">
  <xs:element name="addTrafficClassifier">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="String_1" type="xs:string" />
     <xs:element name="String_2" type="xs:string" />
     <xs:element name="String_3" type="xs:string" />
     <xs:element name="String_4" type="xs:string" />
     <xs:element name="int_5" type="xs:int" />
     <xs:element name="String_6" type="xs:string" />
     <xs:element name="int_7" type="xs:int" />
     <xs:element name="boolean_8" type="xs:boolean" />
     <xs:element name="cmtsIPAddress" type="xs:string"  maxOccurs="1"
minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="addTrafficClassifierResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="result" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="addTrafficClassifierWithExtClassifier">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="String_1" type="xs:string" />
     <xs:element name="String_2" type="xs:string" />
     <xs:element name="String_3" type="xs:string" />
     <xs:element name="String_4" type="xs:string" />
     <xs:element name="int_5" type="xs:int" />
     <xs:element name="String_6" type="xs:string" />
     <xs:element name="int_7" type="xs:int" />
     <xs:element name="boolean_8" type="xs:boolean" />
     <xs:element name="String_9" type="xs:string" />
     <xs:element name="String_10" type="xs:string" />
     <xs:element name="String_11" type="xs:string" />
     <xs:element name="int_12" type="xs:int" />
     <xs:element name="int_13" type="xs:int" />
    <xs:element name="cmtsIPAddress" type="xs:string" maxOccurs="1" minOccurs="0"
```

```
/>
     </xs:sequence>
   </xs:complexType>
 </xs:element>
 <xs:element name="addTrafficClassifierWithExtClassifierResponse">
   <xs:complexType>
     <xs:sequence>
       <xs:element name="result" type="xs:string"/>
     </xs:sequence>
   </xs:complexType>
 </xs:element>
 <xs:element name="addTrafficClassifierWithIPv6Classifier">
   <xs:complexType>
     <xs:sequence>
       <xs:element name="String_1" type="xs:string"/>
       <xs:element name="String_2" type="xs:string"/>
       <xs:element name="String_3" type="xs:string"/>
       <xs:element name="String_4" type="xs:string"/>
       <xs:element name="int_5" type="xs:int"/>
       <xs:element name="String_6" type="xs:string"/>
       <xs:element name="int_7" type="xs:int"/>
       <xs:element name="boolean_8" type="xs:boolean"/>
       <xs:element name="String_9" type="xs:string"/>
       <xs:element name="short_10" type="xs:short"/>
       <xs:element name="short_11" type="xs:short"/>
       <xs:element name="int_12" type="xs:int"/>
       <xs:element name="int_13" type="xs:int"/>
     </xs:sequence>
   </xs:complexType>
 </xs:element>
 <xs:element name="addTrafficClassifierWithIPv6ClassifierResponse">
   <xs:complexType>
     <xs:sequence>
       <xs:element name="result" type="xs:string"/>
     </xs:sequence>
   </xs:complexType>
 </xs:element>
 <xs:element name="createSession">
   <xs:complexType>
     <xs:sequence>
       <xs:element name="String_1" type="xs:string" />
       <xs:element name="String_2" type="xs:string" />
       <xs:element name="String_3" type="xs:string" />
       <xs:element name="int_4" type="xs:int" />
       <xs:element name="int_5" type="xs:int" />
       <xs:element name="int_6" type="xs:int" />
       <xs:element name="int_7" type="xs:int" />
       <xs:element name="int_8" type="xs:int" />
       <xs:element name="String_9" type="xs:string" />
       <xs:element name="String_10" type="xs:string" />
       <xs:element name="int_11" type="xs:int" />
       <xs:element name="String_12" type="xs:string" />
       <xs:element name="int_13" type="xs:int" />
       <xs:element name="String_14" type="xs:string" />
       <xs:element name="int_15" type="xs:int" />
       <xs:element name="int_16" type="xs:int" />
       <xs:element name="boolean_17" type="xs:boolean" />
       <xs:element name="vollimitKbps" type="xs:long" />
       <xs:element name="upRequiredAttrMask" type="xs:long" />
       <xs:element name="upForbiddenAttrMask" type="xs:long" />
       <xs:element name="upAttrAggrRuleMask" type="xs:long" />
       <xs:element name="upPeakTrafficRate" type="xs:long" />
       <xs:element name="downRequiredAttrMask" type="xs:long" />
       <xs:element name="downForbiddenAttrMask" type="xs:long" />
```

```
        <xs:element name="downAttrAggrRuleMask" type="xs:long" />
        <xs:element name="downPeakTrafficRate" type="xs:long" />
        <xs:element name="cmtsIPAddress" type="xs:string" maxOccurs="1"
minOccurs="0" />
       </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="createSessionResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="result" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="createSessionExtension">
    <xs:complexType>
      <xs:sequence>
         <xs:element name="String_1" type="xs:string" />
      <xs:element name="String_2" type="xs:string" />
      <xs:element name="String_3" type="xs:string" />
      <xs:element name="int_4" type="xs:int" />
      <xs:element name="int_5" type="xs:int" />
      <xs:element name="int_6" type="xs:int" />
      <xs:element name="int_7" type="xs:int" />
      <xs:element name="int_8" type="xs:int" />
      <xs:element name="long_9" type="xs:long" />
      <xs:element name="String_10" type="xs:string" />
      <xs:element name="String_11" type="xs:string" />
      <xs:element name="String_12" type="xs:string" />
      <xs:element name="String_13" type="xs:string" />
      <xs:element name="int_14" type="xs:int" />
      <xs:element name="int_15" type="xs:int" />
      <xs:element name="String_16" type="xs:string" />
      <xs:element name="String_17" type="xs:string" />
      <xs:element name="int_18" type="xs:int" />
      <xs:element name="int_19" type="xs:int" />
      <xs:element name="String_20" type="xs:string" />
      <xs:element name="int_21" type="xs:int" />
      <xs:element name="int_22" type="xs:int" />
      <xs:element name="boolean_23" type="xs:boolean" />
      <xs:element name="long_24" type="xs:long" />
      <xs:element name="long_25" type="xs:long" />
      <xs:element name="long_26" type="xs:long" />
      <xs:element name="long_27" type="xs:long" />
      <xs:element name="long_28" type="xs:long" />
      <xs:element name="long_29" type="xs:long" />
      <xs:element name="long_30" type="xs:long" />
      <xs:element name="long_31" type="xs:long" />
      <xs:element name="cmtsIPAddress" type="xs:string" maxOccurs="1"
minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="createSessionExtensionResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="result" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="createSessionIPv6">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="String_1" type="xs:string"/>
```

```
      <xs:element name="String_2" type="xs:string"/>
      <xs:element name="String_3" type="xs:string"/>
      <xs:element name="int_4" type="xs:int"/>
      <xs:element name="int_5" type="xs:int"/>
      <xs:element name="int_6" type="xs:int"/>
      <xs:element name="int_7" type="xs:int"/>
      <xs:element name="int_8" type="xs:int"/>
      <xs:element name="long_9" type="xs:long"/>
      <xs:element name="String_10" type="xs:string"/>
      <xs:element name="String_11" type="xs:string"/>
      <xs:element name="String_12" type="xs:string"/>
      <xs:element name="short_13" type="xs:short"/>
      <xs:element name="int_14" type="xs:int"/>
      <xs:element name="int_15" type="xs:int"/>
      <xs:element name="String_16" type="xs:string"/>
      <xs:element name="short_17" type="xs:short"/>
      <xs:element name="int_18" type="xs:int"/>
      <xs:element name="int_19" type="xs:int"/>
      <xs:element name="String_20" type="xs:string"/>
      <xs:element name="int_21" type="xs:int"/>
      <xs:element name="int_22" type="xs:int"/>
      <xs:element name="boolean_23" type="xs:boolean"/>
      <xs:element name="long_24" type="xs:long"/>
      <xs:element name="long_25" type="xs:long"/>
      <xs:element name="long_26" type="xs:long"/>
      <xs:element name="long_27" type="xs:long"/>
      <xs:element name="long_28" type="xs:long"/>
      <xs:element name="long_29" type="xs:long"/>
      <xs:element name="long_30" type="xs:long"/>
      <xs:element name="long_31" type="xs:long"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="createSessionIPv6Response">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="result" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="deleteSession">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="String_1" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="deleteSessionResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="result" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="deleteTrafficClassifier">
  <xs:complexType>
      <xs:sequence>
        <xs:element name="String_1" type="xs:string" />
     <xs:element name="String_2" type="xs:string" />
     <xs:element name="String_3" type="xs:string" />
     <xs:element name="int_4" type="xs:int" />
     <xs:element name="String_5" type="xs:string" />
     <xs:element name="int_6" type="xs:int" />
     <xs:element name="cmtsIPAddress" type="xs:string" maxOccurs="1"
```

```
minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
  </xs:element>
  <xs:element name="deleteTrafficClassifierResponse">
    <xs:complexType>
        <xs:sequence>
          <xs:element name="result" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
  </xs:element>
  <xs:element name="deleteTrafficClassifierWithExtClassifier">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="String_1" type="xs:string" />
     <xs:element name="String_2" type="xs:string" />
     <xs:element name="String_3" type="xs:string" />
     <xs:element name="int_4" type="xs:int" />
     <xs:element name="String_5" type="xs:string" />
     <xs:element name="int_6" type="xs:int" />
     <xs:element name="String_7" type="xs:string" />
     <xs:element name="String_8" type="xs:string" />
     <xs:element name="String_9" type="xs:string" />
     <xs:element name="int_10" type="xs:int" />
     <xs:element name="int_11" type="xs:int" />
    <xs:element name="cmtsIPAddress" type="xs:string" maxOccurs="1" minOccurs="0"
 />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="deleteTrafficClassifierWithExtClassifierResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="result" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="deleteTrafficClassifierWithIPv6Classifier">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="String_1" type="xs:string"/>
        <xs:element name="String_2" type="xs:string"/>
        <xs:element name="String_3" type="xs:string"/>
        <xs:element name="int_4" type="xs:int"/>
        <xs:element name="String_5" type="xs:string"/>
        <xs:element name="int_6" type="xs:int"/>
        <xs:element name="String_7" type="xs:string"/>
        <xs:element name="short_8" type="xs:short"/>
        <xs:element name="short_9" type="xs:short"/>
        <xs:element name="int_10" type="xs:int"/>
        <xs:element name="int_11" type="xs:int"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="deleteTrafficClassifierWithIPv6ClassifierResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="result" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="modifySession">
    <xs:complexType>
      <xs:sequence>
```

```
        <xs:element name="String_1" type="xs:string"/>
        <xs:element name="int_2" type="xs:int"/>
        <xs:element name="int_3" type="xs:int"/>
        <xs:element name="int_4" type="xs:int"/>
        <xs:element name="int_5" type="xs:int"/>
        <xs:element name="String_6" type="xs:string"/>
        <xs:element name="int_7" type="xs:int"/>
        <xs:element name="String_8" type="xs:string"/>
        <xs:element name="int_9" type="xs:int"/>
        <xs:element name="int_10" type="xs:int"/>
        <xs:element name="int_11" type="xs:int"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="modifySessionResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="result" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="modifySessionExtension">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="String_1" type="xs:string"/>
        <xs:element name="int_2" type="xs:int"/>
        <xs:element name="int_3" type="xs:int"/>
        <xs:element name="int_4" type="xs:int"/>
        <xs:element name="int_5" type="xs:int"/>
<xs:element name="String_6" type="xs:string"/>
<xs:element name="String_7" type="xs:string"/>
 <xs:element name="String_8" type="xs:string"/>
<xs:element name="int_9" type="xs:int"/>
<xs:element name="int_10" type="xs:int"/>
<xs:element name="String_11" type="xs:string"/>
        <xs:element name="String_12" type="xs:string"/>
<xs:element name="int_13" type="xs:int"/>
<xs:element name="int_14" type="xs:int"/>
<xs:element name="int_15" type="xs:int"/>
        <xs:element name="int_16" type="xs:int"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="modifySessionExtensionResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="result" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="modifySessionIPv6">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="String_1" type="xs:string"/>
        <xs:element name="int_2" type="xs:int"/>
        <xs:element name="int_3" type="xs:int"/>
        <xs:element name="int_4" type="xs:int"/>
        <xs:element name="int_5" type="xs:int"/>
<xs:element name="String_6" type="xs:string"/>
      <xs:element name="String_7" type="xs:string"/>
       <xs:element name="short_8" type="xs:short"/>
      <xs:element name="int_9" type="xs:int"/>
      <xs:element name="int_10" type="xs:int"/>
      <xs:element name="String_11" type="xs:string"/>
```

```
        <xs:element name="short_12" type="xs:short"/>
      <xs:element name="int_13" type="xs:int"/>
      <xs:element name="int_14" type="xs:int"/>
      <xs:element name="int_15" type="xs:int"/>
        <xs:element name="int_16" type="xs:int"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="modifySessionIPv6Response">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="result" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
    </wsdl:types>
    <wsdl:message name="addTrafficClassifier">
        <wsdl:part element="tns:addTrafficClassifier"
name="addTrafficClassifier"/>
    </wsdl:message>
    <wsdl:message name="addTrafficClassifierResponse">
        <wsdl:part element="tns:addTrafficClassifierResponse"
name="addTrafficClassifierResponse"/>
    </wsdl:message>
    <wsdl:message name="createSessionIPv6">
        <wsdl:part element="tns:createSessionIPv6" name="createSessionIPv6"/>
    </wsdl:message>
    <wsdl:message name="createSessionIPv6Response">
        <wsdl:part element="tns:createSessionIPv6Response"
name="createSessionIPv6Response"/>
    </wsdl:message>
    <wsdl:message name="deleteTrafficClassifier">
        <wsdl:part element="tns:deleteTrafficClassifier"
name="deleteTrafficClassifier"/>
    </wsdl:message>
    <wsdl:message name="deleteTrafficClassifierResponse">
        <wsdl:part element="tns:deleteTrafficClassifierResponse"
name="deleteTrafficClassifierResponse"/>
    </wsdl:message>
    <wsdl:message name="createSessionExtension">
        <wsdl:part element="tns:createSessionExtension"
name="createSessionExtension"/>
    </wsdl:message>
    <wsdl:message name="createSessionExtensionResponse">
        <wsdl:part element="tns:createSessionExtensionResponse"
name="createSessionExtensionResponse"/>
    </wsdl:message>
    <wsdl:message name="addTrafficClassifierWithIPv6Classifier">
        <wsdl:part element="tns:addTrafficClassifierWithIPv6Classifier"
name="addTrafficClassifierWithIPv6Classifier"/>
    </wsdl:message>
    <wsdl:message name="addTrafficClassifierWithIPv6ClassifierResponse">
        <wsdl:part element="tns:addTrafficClassifierWithIPv6ClassifierResponse"
 name="addTrafficClassifierWithIPv6ClassifierResponse"/>
    </wsdl:message>
    <wsdl:message name="deleteTrafficClassifierWithIPv6Classifier">
        <wsdl:part element="tns:deleteTrafficClassifierWithIPv6Classifier"
name="deleteTrafficClassifierWithIPv6Classifier"/>
    </wsdl:message>
    <wsdl:message name="deleteTrafficClassifierWithIPv6ClassifierResponse">
       <wsdl:part element="tns:deleteTrafficClassifierWithIPv6ClassifierResponse"
 name="deleteTrafficClassifierWithIPv6ClassifierResponse"/>
    </wsdl:message>
```

```
    <wsdl:message name="addTrafficClassifierWithExtClassifier">
        <wsdl:part element="tns:addTrafficClassifierWithExtClassifier"
name="addTrafficClassifierWithExtClassifier"/>
    </wsdl:message>
    <wsdl:message name="addTrafficClassifierWithExtClassifierResponse">
        <wsdl:part element="tns:addTrafficClassifierWithExtClassifierResponse"
name="addTrafficClassifierWithExtClassifierResponse"/>
    </wsdl:message>
    <wsdl:message name="deleteTrafficClassifierWithExtClassifier">
        <wsdl:part element="tns:deleteTrafficClassifierWithExtClassifier"
name="deleteTrafficClassifierWithExtClassifier"/>
    </wsdl:message>
    <wsdl:message name="deleteTrafficClassifierWithExtClassifierResponse">
       <wsdl:part element="tns:deleteTrafficClassifierWithExtClassifierResponse"
 name="deleteTrafficClassifierWithExtClassifierResponse"/>
    </wsdl:message>
    <wsdl:message name="createSession">
        <wsdl:part element="tns:createSession" name="createSession"/>
    </wsdl:message>
    <wsdl:message name="createSessionResponse">
        <wsdl:part element="tns:createSessionResponse"
name="createSessionResponse"/>
    </wsdl:message>
    <wsdl:message name="deleteSession">
        <wsdl:part element="tns:deleteSession" name="deleteSession"/>
    </wsdl:message>
    <wsdl:message name="deleteSessionResponse">
        <wsdl:part element="tns:deleteSessionResponse"
name="deleteSessionResponse"/>
    </wsdl:message>
    <wsdl:message name="modifySession">
        <wsdl:part element="tns:modifySession" name="modifySession"/>
    </wsdl:message>
    <wsdl:message name="modifySessionResponse">
        <wsdl:part element="tns:modifySessionResponse"
name="modifySessionResponse"/>
    </wsdl:message>
    <wsdl:message name="modifySessionExtension">
        <wsdl:part element="tns:modifySessionExtension"
name="modifySessionExtension"/>
    </wsdl:message>
    <wsdl:message name="modifySessionExtensionResponse">
        <wsdl:part element="tns:modifySessionExtensionResponse"
name="modifySessionExtensionResponse"/>
    </wsdl:message>
    <wsdl:message name="modifySessionIPv6">
        <wsdl:part element="tns:modifySessionIPv6" name="modifySessionIPv6"/>
    </wsdl:message>
    <wsdl:message name="modifySessionIPv6Response">
        <wsdl:part element="tns:modifySessionIPv6Response"
name="modifySessionIPv6Response"/>
    </wsdl:message>
    <wsdl:portType name="BodSOAPIF">
        <wsdl:operation name="addTrafficClassifier">
            <wsdl:input message="tns:addTrafficClassifier"
name="addTrafficClassifier"/>
            <wsdl:output message="tns:addTrafficClassifierResponse"
name="addTrafficClassifierResponse"/>
        </wsdl:operation>
        <wsdl:operation name="createSessionIPv6">
          <wsdl:input message="tns:createSessionIPv6" name="createSessionIPv6"/>

            <wsdl:output message="tns:createSessionIPv6Response"
name="createSessionIPv6Response"/>
```

```
        </wsdl:operation>
        <wsdl:operation name="deleteTrafficClassifier">
            <wsdl:input message="tns:deleteTrafficClassifier"
name="deleteTrafficClassifier"/>
            <wsdl:output message="tns:deleteTrafficClassifierResponse"
name="deleteTrafficClassifierResponse"/>
        </wsdl:operation>
        <wsdl:operation name="createSessionExtension">
            <wsdl:input message="tns:createSessionExtension"
name="createSessionExtension"/>
            <wsdl:output message="tns:createSessionExtensionResponse"
name="createSessionExtensionResponse"/>
        </wsdl:operation>
        <wsdl:operation name="addTrafficClassifierWithIPv6Classifier">
            <wsdl:input message="tns:addTrafficClassifierWithIPv6Classifier"
name="addTrafficClassifierWithIPv6Classifier"/>
            <wsdl:output
message="tns:addTrafficClassifierWithIPv6ClassifierResponse"
name="addTrafficClassifierWithIPv6ClassifierResponse"/>
        </wsdl:operation>
        <wsdl:operation name="deleteTrafficClassifierWithIPv6Classifier">
            <wsdl:input message="tns:deleteTrafficClassifierWithIPv6Classifier"
 name="deleteTrafficClassifierWithIPv6Classifier"/>
            <wsdl:output
message="tns:deleteTrafficClassifierWithIPv6ClassifierResponse"
name="deleteTrafficClassifierWithIPv6ClassifierResponse"/>
        </wsdl:operation>
        <wsdl:operation name="addTrafficClassifierWithExtClassifier">
            <wsdl:input message="tns:addTrafficClassifierWithExtClassifier"
name="addTrafficClassifierWithExtClassifier"/>
            <wsdl:output
message="tns:addTrafficClassifierWithExtClassifierResponse"
name="addTrafficClassifierWithExtClassifierResponse"/>
        </wsdl:operation>
        <wsdl:operation name="deleteTrafficClassifierWithExtClassifier">
            <wsdl:input message="tns:deleteTrafficClassifierWithExtClassifier"
name="deleteTrafficClassifierWithExtClassifier"/>
            <wsdl:output
message="tns:deleteTrafficClassifierWithExtClassifierResponse"
name="deleteTrafficClassifierWithExtClassifierResponse"/>
        </wsdl:operation>
        <wsdl:operation name="createSession">
            <wsdl:input message="tns:createSession" name="createSession"/>
            <wsdl:output message="tns:createSessionResponse"
name="createSessionResponse"/>
        </wsdl:operation>
        <wsdl:operation name="deleteSession">
            <wsdl:input message="tns:deleteSession" name="deleteSession"/>
            <wsdl:output message="tns:deleteSessionResponse"
name="deleteSessionResponse"/>
        </wsdl:operation>
        <wsdl:operation name="modifySession">
            <wsdl:input message="tns:modifySession" name="modifySession"/>
            <wsdl:output message="tns:modifySessionResponse"
name="modifySessionResponse"/>
        </wsdl:operation>
        <wsdl:operation name="modifySessionExtension">
            <wsdl:input message="tns:modifySessionExtension"
name="modifySessionExtension"/>
            <wsdl:output message="tns:modifySessionExtensionResponse"
name="modifySessionExtensionResponse"/>
        </wsdl:operation>
        <wsdl:operation name="modifySessionIPv6">
          <wsdl:input message="tns:modifySessionIPv6" name="modifySessionIPv6"/>
```

```
            <wsdl:output message="tns:modifySessionIPv6Response"
name="modifySessionIPv6Response"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="BodSessionManagementServiceSoapBinding"
type="tns:BodSOAPIF">
        <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="addTrafficClassifier">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="addTrafficClassifier">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="addTrafficClassifierResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="createSessionIPv6">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="createSessionIPv6">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="createSessionIPv6Response">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="deleteTrafficClassifier">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="deleteTrafficClassifier">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="deleteTrafficClassifierResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="addTrafficClassifierWithIPv6Classifier">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="addTrafficClassifierWithIPv6Classifier">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="addTrafficClassifierWithIPv6ClassifierResponse">

                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="createSessionExtension">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="createSessionExtension">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="createSessionExtensionResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="deleteTrafficClassifierWithIPv6Classifier">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="deleteTrafficClassifierWithIPv6Classifier">
                <soap:body use="literal"/>
            </wsdl:input>
          <wsdl:output name="deleteTrafficClassifierWithIPv6ClassifierResponse">

                <soap:body use="literal"/>
            </wsdl:output>
```

```
        </wsdl:operation>
        <wsdl:operation name="addTrafficClassifierWithExtClassifier">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="addTrafficClassifierWithExtClassifier">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="addTrafficClassifierWithExtClassifierResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="createSession">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="createSession">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="createSessionResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="deleteTrafficClassifierWithExtClassifier">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="deleteTrafficClassifierWithExtClassifier">
                <soap:body use="literal"/>
            </wsdl:input>
          <wsdl:output name="deleteTrafficClassifierWithExtClassifierResponse">

                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="deleteSession">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="deleteSession">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="deleteSessionResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="modifySession">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="modifySession">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="modifySessionResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="modifySessionExtension">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="modifySessionExtension">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="modifySessionExtensionResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="modifySessionIPv6">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="modifySessionIPv6">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="modifySessionIPv6Response">
                <soap:body use="literal"/>
            </wsdl:output>
```

```
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="BodSessionManagementService">
        <wsdl:port binding="tns:BodSessionManagementServiceSoapBinding"
name="BodSOAPIFPort">
            <soap:address location="http://10.60.25.240/bod/sessionmgt"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

The following WSDL script creates a SOAP session with extended classifiers. The parameter long31 indicates that PCMM I06 is supported.

```
<xs:element name="createSessionExtension">
    <xs:complexType>
      <xs:sequence>
        <xs:element  name="String_1" type="xs:string"/>
        <xs:element  name="String_2" type="xs:string"/>
        <xs:element  name="String_3" type="xs:string"/>
        <xs:element name="int_4" type="xs:int"/>
        <xs:element name="int_5" type="xs:int"/>
        <xs:element name="int_6" type="xs:int"/>
        <xs:element name="int_7" type="xs:int"/>
        <xs:element name="int_8" type="xs:int"/>
        <xs:element name="long_9" type="xs:long"/>
        <xs:element  name="String_10" type="xs:string"/>
        <xs:element  name="String_11" type="xs:string"/>
        <xs:element  name="String_12" type="xs:string"/>
        <xs:element  name="String_13" type="xs:string"/>
        <xs:element name="int_14" type="xs:int"/>
        <xs:element name="int_15" type="xs:int"/>
        <xs:element  name="String_16" type="xs:string"/>
        <xs:element  name="String_17" type="xs:string"/>
        <xs:element name="int_18" type="xs:int"/>
        <xs:element name="int_19" type="xs:int"/>
        <xs:element  name="String_20" type="xs:string"/>
        <xs:element name="int_21" type="xs:int"/>
        <xs:element name="int_22" type="xs:int"/>
        <xs:element name="boolean_23" type="xs:boolean"/>
        <xs:element name="long_24" type="xs:long"/>
        <xs:element name="long_25" type="xs:long"/>
        <xs:element name="long_26" type="xs:long"/>
        <xs:element name="long_27" type="xs:long"/>
        <xs:element name="long_28" type="xs:long"/>
        <xs:element name="long_29" type="xs:long"/>
        <xs:element name="long_30" type="xs:long"/>
        <xs:element name="long_31" type="xs:long"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

The following WSDL script modifies the request parameters for a SOAP session with extended classifiers:

```
<xs:element name="modifySessionExtension">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="String_1" type="xs:string"/> sessionId
        <xs:element name="int_2" type="xs:int"/> upBwKbps
        <xs:element name="int_3" type="xs:int"/> upBwMaxKbps
        <xs:element name="int_4" type="xs:int"/> downBwKbps
        <xs:element name="int_5" type="xs:int"/> downBwMaxKbps
```

```
        <xs:element name="String_6" type="xs:string"/> subIP
        <xs:element name="String_7" type="xs:string"/> extSubIP
        <xs:element name="String_8" type="xs:string"/> extSubIpMask
        <xs:element name="int_9" type="xs:int"/> extSubPortStart
        <xs:element name="int_10" type="xs:int"/> extSubPortEnd
        <xs:element name="String_11" type="xs:string"/> extDestIP
        <xs:element name="String_12" type="xs:string"/> extDestIpMask
        <xs:element name="int_13" type="xs:int"/> extDestPortStart
        <xs:element name="int_14" type="xs:int"/> extDestPortEnd
        <xs:element name="int_15" type="xs:int"/> upBwMinKbps
        <xs:element name="int_16" type="xs:int"/> downBwMinKbps
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

The following WSDL script modifies the session SOAP endpoint:

```
http://bod_service_ip/bod/sessionmgt
```

The following WSDL script modifies the WSDL request parameters for a SOAP session:

```
  <xs:element name="modifySession">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="String_1" type="xs:string"/> sessionId
        <xs:element name="int_2" type="xs:int"/> upBwKbps
        <xs:element name="int_3" type="xs:int"/> upBwMaxKbps
        <xs:element name="int_4" type="xs:int"/> downBwKbps
        <xs:element name="int_5" type="xs:int"/> downBwMaxKbps
         <xs:element name="String_6" type="xs:string"/> subIp
        <xs:element name="int_7" type="xs:int"/> subPort
        <xs:element name="String_8" type="xs:string"/> destIp
        <xs:element name="int_9" type="xs:int"/> destPort
        <xs:element name="int_10" type="xs:int"/> upBwMinKbps
        <xs:element name="int_11" type="xs:int"/> downBwMinKbps
      </xs:sequence>
</xs:complexType>
 </xs:element>
```

The response code for the modifySession script:

```
<xs:element name="modifySessionResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element   name="result" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

The response code for the modifySessionExtension script:

```
<xs:element name=" modifySessionExtensionResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element   name="result" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

The following WSDL script creates an IPv6 SOAP session. The parameter long31 indicates that PCMMI06 is supported.

```
<xs:element name="createSessionIPv6">
    <xs:complexType>
      <xs:sequence>
        <xs:element  name="String_1" type="xs:string"/>
        <xs:element  name="String_2" type="xs:string"/>
        <xs:element  name="String_3" type="xs:string"/>
        <xs:element name="int_4" type="xs:int"/>
        <xs:element name="int_5" type="xs:int"/>
        <xs:element name="int_6" type="xs:int"/>
        <xs:element name="int_7" type="xs:int"/>
        <xs:element name="int_8" type="xs:int"/>
        <xs:element name="long_9" type="xs:long"/>
        <xs:element  name="String_10" type="xs:string"/>
        <xs:element  name="String_11" type="xs:string"/>
        <xs:element  name="String_12" type="xs:string"/>
        <xs:element name="short_13" type="xs:short"/>
        <xs:element name="int_14" type="xs:int"/>
        <xs:element name="int_15" type="xs:int"/>
        <xs:element  name="String_16" type="xs:string"/>
        <xs:element name="short_17" type="xs:short"/>
        <xs:element name="int_18" type="xs:int"/>
        <xs:element name="int_19" type="xs:int"/>
        <xs:element  name="String_20" type="xs:string"/>
        <xs:element name="int_21" type="xs:int"/>
        <xs:element name="int_22" type="xs:int"/>
        <xs:element name="boolean_23" type="xs:boolean"/>
        <xs:element name="long_24" type="xs:long"/>
        <xs:element name="long_25" type="xs:long"/>
        <xs:element name="long_26" type="xs:long"/>
        <xs:element name="long_27" type="xs:long"/>
        <xs:element name="long_28" type="xs:long"/>
        <xs:element name="long_29" type="xs:long"/>
        <xs:element name="long_30" type="xs:long"/>
        <xs:element name="long_31" type="xs:long"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

The following WSDL script modifies the request parameters for an IPv6 SOAP session:

```
  <xs:element name="modifySessionIPv6">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="String_1" type="xs:string"/>  sessionId
        <xs:element name="int_2" type="xs:int"/> upBwKbps
        <xs:element name="int_3" type="xs:int"/> upBwMaxKbps
        <xs:element name="int_4" type="xs:int"/> downBwKbps
        <xs:element name="int_5" type="xs:int"/> downBwMaxKbps
        <xs:element name="String_6" type="xs:string"/> subIP
        <xs:element name="String_7" type="xs:string"/> extSubIP
        <xs:element name="short_8" type="xs:short"/> subIpPrefixLength
        <xs:element name="int_9" type="xs:int"/> extSubPortStart
        <xs:element name="int_10" type="xs:int"/> extSubPortEnd
        <xs:element name="String_11" type="xs:string"/> extDestIP
        <xs:element name="short_12" type="xs:short"/>destIpPrefixLength
        <xs:element name="int_13" type="xs:int"/> extDestPortStart
        <xs:element name="int_14" type="xs:int"/> extDestPortEnd
        <xs:element name="int_15" type="xs:int"/> upBwMinKbps
        <xs:element name="int_16" type="xs:int"/> downBwMinKbps
      </xs:sequence>
```

```
    </xs:complexType>
  </xs:element>
```

The response for the modifySessionIPv6 script:

```
<xs:element name=" modifySessionIPv6Response">
    <xs:complexType>
      <xs:sequence>
        <xs:element   name="result" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

The following WSDL script modifies the request parameters for a multiple classifier SOAP session:

```
<xs:element name="createSessionWithMultiClassifiers">
<xs:complexType>
  <xs:sequence>
   <xs:element name="String_1" type="xs:string" />
<xs:element name="String_2" type="xs:string" />
<xs:element name="String_3" type="xs:string" />
<xs:element name="int_4" type="xs:int" />
<xs:element name="int_5" type="xs:int" />
<xs:element name="int_6" type="xs:int" />
<xs:element name="int_7" type="xs:int" />
<xs:element name="int_8" type="xs:int" />
<xs:element name="String_9" type="xs:string" />
<xs:element name="String_10" type="xs:string" />
<xs:element name="String_11" type="xs:string" />
<xs:element name="String_12" type="xs:string" />
<xs:element name="String_13" type="xs:string" />
<xs:element name="String_14" type="xs:string" />
<xs:element name="String_15" type="xs:string" />
<xs:element name="String_16" type="xs:string" />
<xs:element name="String_17" type="xs:string" />
<xs:element name="String_18" type="xs:string" />
<xs:element name="String_19" type="xs:string" />
<xs:element name="String_20" type="xs:string" />
<xs:element name="String_21" type="xs:string" />
<xs:element name="String_22" type="xs:string" />
<xs:element name="String_23" type="xs:string" />
<xs:element name="String_24" type="xs:string" />
<xs:element name="String_25" type="xs:string" />
<xs:element name="String_26" type="xs:string" />
<xs:element name="String_27" type="xs:string" />
<xs:element name="String_28" type="xs:string" />
<xs:element name="String_29" type="xs:string" />
<xs:element name="String_30" type="xs:string" />
<xs:element name="int_31" type="xs:int" />
<xs:element name="int_32" type="xs:int" />
<xs:element name="boolean_33" type="xs:boolean" />
<xs:element name="long_34" type="xs:long" />
<xs:element name="long_35" type="xs:long" />
<xs:element name="long_36" type="xs:long" />
<xs:element name="long_37" type="xs:long" />
<xs:element name="long_38" type="xs:long" />
<xs:element name="long_39" type="xs:long" />
<xs:element name="long_40" type="xs:long" />
<xs:element name="long_41" type="xs:long" />
<xs:element name="long_42" type="xs:long" />
<xs:element name="cmtsIPAddress" type="xs:string" maxOccurs="1" minOccurs="0"
/>
```

```
  </xs:sequence>
</xs:complexType>
</xs:element>
```

Appendix B

BoD Interface Error Codes

Topics:

This appendix lists the BoD interface error codes displayed by the Bandwidth on Demand (BoD) application.

BoD Interface Error Codes

The following is a list of BoD Interface error codes returned from the SOAP interface.

```
PCMM_CANT_INITIALIZE                        = -100
PCMM_CANT_FIND_SERVICE                      = -101
PCMM_SERVICE_NAME_NOT_PROVIDED              = -102
PCMM_INTERNAL_APP_ERROR                     = -103
PCMM_CANT_CONNECT_TO_PS                     = -104
PCMM_REQUEST_TIMEOUT                        = -105
PCMM_REQUEST_COMM_ERROR                     = -106

DB_CANT_START_TRANSACTION                   = -200
DB_INTERNAL_APP_ERROR                       = -201

INVALID_INPUT_ARG                           = -300
INVALID_UPBW_ARG                            = -301
INVALID_DOWNBW_ARG                          = -302
INVALID_UPBWMIN_ARG                         = -303
INVALID_UPBWMAX_ARG                         = -304
INVALID_DOWNBWMIN_ARG                       = -305
INVALID_DOWNBWMAX_ARG                       = -306
INVALID_DUR_ARG                             = -307
INVALID_SUBPORT_ARG                         = -308
INVALID_DESTPORT_ARG                        = -309
INVALID_SUBIP_ARG                           = -310
INVALID_DESTIP_ARG                          = -311
INVALID_STARTTIME_ARG                       = -312
INVALID_SESSIONIDFORMAT_ARG                 = -313
INVALID_SESSIONID_ARG                       = -314
INVALID_RCIP_ARG                            = -315
INVALID_DESTIPFMT_ARG                       = -316
INVALID_URL_ARG                             = -317
INVALID_GATEINFO_ARG                        = -318
MISSING_UPBW_ARG                            = -319
MISSING_DOWNBW_ARG                          = -320
MISSING_UPBWMIN_ARG                         = -321
MISSING_UPBWMAX_ARG                         = -322
MISSING_DOWNBWMIN_ARG                       = -323
MISSING_DOWNBWMAX_ARG                       = -324
MISSING_DUR_ARG                             = -325
MISSING_SUBPORT_ARG                         = -326
MISSING_SUBIP_ARG                           = -327
MISSING_DESTIP_ARG                          = -328
MISSING_DESTPORT_ARG                        = -329
MISSING_MACADDR_ARG                         = -330
MISSING_SUBIP_OR_SUBMAC_ARG                 = -331
INVALID_MACADDRFORMAT_ARG                   = -332
INVALID_GROUPNAME_ARG                       = -333
INVALID_SSIDNAME_ARG                        = -334

CANT_EDIT_ACTIVE_SESSION                    = -400
CANT_FIND_SESSION_BY_CALLER_ID              = -401
SESSION_NO_LONGER_ACTIVE = -402
SESSION_ALREADY_EXISTS_BY_CALLER_ID         = -403
CANT_DEL_CLASSIFIER_INACTIVE_SESSION        = -404
CANT_DEL_CLASSIFIER_NOT_FOUND               = -405
MACTRANS_CONNECT_FAILURE                    = -406
MACTRANS_RESPONSE_PARSE_FAILURE             = -407
MACTRANS_MACNEEDTRANS_NO_TRANSLATOR         = -408
```

```
MACTRANS_BADIP_RETURNED                    = -409
CANT_CREATE_SUBS_NONUNIQUE_NAME            = -500
BL_USER_BADUNPW                            = -600
BL_USER_BADPW                              = -601
BL_USER_EXISTS                             = -602
BL_CANNOT_DELETE_USER                      = -603
APP_MAX_SESSIONS                           = -800

SERVICE_NO_FILE_RECORD_EXISTS              = -900
```

Glossary

A

AM	Application Manager A server within a network that is responsible for establishing and managing subscriber sessions associated with a specific application.
API	Application Programming Interface An interface with commands, possibly routines and/or macros, provided by an operating system or an add-on for an operating system (that support network use, for example). Application programs can use this interface to tell the operating system to perform specific actions.

B

Bandwidth on Demand	See BoD.
BoD	Bandwidth on Demand An application that provides dynamic allocation of bandwidth; for example, a broadband speed promotion.

C

CMP	Configuration Management Platform A centralized management interface to create policies, maintain policy libraries, configure, provision, and manage multiple distributed MPE policy server

C

	devices, and deploy policy rules to MPE devices. The CMP has a web-based interface.
CMTS	Cable Modem Termination System An edge device connecting to subscribers' cable modems in a broadband network. A CMTS device can function as a PCEF device; see PCEF. Equipment used by cable companies to provide high speed data services to cable subscribers.
COMCOL	Communications Core Object Library A suite of re-usable C++ libraries, as well as processes and procedures available for use in Tekelec products. Many of its features are focused toward the communications area of software developments, although it purpose is not intended to restrict its functionality to any particular area

D

DOCSIS	Data Over Cable Service Interface Specification - An international telecommunications standard for adding high-speed data transfer to an existing cable TV system. Employed by many cable television operators to provide Internet access over their existing infrastructure.
DSCP	Differentiated Service Code Point Differentiated Services Code Point Provides a framework and building blocks to enable deployment of scalable service discrimination in

D

	the internet. The differentiated services are realized by mapping the code point contained in a field in the IP packet header to a particular forwarding treatment or per-hop behavior (PHB). Differentiated services or DiffServ is a computer networking architecture that specifies a simple, scalable and coarse-grained mechanism for classifying and managing network traffic and providing quality of service (QoS) on modern IP networks.

F

FQDN	Fully Qualified Domain Name The complete domain name for a specific computer on the Internet (i.e., www.oracle.com). A domain name that specifies its exact location in the tree hierarchy of the DNS.

H

HTTP	Hypertext Transfer Protocol

M

MAC	Media Access Control Address The unique serial number burned into the Ethernet adapter that identifies that network card from all others.
MPE	Multimedia Policy Engine A high-performance, high-availability platform for operators to deliver and manage differentiated services over high-speed data networks. The MPE includes a protocol-independent policy rules

M

	engine that provides authorization for services based on policy conditions such as subscriber information, application information, time of day, and edge resource utilization.
MSO	Multiple-service operator
Multimedia Policy Engine	See MPE.

O

OAM	Operations, Administration, and Maintenance The application that operates the Maintenance and Administration Subsystem that controls the operation of many products.

P

PCMM	PacketCable MultiMedia

Q

QoS	Quality of Service Control mechanisms that guarantee a certain level of performance to a data flow.

S

SOAP	Simple Object Access Protocol

T

TCP	Transfer-Cluster-Prohibited Transfer Control Protocol Transmission Control Protocol A connection-oriented protocol used by applications on networked hosts to connect to one another and

T

	to exchange streams of data in a reliable and in-order manner.

U

UDP	User Datagram Protocol
UI	User Interface

W

WSDL	Web Service Definition Language