broadsoft

BroadSoft is
now part of Cisco.

broadsoft
broadworks

# Diameter, Rf, Ro, and Sh Interfaces

## Configuration Guide

Release 23.0

Document Version 1

BroadSoft | 9737 Washingtonian Blvd | Suite 350 | Gaithersburg, MD 20878 | broadsoft.com

# BroadSoft® Guide

## Copyright Notice

## Trademarks

This document is printed in the United States of America.

## Document Revision History

| Release | Version | Reason for Change | Date | Author |
|---|---|---|---|---|
| 14.sp4 | 1 | Created document. | December 5, 2006 | Steve Liao |
| 15.0 | 1 | Renamed document from *BW-ShInterfaceConfigGuide-R14.sp4* to *BW-DiameterRfShConfigGuide-R150*.<br>Added Rf information. | June 20, 2008 | Martin Piotte |
| 15.0 | 1 | Edited and published document. | August 18, 2008 | Andrea Fitzwilliam |
| 16.0 | 1 | Added Ro information, renamed document, and updated document for Release 16.0. | April 29, 2009 | Nicolas Turgeon |
| 16.0 | 1 | Edited changes and published document. | July 9, 2009 | Andrea Fitzwilliam |
| 17.0 | 1 | Updated document for Release 17.0, moving from Condor Diameter stack to BroadSoft's own stack. | March 5, 2010 | Nicolas Turgeon |
| 17.0 | 1 | Edited changes and published document. | April 8, 2010 | Andrea Fitzwilliam |
| 17.0 | 2 | Added information to section *3.1.2.3 Configure Centralized Data using Command Line Interface* for EV 117808. | September 9, 2010 | Jocelyn Lepage |
| 17.0 | 2 | Reviewed changes. | September 9, 2010 | Patricia Renaud |
| 17.0 | 2 | Updated company address, copyright information, and references to Xchange. | February 3, 2011 | Goska Auerbach |
| 17.0 | 2 | Edited changes and published document. | April 12, 2011 | Jessica Boyle |
| 17.0 | 3 | Added a conversion factor for AVPs written to disk in XML format in section *4.4 Select Optional Content, Triggers, Routing Mode, and Polling Interval* for EV 125969. | May 18, 2011 | Eric Bernier |
| 17.0 | 3 | Edited changes and published document. | July 13, 2011 | Jessica Boyle |
| 18.0 | 1 | Updated document for Release 18.0. | November 4, 2011 | Martin Piotte |
| 18.0 | 1 | Edited changes and published document. | November 10, 2011 | Patricia Renaud |
| 19.0 | 1 | Updated document for Release 19.0. | September 12, 2012 | Martin Piotte |
| 19.0 | 1 | Edited changes and published document. | October 17, 2012 | Patricia Renaud |
| 19.0 | 2 | Removed obsolete exclusion for Subscriber Location Function (SLF) in section *6.4 Configure HSS Realm* for EV 176487. | December 12, 2012 | Martin Piotte |
| 19.0 | 2 | Edited changes and published document. | March 4, 2013 | Joan Renaud |
| 20.0 | 1 | Updated document for Release 20.0. | September 13, 2013 | Martin Piotte |
| 20.0 | 1 | Edited changes and published document. | October 8, 2013 | Joan Renaud |
| 21.0 | 1 | Updated document for Release 21.0. | August 26, 2014 | Martin Piotte |
| 21.0 | 1 | Edited changes. | September 22, 2014 | Joan Renaud |
| 21.0 | 1 | Rebranded and published document. | December 18, 2014 | Patricia Renaud |

| Release | Version | Reason for Change | Date | Author |
|---|---|---|---|---|
| 21.0 | 2 | Updated information related to the listening address. | October 7, 2015 | Martin Piotte |
| 21.0 | 2 | Edited changes and published document. | June 10, 2016 | Joan Renaud |
| 22.0 | 1 | Updated document for Release 22.0. | November 18, 2016 | Isaël St-André |
| 22.0 | 1 | Edited changes and published document. | December 16, 2016 | Patricia Renaud |
| 23.0 | 1 | Updated document for Release 23.0. | September 14, 2018 | Isaël St-André |
| 23.0 | 1 | Rebranded document for Cisco.<br>Edited changes and published document. | September 27, 2018 | Joan Renaud |

## Table of Contents

## Table of Figures

# 1 Summary of Changes

This section describes the changes to this document for each release and document version.

## 1.1 Changes for Release 23.0, Document Version 1

This version of the document includes the following change:

- Added configuration for the Government Emergency Telecommunications Service (GETS) in section *3.7 Government Emergency Telecommunications Service*.

## 1.2 Changes for Release 22.0, Document Version 1

This version of the document was upgraded from Release 21.0 to Release 22.0.

## 1.3 Changes for Release 21.0, Document Version 2

This version of the document includes the following change:

- Updated information related to the listening address.

## 1.4 Changes for Release 21.0, Document Version 1

This version of the document includes the following changes:

- Added configuration for *calledPartyAddressStrictCompliance* and *InhibitedAVPCodeList* in sections *4.4 Select Optional Content, Triggers, Routing Mode, and Polling Interval* and *5.5 Select Optional Content, Triggers, and Routing Mode*.
- Added configuration for *backlogMaxParallelTransactions* in section *4.4 Select Optional Content, Triggers, Routing Mode, and Polling Interval*.
- Added configuration for *serviceContextID* and *serviceContextIDAoC* in section *5.5 Select Optional Content, Triggers, and Routing Mode*.

## 1.5 Changes for Release 20.0, Document Version 1

This version of the document includes the following changes:

- Added configuration for the *callingPartyAddressStrictCompliance* and *InhibitedAVPCodeList* in sections *4.4 Select Optional Content, Triggers, Routing Mode, and Polling Interval* and *5.5 Select Optional Content, Triggers, and Routing Mode*.

## 1.6 Changes for Release 19.0, Document Version 2

This version of the document includes the following change:

- Removed obsolete exclusion for Subscriber Location Function (SLF) in section *6.4 Configure HSS Realm* for EV 176487.

## 1.7 Changes for Release 19.0, Document Version 1

This version of the document includes the following change:

- Added configuration of the pollingIntervalSeconds for the offline billing requests saved on disk to section *4.4 Select Optional Content, Triggers, Routing Mode, and Polling Interval*.

## 1.8 Changes for Release 18.0, Document Version 1

This version of the document includes the following changes:

- Added configuration of the *advertisedOfflineBillingApplication* and *advertisedOnlineBillingApplication* parameters to section *3.1.1 Diameter Interface Base System Data.*
- Added usage of IPv6 to section *3.6 Usage of IPv6*.

## 1.9 Changes for Release 17.0, Document Version 3

This version of the document includes the following change:

- Added a conversion factor for AVPs written to disk in XML format in section *4.4 Select Optional Content, Triggers, Routing Mode, and Polling Interval* for EV 125969.

## 1.10 Changes for Release 17.0, Document Version 2

This version of the document includes the following change:

- Added information for the command line interface (CLI) alternative that is used to configure centralized data (that is, server-specific data) to section *3.1.2.3 Configure Centralized Data using Command Line Interface* for EV 117808.

## 1.11 Changes for Release 17.0, Document Version 1

This version of the document had a major revision due to the move from the Condor Diameter stack to BroadSoft's own stack.

## 1.12 Changes for Release 16.0, Document Version 1

This version of the document includes the following changes:

- The document has been renamed from *BW-Diameter-Rf-ShConfigGuide-R150* to *BW-Diameter-Rf-Ro-ShConfigGuide-R160*. The title has changed from *BroadWorks Diameter, Rf, and Sh Interface Configuration Guide* to *BroadWorks Diameter, Rf, Ro, and Sh Interface Configuration Guide*.
- The Ro interface information has been added to the document.
- The ApnDebug CLI level has been moved up one level (now under Diameter instead of Diameter/Offline).
- The *rfApn.conf* file is now shared with the Ro interface, becoming *rorfApn.conf*.

## 1.13 Changes for Release 15.0, Document Version 1

This version of the document includes the following changes:

- The document has been renamed from *BW-ShInterfaceConfigGuide-R14.sp4* to *BW-DiameterRfShConfigGuide-R150*. The title has changed from *BroadWorks Sh Interface Configuration Guide* to *BroadWorks Diameter, Rf, and Sh Interface Configuration Guide*.
- The Diameter and the Sh configuration have been divided into separate sections.
- The Rf interface information has been added to the document.

## 1.14 Changes for Release 14.sp4, Document Version 1

This document was created for Release 14.sp4.

# 2 Overview

This document describes the steps necessary to configure the Rf, Ro, and Sh interfaces on the BroadWorks Application Server.

This document assumes that the reader is familiar with the basic Diameter concepts such as realms, peers, and peer routing. For an overview of Diameter, Rf, Ro, and Sh interface architecture, see the *Diameter Base Protocol, RFC 3588* [3], *Diameter Credit-Control Application, RFC 4006* [4], *BroadWorks Rf/Ro Interface Specification* [6], and *BroadWorks Sh Interface Specification* [7].

This document is organized as follows:

- The first section describes the steps required to enable and configure the Diameter stack, which is a required component for the Rf, Ro, and Sh interfaces.
- The second section describes the configuration steps specific to the Rf interface.
- The third section describes the configuration steps specific to the Ro interface.
- The fourth section describes the steps required to enable and configure the Sh interface. It also describes related requirements on the Home Subscriber server (HSS).

# 3 Diameter Stack

This section describes the steps required to configure the Diameter stack. The Diameter stack runs inside the Provisioning Server and Execution Server processes on the BroadWorks Application Server. The stack is shared between the Rf, Ro, and Sh interfaces. It must be configured first.

Configuring the Diameter stack consists of the following steps:

- Setting the Diameter base system data (Diameter identities, realms, listening addresses, ports, and so on).
- Adding Diameter peers.
- Adding routing realms (optional, only if realm routing is used).

All the CLI configuration items in this document are also available through the Open Client Interface-Provisioning (OCI-P) interface. A few items are only configurable centrally on the Element Management System (EMS) through the configuration agent (see section *3.1.2 Centralized Configuration Data*).

## 3.1 Diameter Stack System Data

### 3.1.1 Diameter Interface Base System Data

This section describes the low-level Diameter stack configuration. Most configuration items are in the *AS_CLI/Interface/Diameter* CLI level.

```
AS_CLI/Interface/Diameter> get
  xsRealm = operator.com
  xsListeningPort = 3868
  psRealm =
  psListeningPort = 13868
  psRelayThroughXs = true
  xsRelayListeningPort = 13869
  tcTimerSeconds = 30
  twTimerSeconds = 30
  requestTimerSeconds = 300
  busyPeerDetectionOutstandingTxnCount = 10000
  busyPeerRestoreOutstandingTxnCount = 8000
  dynamicEntryInactivityTimerHours = 24
  advertisedOfflineBillingApplication = base
  advertisedOnlineBillingApplication = base
```

When Diameter is used on the Execution Server (for example, for the Rf or Ro interfaces), the *xsRealm* value must be set, as shown in the following example, to specify the Diameter realm associated with the Execution Server external Diameter interface.

```
AS_CLI/Interface/Diameter> set xsRealm operator.com
...Done
```

Changing this value while the Execution Server external Diameter interface is enabled causes a restart of the Diameter interface.

The Transmission Control Protocol (TCP) port on which the Execution Server external Diameter interface is listening for incoming Diameter connections can be set with the following command.

```
AS_CLI/Interface/Diameter> set xsListeningPort 3868
...Done
```

Changing this value while the Execution Server external Diameter interface is enabled causes a restart of the interface.

The Diameter realm associated with the Provisioning Server is only used when using Diameter on the Provisioning Server (for example, for the Sh interface) if the Execution Server relay is disabled (that is, if *psRelayThroughXS* is "false"). For more information on the Execution Server relay, see section *6.9 Supported Application Server Deployment Topologies (Relay)*. The following example shows how this value is configured.

```
AS_CLI/Interface/Diameter> set psRealm operator.com
...Done
```

Changing this value while the Provisioning Server Diameter interface is enabled and the Execution Server relay is not used causes a restart of the Diameter interface.

The TCP port on which the Provisioning Server Diameter interface is listening for incoming Diameter connections can be set with the following command. The same port value is used whether or not the Provisioning Server interface is using Execution Server relay.

```
AS_CLI/Interface/Diameter> set psListeningPort 3868
...Done
```

Changing this value while the Provisioning Server Diameter interface is enabled and the Execution Server relay is not used causes a restart of the interface.

To enable or disable the Relayed Diameter interface, configure the *psRelayThroughXs* parameter as follows (in this example, it is enabled). For information on the Execution Server relay, see section *6.9 Supported Application Server Deployment Topologies (Relay)*.

```
AS_CLI/Interface/Diameter> set psRelayThroughXs true
...Done
```

When relaying is enabled, configure the TCP port on which the Execution Server internal Diameter interface is listening for incoming connections from the Provisioning Server. The Execution Server internal Diameter interface is always listening on the local host interface.

```
AS_CLI/Interface/Diameter> set xsRelayListeningPort 13869
...Done
```

The TC timer controls the delay after which the Diameter stack attempts to reconnect to a peer after detecting a connection failure. To configure the TC timer, use the following parameter.

```
AS_CLI/Interface/Diameter> set tcTimerSeconds 45
...Done
```

The TW timer controls the frequency of Device-Watchdog messages (when no other messages are exchanged on the peer connection) and is used in the connection failure detection algorithm. To configure the TW timer, use the following parameter.

```
AS_CLI/Interface/Diameter> set twTimerSeconds 60
...Done
```

The request timer is the timer after which the Diameter stack reports a timeout to applications and removes the request from its requests queue. It is set as follows.

```
AS_CLI/Interface/Diameter> set requestTimerSeconds 300
...Done
```

The following example shows how to configure the maximum number of concurrent outstanding transactions allowed on a single peer connection, after which the Diameter stack reports busy peer to applications. The Diameter stack no longer sends requests to the peer until the number of concurrent outstanding transactions has been lowered below this parameter.

```
AS_CLI/Interface/Diameter> set busyPeerDetectionOutstandingTxnCount 5000
...Done
```

The following example shows how to configure the number of concurrent outstanding transactions at which the Diameter stack starts sending requests again after having detected a busy peer condition.

```
AS_CLI/Interface/Diameter> set busyPeerRestoreOutstandingTxnCount 4000
...Done
```

The following example shows how to configure the idle time allowed for a dynamic entry (either a Realm Routing Table entry or a Peer Table entry) before it is removed. Dynamic Peer Table entries are not removed if referenced by at least one Realm Routing Table entry.

```
AS_CLI/Interface/Diameter> set dynamicEntryInactivityTimerHours 12
...Done
```

The Diameter interface advertises the applications it supports during the capability exchange. For the Ro and Rf interface, this can be advertised as the base applications defined in *RFC 3588* and *RFC 4006* or as 3GPP vendor-specific application according to the following configuration.

```
AS_CLI/Interface/Diameter> h set
This command is used to modify Diameter-related attributes in the system.
...
advertisedOfflineBillingApplication:
                 The application Id advertised for offline billing during the
                 Diameter Capability Exchange. When base is selected,
                 Acct-Application-Id=3 is advertised. When 3gpp is selected,
                 Vendor-Specific-Application-Id(Acct-Application-Id=3;
                 Vendor-Id=10415) is advertised.
advertisedOnlineBillingApplication:
                 The application Id advertised for online billing during the
                 Diameter Capability Exchange. When base is selected,
                 Auth-Application-Id=4 is advertised. When 3gpp is selected,
                 Vendor-Specific-Application-Id(Auth-Application-Id=4;
                 Vendor-Id=10415) is advertised.

AS_CLI/Interface/Diameter> set advertisedOfflineBillingApplication base
...Done

AS_CLI/Interface/Diameter> set advertisedOnlineBillingApplication base
...Done
```

The last base system data parameter is configured via another CLI level, as shown in the following example. The default Destination-Realm is used for failover accounting request (ACR) messages on the Rf interface to populate the Destination-Realm Attribute-Value Pair (AVP). Normally, this AVP is set using the domain part of the User-Name AVP, but in the case of failover call detail records (CDRs), the User-Name AVP is not specified, and so the mandatory Destination-Realm AVP is populated with this value instead.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter> set
       defaultDestinationRealm service.provider.com
...Done
```

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter> get
  defaultDestinationRealm = service.provider.com
```

### 3.1.2 Centralized Configuration Data

The Diameter stack configuration not being supported by the Application Server database is made available on the Element Management System (EMS) through the configuration agent. This consists of server-specific data.

When an Element Management System is not used, server-specific data can be configured on the command line interface (CLI).

#### 3.1.2.1 executionServerSubsystem

The following configuration items are available in the Diameter settings category.

| Name | Type | Content Restrictions | Default Value | Description |
|---|---|---|---|---|
| xsIdentity | String | 1 to 80 ASCII characters | | Server-specific Diameter identity associated with the Execution Server, value of Origin-Host AVP in outgoing Diameter messages. |
| listeningAddress | String | 1 to 39 characters numerical IP address (IPv4 or IPv6) | | Server-specific address on which the Execution Server Diameter stack binds a server socket for incoming connections. |
| maxPoolSize | Integer | Positive | | Number of Execution Server Diameter worker threads. |

**NOTE**: The *listeningAddress* must be set to a numerical IP address (IPv4 or IPv6). It cannot be set to a wildcard address (for example, 0.0.0.0) or to a domain name (for example, domain.invalid).

#### 3.1.2.2 provisioningServerSubsystem

The following configuration items are available in the Diameter settings category.

| Name | Type | Content Restrictions | Default Value | Description |
|---|---|---|---|---|
| xsIdentity | String | 1 to 80 ASCII characters<br>Same value as xsIdentity under *executionServerSubsystem/diameter*. | | Server-specific Diameter - identity associated with the Execution Server. Same value as xsIdentity in the *executionServerSubsystem/diameter*. Required when operating in relay mode. |
| psIdentity | String | 1 to 80 ASCII characters | | Server-specific Diameter identity associated with the Provisioning Server, value of Origin-Host AVP in outgoing Diameter messages. |

| Name | Type | Content Restrictions | Default Value | Description |
|---|---|---|---|---|
| listeningAddress | String | 1 to 39 characters numerical IP address (IPv4 or IPv6) | | Server-specific address on which the Provisioning Server Diameter stack binds a server socket for incoming connections. |
| maxPoolSize | Integer | Positive | | Number of Provisioning Server Diameter worker threads. |

**NOTE**: The *listeningAddress* must be set to a numerical IP address (IPv4 or IPv6). It cannot be set to a wildcard address (for example, 0.0.0.0) or to a domain name (for example, domain.invalid).

#### 3.1.2.3 Configure Centralized Data using Command Line Interface

If an EMS is not deployed, the CLI must be used to configure the server-specific data. This is done using the *AS_CLI/System/StartupParam* CLI level. The mapping to centralized data is as follows.

| Startup Param Name | Centralized Data Mapping |
|---|---|
| bw.diameter.xs.identity | executionServerSubsystem/diameter/xsIdentity AND provisioningServerSubsystem/diameter/xsIdentity |
| bw.diameter.xs.listeningAddress | executionServerSubsystem/diameter/listeningAddress |
| bw.diameter.xs.maxPoolSize | executionServerSubsystem/diameter/maxPoolSize |
| bw.diameter.ps.identity | provisioningServerSubsystem/diameter/psIdentity |
| bw.diameter.ps.listeningAddress | provisioningServerSubsystem/diameter/listeningAddress |
| bw.diameter.ps.maxPoolSize | provisioningServerSubsystem/diameter/maxPoolSize |

Note that the server-specific data MUST be set on each Application Server node in the cluster (that is, the operator must log in to the CLI on each Application Server node and set the values). Each Application Server node MUST have distinct Diameter identities and listening addresses. Diameter identities assigned to provisioning subsystems MUST be distinct from identities assigned to the execution subsystems, unless the execution subsystem relay is used. When the relay is enabled, the Diameter identities assigned to the provisioning subsystems are not used.

**NOTE**: The *bw.diameter.xs.listeningAddress* and *bw.diameter.ps.listeningAddress* must be set to numerical IP addresses (IPv4 or IPv6). They cannot be set to a wildcard address (for example, 0.0.0.0) or to a domain name (for example, domain.invalid).

## 3.2 Diameter Peers

The Diameter stack uses a Peer Table and a Realm Routing Table to determine where to send outgoing Diameter requests. Outgoing answers are always sent back on the connection from which the incoming request has been received; therefore, it does not involve the Peer Table and Realm Routing Table.

The routing mode (direct, dynamic, or realm) is configurable separately for Rf and Ro (see their respective sections later in this document). For the Sh interface, realm routing is always used. For specific request forwarding and routing procedures, see the *BroadWorks Rf/Ro Interface Specification* [6] and the *BroadWorks Sh Interface Specification* [7].

The administrator manages the Peer Table and the Realm Routing Table using the CLI or via OCI-P. There are distinct Peer Tables and Realm Routing Tables for the Provisioning Server (PS) and the Execution Server (XS). When operation is in Relayed Diameter interfaces, only the Execution Server tables are used for routing outgoing requests. (The Provisioning Server forwards all outgoing requests to the Execution Server internal Diameter interface.)

The Peer Table determines which peers the Diameter stack is allowed to directly communicate with. The Diameter stack attempts to maintain an opened Diameter connection for every enabled peer defined in the Peer Table.

Static Diameter peers (that is, the remote Diameter servers BroadWorks communicates with) are configured via the *AS_CLI/Interface/Diameter/Peers* level.

```
AS_CLI/Interface/Diameter/Peers> help
    0)         get : show Peers-related attributes
    1)         add : add a new Peers
    2)         set : modify Peers-related attributes
    3)      delete : delete an existing Peers
    4)       clear : clear Peers-related attributes

  h (help),  e (exit),  q (quit),  r (read),  w (write),  t (tree),
  c (config),  cd (cd),  a (alias),  hi (history),  p (pause),  re (repeat)

AS_CLI/Interface/Diameter/Peers> get
  Instance           Identity  IP Address  Port  Enabled
  ======================================================
        xs   cdf.provider.com              3868    false
        xs  hss1.provider.com              3868     true
        ps  hss2.provider.com              3868     true
        xs   ocs.provider.com              3868     true

  4 entries found.
```

Note that the get command shows static peers only and does not show any dynamic peers. To see all peers (static and dynamic), go to the *AS_CLI/ASDiagnostic/Diameter/ Peers* level.

Entries are added as follows. The help set command also includes each parameter's description.

```
S_CLI/Interface/Diameter/Peers> help add
This command is used to add Diameter Peers-related entries in the system.

Parameters description:
instance : The process for which the Peer Table is displayed.
identity : The Diameter Identity of the peer.
port     : The TCP port on which the peer is listening for Diameter
           connections.
enabled  : Indicates if connection to the peer shall be established. If set to
           false, the Diameter Stack won't attempt to connect to the peer and
           will reject incoming connection attempts from the peer.
attribute: Addionnal attributes to include through the add command.
ipAddress: The IP address or the peer. This parameter is optional; if not
           specified the actual IP address is resolved on the DNS.
```

```
======================================================================
add
    <instance>, Choice = {ps, xs}
    <identity>, Domain | Host (1 to 80 chars)
    <port>, Integer {1025 to 65535}
    <enabled>, Choice = {false, true}
    [<attribute>, Multiple Choice = {ipAddress}]
        <ipAddress>, IP address (1 to 39 chars)

AS_CLI/Interface/Diameter/Peers> add ps hss2.provider.com 3868 true
...Done
```

In addition, entries can be modified (set) and deleted, and parameters cleared in the standard CLI fashion.

## 3.3 Diameter Realms

### 3.3.1 Static Realms

Static Diameter realms are configured via the *AS_CLI/Interface/Diameter/Realms* level.

```
AS_CLI/Interface/Diameter/Realms> help
    0)           get : show Realms-related attributes
    1)           add : add a new Realms
    2)           set : modify Realms-related attributes
    3)        delete : delete an existing Realms
    4)  RoutingPeers : go to level RoutingPeers

  h (help),  e (exit),  q (quit),  r (read),  w (write),  t (tree),
  c (config),  cd (cd),  a (alias),  hi (history),  p (pause),  re (repeat)

AS_CLI/Interface/Diameter/Realms> get
  Instance         Realm  ApplicationID  Default
  ==============================================
        xs   billing.net            Rf    false
        xs   billing.net            Ro    false
        ps  operator.com            Sh    false

  3 entries found.
```

Note that the get command above shows static realms only and does not show any dynamic information. Go to the *AS_CLI/ASDiagnostic/Diameter/Realms* level to see all realms (both static and dynamic).

Entries are added as follows. The help add command also includes each parameter's description.

```
AS_CLI/Interface/Diameter/Realms> help add
This command is used to add Realm Routing Table's static entries.

Parameters description:
instance     : The process for which the Realm Routing Table is displayed.
realm        : The Realm Name
applicationId: The application associated with the Realm Routing Table entry.
               All is a wildcard indicating the entry applies for all supported
               Diameter applications.
default      : Indicates if the entry is also used as a default Realm Routing
               Table entry for the specified applicationId. There can be at
               most one default entry per applicationId.

======================================================================
add
```

```
    <instance>, Choice = {ps, xs}
    <realm>, Domain | Host (1 to 80 chars)
    <applicationId>, Choice = {Rf, Ro, Sh, All}
    <default>, Choice = {false, true}

AS_CLI/Interface/Diameter/Realms> add ps operator.com Sh false
...Done
```

In addition, entries can be modified (set command) and deleted in the standard CLI fashion.

### 3.3.2 Routing Peers

To associate static peers with realms, the *AS_CLI/Interface/Diameter/Realms/RoutingPeers* level is used. Each entry links a given *instance/realm/application* to a given static peer. In addition, priorities and weights can be set, for use when multiple peer identities match a given instance/realm/application. Entries are then sorted by priority, and same priority entries are randomized according to their weights.

```
This command is used to manage Routing Peers associated to a static Realm
Routing Table entry.

Commands:
    0)         get : show RoutingPeers-related attributes
    1)         add : add a new RoutingPeers
    2)         set : modify RoutingPeers-related attributes
    3)      delete : delete an existing RoutingPeers

  h (help),  e (exit),  q (quit),  r (read),  w (write),  t (tree),
  c (config),  cd (cd),  a (alias),  hi (history),  p (pause),  re (repeat)

AS_CLI/Interface/Diameter/Realms/RoutingPeers> get
  Instance         Realm  ApplicationID          Identity  Priority  Weight
  ============================================================================
        ps   operator.com             Sh   hss2.provider.com         5      10
        xs    billing.net             Rf   cdf1.provider.com         1     200
        xs    billing.net             Rf   cdf2.provider.com         1     100
        xs    billing.net             Rf   cdf3.provider.com         1      50

  4 entries found.
```

Note that the get command above shows static routing peers only and does not show any dynamic information. To see all routing peers (static and dynamic); go to the *AS_CLI/ASDiagnostic/Diameter/Realms/RoutingPeers* level.

Entries are added as follows. The help add command also includes the description of each parameter.

```
AS_CLI/Interface/Diameter/Realms/RoutingPeers> help add
This command is used to add a Routing Peer associated to a static Realm Routing
Table entry. The Routing Peer's instance and identity must correspond to a
Static Peer with the same instance and identity.

Parameters description:
instance     : The process for which the Realm Routing Peer is displayed.
realm        : The Realm Name
applicationId: The application associated with the Realm Routing Table entry.
               All is a wildcard indicating the entry applies for all supported
               Diameter applications.
identity     : The Peer Identity. Must correspond to a Static Peer of the same
               instance and identity. A Routing Peer may correspond to only one
               Realm Routing Table entry.
priority     : The priority of this peer against other routing peers associated
               to the same Realm Routing Table entry.
weight       : The weight used for ordering peers having the same priority.

======================================================================
add
    <instance>, Choice = {ps, xs}
    <realm>, Domain | Host (1 to 80 chars)
    <applicationId>, Choice = {Rf, Ro, Sh, All}
    <identity>, Domain | Host (1 to 80 chars)
    <priority>, Integer {0 to 65535}
    <weight>, Integer {0 to 65535}

AS_CLI/Interface/Diameter/Realms/RoutingPeers> add xs billing.net Rf
       cdf1.provider.com 1 100
...Done
```

As usual, entries can be modified (set command) and deleted in the standard CLI fashion.

## 3.4 Diameter Diagnostics

The Diameter diagnostics CLI level (under *AS_CLI/ASDiagnostic/Diameter*) allows the operator to see live static and dynamic peer and realm information, delete dynamic entries, and debug Diameter routing with the fetchRoutes command.

### 3.4.1 Live Peer Table

The following is an example of a live Peer Table with the Execution Server relay disabled.

```
AS_CLI/ASDiagnostic/Diameter/Peers> get
Instance Id     Type                 Identity   IP Address  Port  Enabled  Trans  Conn
        Down Cause          Last Used
===========================================================================================
        ============================
      ps  1  statichss1.hss.operator.com 192.168.4.20  3868     true I_OPEN  OKAY
        07/17 17:45:00:123
      xs  1  static cdf1.cdf.operator.com 192.168.5.20  3868    false CLOSED  DOWN
        LOCALLY_DISABLED 07/17 17:45:00:321
      xs  2 dynamicocs1.ocs.operator.com 192.168.6.22  3868     true I_OPEN  OKAY
        07/17 17:55:00:456
```

The following is an example of a live Peer Table with the Execution Server relay enabled. It shows an internal connection between the internal Execution Server Diameter stack instance and the Provisioning Server. The related Peer Table entries are defined as static but are not modifiable.

```
AS_CLI/ASDiagnostic/Diameter/Peers> get
    Instance  Id    Type                       Identity     IP Address    Port  Enabled
        Trans  Conn         Down Cause           Last Used
===========================================================================================
        ==================================================
  internalXs   2  static               as.operator.com     127.0.0.1    3350      yes
        R_OPEN  OKAY                    11/27 09:27:32.212
          ps   1  static      internal-as.operator.com     127.0.0.1   13869      yes
        I_OPEN  OKAY                    11/27 09:27:38.962
```

### 3.4.2 Delete Dynamic Peers

The following is an example showing how to delete a dynamic Peer Table entry.

```
AS_CLI/ASDiagnostic/Diameter/Peers> help delete
This command is used to delete a dynamic Peer Table entry.

Parameters description:
instance: The process for which the Peer Table live information is displayed.
id      : The internal id of the peer, as displayed in the "Id" column from the
          "get" command.

======================================================================
delete
    <instance>, Choice = {xs, internalXs, ps}
    <id>, Long {1 to 9223372036854775807}

AS_CLI/ASDiagnostic/Diameter/Peers> delete xs 1
```

### 3.4.3 Live Realm Table

The following is an example of a live Realm Table.

```
AS_CLI/ASDiagnostic/Diameter/Realms> get
Instance Id     Type             Realm  App. ID           Last Used
===================================================================
ps        1   static  hss.operator.com       Sh  07/17 17:45:00:123
ps        2  dynamic  ocs.operator.com       Ro  07/17 17:55:00:456
xs        1   static  hss.operator.com       Sh  07/17 17:55:00:456
xs        2  dynamic  ocs.operator.com       Ro  07/17 18:55:00:221
```

### 3.4.4 Delete Dynamic Realms

The following is an example showing how to delete a dynamic Realm Table entry.

```
AS_CLI/ASDiagnostic/Diameter/Realms> help delete
This command is used to delete a dynamic Realm Routing Table entry along with
all associated Routing Peers. Static Realm Routing Table entries are deleted
using the delete command under the AS_CLI/Interface/Diameter/Realms context.

Parameters description:
instance: The process for which the Realm Routing Table entry will be deleted.
id      : The internal id of the realm, as displayed in the "Id" column from
          the "get" command.

======================================================================
delete
    <instance>, Choice = {xs, ps, internalXs}
    <id>, Long {1 to 9223372036854775807}

AS_CLI/ASDiagnostic/Diameter/Realms> delete xs 1
```

### 3.4.5 Live Routing Peers

The following is an example showing all routing peers associated with either a static or a dynamic Realm Routing Table entry. For dynamic Realm Routing Table entries, the priority and weight were obtained from a domain name system (DNS) service locator (SRV) lookup.

```
AS_CLI/ASDiagnostic/Diameter/Realms/RoutingPeers> get
Instance  Id              Realm  App. ID                 Identity  Priority  Weight
===================================================================================
      ps   1   hss.operator.com       Sh  hss1.hss.operator.com         1      50
      ps   2   hss.operator.com       Sh  hss2.hss.operator.com         1      50
      ps   3   hss.operator.com       Sh  hss3.hss.operator.com         2     100
      xs   1   ocs.operator.com       Ro  ocs1.ocs.operator.com         1      50
      xs   2   ocs.operator.com       Ro  ocs2.ocs.operator.com         1      50
      xs   3   ocs.operator.com       Ro  ocs3.ocs.operator.com         2     100
```

### 3.4.6 Troubleshoot Diameter Routing

The fetchRoutes command can be used to troubleshoot Diameter routing. Be warned that using this command might cause the Diameter stack to create dynamic peer and/or Realm Routing Table entries. The following are some examples.

```
AS_CLI/ASDiagnostic/Diameter> help fetchRoutes
This command is used to troubleshoot the Diameter routing configuration
according to the current Peer Table/Realm Routing Table configuration and
application. This commands displays the potential next hop(s) and final
destination of an ACR (given a CCF), a CCR (given an OCS) or a Sh request. Only
reachable next hops are displayed. Warning: using this command might cause the
stack to create dynamic Peer and/or Realm Routing Table entries.

Parameters description:
instance     : The process for which the routing configuration is fetched.
applicationId: The application for which the routing configuration is fetched.
Rf           : Fetch routes for the Rf applicationId.
ccf          : A CCF, as it would be received in P-CFA header.
realm        : Realm name.
Ro           : Fetch routes for the Ro applicationId.
ocs          : An OCS, as it would be received in P-CFA header.
Sh           : Fetch routes for the Sh applicationId.

=======================================================================
fetchRoutes
    <instance>, Choice = {ps, xs}
    <applicationId>, Choice = {Rf, Ro, Sh}
        Rf:
            <ccf>, String {1 to 1024 characters}
            <realm>, Domain | Host (1 to 80 chars)
        Ro:
            <ocs>, String {1 to 1024 characters}
            <realm>, Domain | Host (1 to 80 chars)
        Sh:
            <realm>, Domain | Host (1 to 80 chars)
```

In this example, direct connections exist to the specified charging data functions (CDFs). The Execution Server Diameter stack instance has a connection to 192.168.4.21 and peer identity is ccf2.ccf.operator.com.

```
AS_CLI/ASDiagnostic/Diameter> fetchRoutes xs Rf
       ccf1.ccf.operator.com,192.168.4.21 operator.com
  Nb                 Next Hop  Destination-Host  Destination-Realm  Routing Mode
==============================================================================
   1  ccf1.ccf.operator.com                                             DIRECT
   2  ccf2.ccf.operator.com                                             DIRECT
```

In this example, the CDF is specified as a Diameter uniform resource identifier (URI) for which the Diameter Stack instance does not have a connection. *RoutingMode* is set to "dynamic".

```
AS_CLI/ASDiagnostic/Diameter> fetchRoutes xs Rf
       \"aaa://dynccf1.ccf.operator.com:3868\" operator.com
  Nb                 Next Hop  Destination-Host  Destination-Realm  Routing Mode
==============================================================================
   1  ccf1.ccf.operator.com                                            DYNAMIC
```

In this example, *routingMode* is set to "realm" and the Diameter stack does not have direct connection to the OCSs. The User-Name AVP in OCS would be something similar to john@operator.com (with the domain part set to "operator.com").

```
AS_CLI/ASDiagnostic/Diameter> fetchRoutes xs Ro \"aaa://dynocs1.ocs.operator.com:3868\"
       operator.com
  Nb                    Next Hop          Destination-Host  Destination-Realm Routing Mode
===========================================================================================
   1  ocsagent1.ocs.operator.com  dynocs1.ocs.operator.com       operator.com        REALM
   2  ocsagent2.ocs.operator.com  dynocs1.ocs.operator.com       operator.com        REALM
```

The final example is for the Sh interface.

```
AS_CLI/ASDiagnostic/Diameter> fetchRoutes ps Sh operator.com
  Nb                 Next Hop  Destination-Host  Destination-Realm  Routing Mode
==============================================================================
   1  hss1.hss.operator.com                          operator.com          REALM
   2  hss2.hss.operator.com                          operator.com          REALM
```

## 3.5 Diameter Stack Logging

The Diameter input channel is dedicated to the Diameter stack (and the Rf, Ro, and Sh applications). It can be enabled independently on the Execution Server and Provisioning Server. For example, to enable the Diameter input channel in the Execution Server and set its severity to "FieldDebug", execute the following command.

```
AS_CLI/Applications/ExecutionAndProvisioning/XS/Logging/InputChannels> set
       Diameter enabled true severity FieldDebug
...Done

AS_CLI/Applications/ExecutionAndProvisioning/XS/Logging/InputChannels> get
                                            Name  Enabled    Severity
=====================================================================
                                         Generic     true       Debug
                                      Accounting     true
                                           CallP     true
                                            SMAP    false
                                             Sip     true
                                        SipMedia     true
                                            MGCP    false
                                           Timer    false
                                          NRSLog    false
                               CommonPersistency     true        Info
           BroadsoftCommonCommunicationTransport     true        Info
  BroadsoftCommonCommunicationTransportKeepAlive     true      Notice
                                       ServiceOS     true
                               EventNotification    false
                                      FileSystem    false
                                      HttpClient    false
                          ProvisioningValidation     true       Debug
                                        Overload     true
                                            SMDI    false
                                            SMPP    false
                                     NameService     true
                                           OCI-C     true        Info
                                      CallCenter     true       Debug
                                             CAP     true       Debug
                                        Diameter     true  FieldDebug
                                StateReplication     true       Debug
                                            ECCR     true      Notice
                                         MediaCr     true      Notice
                                         CallLog     true      Notice
29 entries found.
```

## 3.6 Usage of IPv6

The Application Server and Xtended Services Platform (Xsp) can have Diameter interfaces as shown in *Figure 1.*

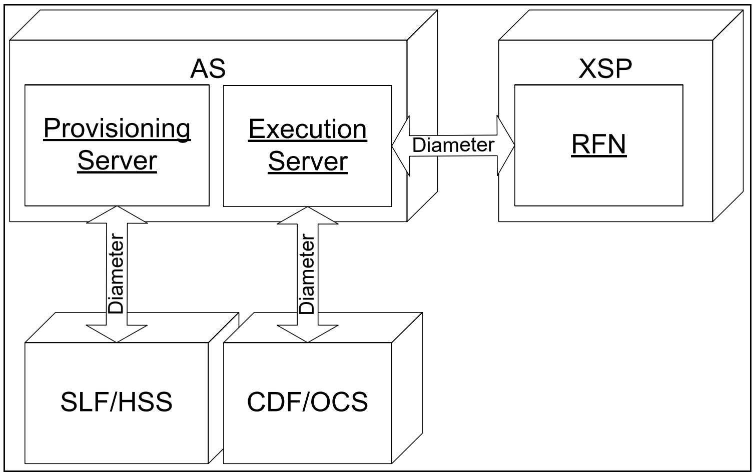

Figure 1 Diameter Messaging Path

Support for IPv6 is available on all these nodes. To use Diameter over IPv6, the following configuration is required:

- The Diameter listening address must be configured with an IPv6 address.
- Statically configured peers in the peer configuration table, if any, must have IPv6 addresses configured, or the address must be left unspecified.

The Diameter stack operates in one of two modes: IPv4 or IPv6. There is no explicit configuration; the mode is inferred from the listening address. The Diameter stack operates in IPv6 mode if the listening address is an IPv6 address. Otherwise, the Diameter stack operates in IPv4 mode.

Note that the different instances of the Diameter stack can be independently configured to use IPv4 or IPv6. Therefore, it is possible to have connections to the SLF/HSS using IPv6 while the connections to CDF/OCS use IPv4, or vice versa.

## 3.7 Government Emergency Telecommunications Service

For Government Emergency Telecommunications Service (GETS) calls, the Ro interface for online billing is modified to have the Session-Priority AVP (with the configurable *defaultPriorityAVP* system parameter value) in the Diameter Credit Control Request (CCR) asking for credits. This AVP is part of the IMS-Information group AVP in the CCR GETS. For more information, see the *Support Government Emergency Telecommunications Service (GETS) Feature Description* [11].

The following configurations are available at the Application Server CLI level.

```
AS_CLI/System/CallP/GETS> get
  enabled = true
  enableRequireResourcePriority = false
  sendAccessResourcePriority = false
  callIdentifierMode = rph
  defaultPriorityAVP = 2
  signalingDSCP = 45
  defaultRValue = ets.0
  bypassRoRelease = true
```

# 4 Rf Interface

This section describes the steps required to configure the Rf interface. It assumes that the Application Server is running and Realtime Accounting is licensed. For an overview of the Rf interface behavior, see the *BroadWorks Rf/Ro Interface Specification* [6].

Configuration of the Rf interface consists of the following steps:

1) Configure the Diameter stack, peers and realms.
2) Configure the default servers.
3) Enable Diameter Offline Accounting.
4) Select record content, optional triggers, and routing mode.

## 4.1 Diameter Configuration

For accounting to be able to send Rf messages, the Diameter stack and peers (and optionally realms) must be configured as documented in section *3 Diameter Stack*. Add as many peers and realms (if the routing mode is realm) as required to connect to the offline billing servers.

## 4.2 Default Server Configuration

Default servers are configured on the Application Server. They are used as the destination of accounting requests when the list of desired servers has not been received in the *P-Charging-Function-Addresses SIP* header or when not in IP Multimedia Subsystem (IMS) mode.

It is recommended to configure at least two charging collection function (CCF) servers. The first one acts as primary, and the second one acts as secondary, and so on. Note that these values are ignored if a *P-Charging-Function-Addresses SIP* header is received in IMS mode. For more information, see the *BroadWorks Rf/Ro Interface Specification* [6].

Enter the Application Server CLI and go to the menu.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/ChargingFunct
ionElement>
```

Servers can be configured by their Diameter identity or their IP address as follows.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/ChargingFunct
ionElement> add primary.domain.com false ccf
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/ChargingFunct
ionElement> add secondary.domain.com false ccf
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/ChargingFunct
ionElement> get
               Address  Net Address Extended   Type  Description
  ===============================================================
    primary.domain.com                  false   CCF
  secondary.domain.com                  false   CCF

  2 entries found.
```

Alternatively, servers can be configured with a Diameter URI as follows.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/ChargingFunct
ionElement> add aaa://primary.domain.com true ccf
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/ChargingFunct
ionElement> add aaa://secondary.domain.com true ccf
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/ChargingFunct
ionElement> get
                     Address  Net Address Extended   Type  Description
  ====================================================================
    aaa://primary.domain.com                  true   CCF
  aaa://secondary.domain.com                  true   CCF

  2 entries found.
```

The first column, *Address*, contains the Diameter identity, the IP address, or the Diameter URI of the peer.  The Diameter URI must start with aaa:// (note, however, that aaas:// is not supported).

The second column, *Net Address Extended*, is set to “true” for the Diameter URI, and is set to “false” otherwise.  The last column is always set to “CCF” for the Rf interface.

## 4.3 Enable Diameter Offline Accounting

To enable Diameter Offline Accounting (Rf interface), accounting must first be enabled.

Enter the Application Server CLI and go to the menu.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface>
```

Accounting is enabled as follows.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface> set enabled true
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface> get
  enabled = true
  longCallDurationRecordControl = enabled
  longCallDurationTimerMinutes = 1440
  enableCDRCodecChanges = false
  enableCDRInternalConference = false
  enableCDRStartEnd = true
  enableFailoverInfo = true
  enableIntraGroupCDR = true
  enableModuleCentrex = true
  enableModuleIP = true
  enableModule3GPP = true
  enableTerminatingCDR = true
  use3xxAsRouteConfirmationForAccounting = true
  customSchemaVersion =
  enableAVPQuotes = false
```

Once accounting is enabled, the Rf interface is enabled from the menu.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline>
```

Execute the following command.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> set
enabled true
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> get
  enabled = true
  sendBroadWorksInfo = true
  send3GPPInfo = true
  enableFileQueuing = false
  fileRetentionHours = 1
  routingMode = Direct
  pollingIntervalSeconds = 1800
```

## 4.4 Select Optional Content, Triggers, Routing Mode, and Polling Interval

The BroadWorks Application Server can be configured to tailor the content and the circumstances where accounting requests are sent. This can be used to adjust the message size and the message frequency.

The following command can be used to control whether or not BroadSoft vendor-specific AVPs are included. BroadSoft vendor-specific AVPs are included if the value is “true”. They are not included if the value is “false”. BroadSoft vendor-specific AVPs are defined in the *BroadWorks Accounting Call Detail Record Interface Specification* [5] and the *BroadWorks Rf/Ro Interface Specification* [6].

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> set
sendBroadWorksInfo true
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> get
  enabled = true
  sendBroadWorksInfo = true
  send3GPPInfo = true
  enableFileQueuing = false
  fileRetentionHours = 1
  routingMode = Direct
  pollingIntervalSeconds = 1800
```

The following command can be used to control whether or not 3GPP optional AVPs are included. 3GPP optional AVPs are included if the value is “true”. They are not included if the value is “false”. A list of the affected AVPs is provided in the *BroadWorks Rf/Ro Interface Specification* [6].

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> set
send3GPPInfo true
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> get
  enabled = true
  sendBroadWorksInfo = true
  send3GPPInfo = true
  enableFileQueuing = false
  fileRetentionHours = 1
  routingMode = Direct
  pollingIntervalSeconds = 1800
```

The following command can be used to inhibit the generation of specific AVPs individually. Note that this inhibits the AVP from both the Rf and the Ro interfaces.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/
InhibitedAVPCodeList> add 886 10415
  ...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/diameter/
InhibitedAVPCodeList> get
avpCode  vendorId
====================
  886     10415
```

The following command can be used to determine how the *Calling-Party-Address* AVP (3GPP AVP code 831) is populated. For details, see the *BroadWorks Rf/Ro Interface Specification* [6]. Note that this setting applies to both the Rf and the Ro interfaces.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter> set
callingPartyAddressStrictCompliance false
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter> get
  defaultDestinationRealm =
  useRealmFromCapabilitiesExchange = true
  callingPartyAddressStrictCompliance = false
```

The following command can be used to determine how the *Called-Party-Address* AVP (3GPP AVP code 832) is populated. For details, see the *BroadWorks Rf/Ro Interface Specification* [6]. Note that this setting applies to both the Rf and the Ro interfaces.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter> set
calledPartyAddressStrictCompliance false
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter> get
  defaultDestinationRealm =
  useRealmFromCapabilitiesExchange = true
  callingPartyAddressStrictCompliance = false
  calledPartyAddressStrictCompliance = false
```

When an accounting request cannot be delivered, BroadWorks can optionaly save the request to disk until the server(s) becomes available. This behavior is enabled using the following command and is turned off in a similar manner.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> set
enableFileQueuing true
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> get
  enabled = true
  sendBroadWorksInfo = true
  send3GPPInfo = true
  enableFileQueuing = true
  fileRetentionHours = 1
  routingMode = Direct
  pollingIntervalSeconds = 1800
  backlogMaxParallelTransactions = 1
```

BroadWorks periodically checks for the presence of requests saved to disk, and attempts to send them to the CDF.  This is done every 30 minutes by default, but can be changed to a different value as follows.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> set
pollingIntervalSeconds 900
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> get
  enabled = true
  sendBroadWorksInfo = true
  send3GPPInfo = true
  enableFileQueuing = true
  fileRetentionHours = 1
  routingMode = Direct
  pollingIntervalSeconds = 900
  backlogMaxParallelTransactions = 1
```

When using the file queuing capability, it is possible to configure the maximum time that the requests are kept on disk.  Requests are deleted from the disk when they have been successfully delivered or when the maximum time is reached.  The maximum time prevents the files from filling the disk.  The maximum time is expressed in hours and is a value from 1 through 168 hours (one week).

Note that the size of the accounting records queued on disk is much bigger than the size of the raw binary information being stored.  An estimate of 6 to 10 times the size of the raw data should be used to evaluate the required disk space.  The actual size varies depending on factors such as provisioning name lengths and enabled modules or services.  Also note that the call rate is an important variable in estimating the disk space required for file queuing.

For example, if the average size of AVP (over a given time period) is 2.5 kbytes, then those AVPs (for the same time period) convered to XML will require 15 to 25 kbytes of disk space.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> set
fileRetentionHours 24
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> get
  enabled = true
  sendBroadWorksInfo = true
  send3GPPInfo = true
  enableFileQueuing = true
  fileRetentionHours = 24
  routingMode = Direct
  pollingIntervalSeconds = 1800
  backlogMaxParallelTransactions = 1
```

When using the file queuing capability, BroadWorks limits the number of outstanding ACR being retransmitted during recovery to avoid overloading the CDF. Once the limit is reached, BroadWorks wait for the CDR to answer before sending new requests. Per default, only one outstanding request is allowed, but the value can be configured using the *backlogMaxParallelTransactions* parameter.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> set
backlogMaxParallelTransactions 1
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> get
  enabled = false
  sendBroadWorksInfo = true
  send3GPPInfo = true
  enableFileQueuing = false
  fileRetentionHours = 1
  routingMode = Direct
  pollingIntervalSeconds = 1800
  backlogMaxParallelTransactions = 1
```

The routing mode controls how the peers and realms are used.

- When set to “direct”, it indicates that the Execution Server shall have a direct connection to the CDFs specified in the *P-Charging-Function-Addresses (P-CFA)* header. For a given Rf accounting session, the CDFs are tried in the order received in the P-CFA; if no Diameter connection exists for a given CDF, the next CDF from the P-CFA is tried.
- When set to “dynamic”, this is similar to direct, except that if no direct connection exists to a given CDF and the Execution Server tries to dynamically establish a Diameter connection to the CDF using the information available in the P-CFA. This results in a dynamic entry to be created in the Peer Table (this does not impact the Realm Routing Table). Note that the entry is added even if the connection fails.
- When set to “realm”, it indicates that realm routing procedures are applied in case the Execution Server does not have direct connection to the CDF. The realm is extracted from the User-Name AVP present in the ACR. Note that using realm routing for Rf implies that the Destination-Host AVP is present in the ACRs, which is not compliant with the ACR Augmented Backus-Naur Format (ABNF) as defined in *3GPP TS 32.299*. The Destination-Host MUST be inserted in realm-based routing for a request to reach a specific server, which in this case is a given CDF.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> set
routingMode realm
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Offline> get
  enabled = true
  sendBroadWorksInfo = true
  send3GPPInfo = true
  enableFileQueuing = true
  fileRetentionHours = 24
  routingMode = Realm
  pollingIntervalSeconds = 1800
  backlogMaxParallelTransactions = 1
```

BroadWorks can optionally generate long duration CDRs periodically for answered calls. This behavior can be enabled, disabled, or controlled by the Diameter server.

To enable this behavior, set the *longCallDurationRecordControl* parameter to “enabled” and set the period for *longCallDurationTimerMinutes*.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface> set
longCallDurationRecordControl enabled
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface> set
longCallDurationTimerMinutes 10
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface> get
  enabled = true
  longCallDurationRecordControl = enabled
  longCallDurationTimerMinutes = 10
  enableCDRCodecChanges = false
  enableCDRInternalConference = false
  enableCDRStartEnd = true
  enableFailoverInfo = true
  enableIntraGroupCDR = true
  enableModuleCentrex = true
  enableModuleIP = true
  enableModule3GPP = true
  enableTerminatingCDR = true
  use3xxAsRouteConfirmationForAccounting = true
```

To disable this behavior, set the *longCallDurationRecordControl* parameter to “disabled”.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface> set
longCallDurationRecordControl disabled
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface> get
  enabled = true
  longCallDurationRecordControl = disabled
  longCallDurationTimerMinutes = 10
  enableCDRCodecChanges = false
  enableCDRInternalConference = false
  enableCDRStartEnd = true
  enableFailoverInfo = true
  enableIntraGroupCDR = true
  enableModuleCentrex = true
  enableModuleIP = true
  enableModule3GPP = true
  enableTerminatingCDR = true
  use3xxAsRouteConfirmationForAccounting = true
```

It is also possible to let the Diameter server control this behavior on a call-by-call basis by setting the *longCallDurationRecordControl* parameter to “controlledByDiameterServer”. The long duration CDRs are then controlled by the Acct-Interim-Interval AVP as described in the *BroadWorks Rf/Ro Interface Specification* [6].

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface> set
longCallDurationRecordControl controlledByDiameterServer
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface> get
  enabled = true
  longCallDurationRecordControl = controlledByDiameterServer
  longCallDurationTimerMinutes = 10
  enableCDRCodecChanges = false
  enableCDRInternalConference = false
  enableCDRStartEnd = true
  enableFailoverInfo = true
  enableIntraGroupCDR = true
  enableModuleCentrex = true
  enableModuleIP = true
  enableModule3GPP = true
  enableTerminatingCDR = true
  use3xxAsRouteConfirmationForAccounting = true
```

There are other parameters that affect the behavior of the accounting subsystem in the same menu.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface>
```

These parameters are described in the *BroadWorks Accounting Call Detail Record Interface Specification* [5].

# 5 Ro Interface

This section describes the steps required to configure the Ro interface. It assumes that the Application Server is running. Note that the Realtime Accounting system license is not required for the Ro interface since there are user licenses for the Prepaid and Advice of Charge services (the two services making use of the Ro interface). For an overview of the Ro interface behavior, see the *BroadWorks Rf/Ro Interface Specification* [6]. For a description of the Prepaid service, see the *IMS Accounting Ro Interface Feature Description* [9]. For a description of the Advice of Charge service, see the *Advice of Charge Feature Description* [8].

Configuration of the Ro interface consists of the following steps:

1) Configure the Diameter stack, peers, and realms.

2) Configure the default servers.

3) Enable Diameter Online Accounting.

4) Configure cost information source.

5) Select request content, optional triggers, and routing mode.

## 5.1 Diameter Configuration

For accounting to be able to send Ro requests, the Diameter stack and peers (and optionally realms) must be configured as documented in section *3 Diameter Stack*. Add as many peers and realms (if the routing mode is realm) as required to connect to the online billing servers.

## 5.2 Default Server Configuration

Default servers are configured on the Application Server. They are used as the destination of credit-control accounting requests for Prepaid subscribers when the list of desired servers has not been received in the *P-Charging-Function-Addresses SIP* header or when not in IMS mode. They are not used by Advice of Charge requests, which are configured separately. For more information, see section *5.4 Configure Cost Information Source* and the *Advice of Charge Feature Description* [8].

It is recommended that at least two online charging function (OCF) servers be configured. The first one acts as primary, the second one acts as secondary, and so on. Note that these values are ignored if a *P-Charging-Function-Addresses SIP* header is received in IMS mode. For more information, see the *BroadWorks Rf/Ro Interface Specification* [6].

Enter the Application Server CLI and go to the menu.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/ChargingFunct
ionElement>
```

Servers can be configured by their Diameter identity or their IP address as follows.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/ChargingFunct
ionElement> add primary.domain.com false ecf
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/ChargingFunct
ionElement> add secondary.domain.com false ecf
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/ChargingFunct
ionElement> get
```

```
               Address  Net Address Extended  Type  Description
  =============================================================
    primary.domain.com                 false   ECF
  secondary.domain.com                 false   ECF

  2 entries found.
```

Alternatively, servers can be configured with a Diameter URI as follows.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/ChargingFunct
ionElement> add aaa://primary.domain.com true ecf
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/ChargingFunct
ionElement> add aaa://secondary.domain.com true ecf
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/ChargingFunct
ionElement> get
                     Address  Net Address Extended  Type  Description
  ===================================================================
    aaa://primary.domain.com                  true   ECF
  aaa://secondary.domain.com                  true   ECF

  2 entries found.
```

The first column, *Address*, contains the Diameter identity, the IP address, or the Diameter URI of the peer. The Diameter URI must start with aaa:// (aaas:// is not supported).

The second column, *Net Address Extended*, is set to "true" for the Diameter URI and is set to "false" otherwise. The last column is always set to "ECF".

## 5.3 Enable Diameter Online Accounting

To enable Diameter Online Accounting (Ro interface), accounting must first be enabled.

Enter the Application Server CLI and go to the menu.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface>
```

Accounting is enabled as follows.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface> set enabled true
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface> get
  enabled = true
  longCallDurationRecordControl = enabled
  longCallDurationTimerMinutes = 1440
  enableCDRCodecChanges = false
  enableCDRInternalConference = false
  enableCDRStartEnd = true
  enableFailoverInfo = true
  enableIntraGroupCDR = true
  enableModuleCentrex = true
  enableModuleIP = true
  enableModule3GPP = true
  enableTerminatingCDR = true
  use3xxAsRouteConfirmationForAccounting = true
  customSchemaVersion =
  enableAVPQuotes = false
```

Once accounting is enabled, the Ro interface is enabled from the menu.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online>
```

Execute the following command.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> set
enabled true
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> get
  enabled = true
  simultaneousOnlineOfflineBilling = false
  defaultTimeQuotaThresholdSeconds = 30
  defaultCreditControlFailureHandling = Retry And Terminate
  unitDetermination = Centralized
  timeSliceSeconds = 300
  warningToneThresholdSeconds = 30
  enableBroadsoftAvps = true
  routingMode = Direct
  txTimerSeconds = 5
```

## 5.4 Configure Cost Information Source

To enable the optional Advice of Charge Ro queries, the cost information source must be configured for the service. For complete information, see the *Advice of Charge Feature Description* [8].

Enter the Application Server CLI and go to the menu.

```
AS_CLI/Service/AdviceOfCharge>
```

The cost information source is set as follows.

```
AS_CLI/Service/AdviceOfCharge> set costInformationSource rfn.domain.com
...Done

AS_CLI/Service/AdviceOfCharge> get
  delayBetweenNotificationInSeconds = 15
  incomingAocHandling = charge
  costInformationSource = rfn.domain.com
```

## 5.5 Select Optional Content, Triggers, and Routing Mode

The BroadWorks Application Server can be configured to tailor the content and the circumstances where credit control requests are sent. This can be used to adjust the message size and the message frequency.

The following command can be used to control whether or not BroadSoft vendor-specific AVPs are included. BroadSoft vendor-specific AVPs are included if the value is “true”. They are not included if the value is “false”. BroadSoft vendor-specific AVPs are defined in the *BroadWorks Accounting Call Detail Record Interface Specification* [5] and the *BroadWorks Rf/Ro Interface Specification* [6].

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> set
enableBroadsoftAvps true
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> get
  enabled = true
  simultaneousOnlineOfflineBilling = false
  defaultTimeQuotaThresholdSeconds = 30
  defaultCreditControlFailureHandling = Retry And Terminate
```

```
   unitDetermination = Centralized
   timeSliceSeconds = 300
   warningToneThresholdSeconds = 30
   enableBroadsoftAvps = true
   routingMode = Direct
   txTimerSeconds = 5
```

The following command can be used to inhibit the generation of specific AVPs individually. Note that this inhibit the AVP from both the Rf and the Ro interfaces.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/
InhibitedAVPCodeList> add 886 10415
  ...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/diameter/
InhibitedAVPCodeList> get
avpCode  vendorId
====================
  886      10415
```

The following command can be used to determine how the *Calling-Party-Address* AVP (3GPP AVP code 831) is populated.  For details, see the *BroadWorks Rf/Ro Interface Specification* [6].  Note that this setting applies to both the Rf and Ro interfaces.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter> set
callingPartyAddressStrictCompliance false
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter> get
  defaultDestinationRealm =
  useRealmFromCapabilitiesExchange = true
  callingPartyAddressStrictCompliance = false
```

The following command can be used to determine how the *Called-Party-Address* AVP (3GPP AVP code 832) is populated.  For details, see the *BroadWorks Rf/Ro Interface Specification* [6].  Note that this setting applies to both the Rf and the Ro interfaces.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter> set
calledPartyAddressStrictCompliance false
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter> get
  defaultDestinationRealm =
  useRealmFromCapabilitiesExchange = true
  callingPartyAddressStrictCompliance = false
  calledPartyAddressStrictCompliance = false
```

The *serviceContextID* configuration is used to populate the Service-Context-Id (AVP 461) for Prepaid CCR.  Similarly, the *serviceContextIDAoC* is used to populate the Service-Context-Id (AVP 461) for Advice of Charge tariff information requests.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> set
serviceContextID 8.32260@3gpp.org
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> set
serviceContextIDAoC 11.32280@3gpp.org
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> get
  enabled = false
  simultaneousOnlineOfflineBilling = false
  defaultTimeQuotaThresholdSeconds = 30
  defaultCreditControlFailureHandling = Retry And Terminate
  unitDetermination = Centralized
  timeSliceSeconds = 300
  warningToneThresholdSeconds = 30
  enableBroadsoftAvps = true
  routingMode = Direct
  txTimerSeconds = 5
  serviceContextID = 8.32260@3gpp.org
  serviceContextIDAoC = 11.32280@3gpp.org
```

The following command can be used to control whether or not offline accounting is also enabled for calls that are subject to online billing.  Calls not subject to online billing are not affected.  When set to "false", offline accounting messages (ACR) are not sent for Prepaid users.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> set
simultaneousOnlineOfflineBilling false
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> get
  enabled = true
  simultaneousOnlineOfflineBilling = false
  defaultTimeQuotaThresholdSeconds = 30
  defaultCreditControlFailureHandling = Retry And Terminate
  unitDetermination = Centralized
  timeSliceSeconds = 300
  warningToneThresholdSeconds = 30
  enableBroadsoftAvps = true
  routingMode = Direct
  txTimerSeconds = 5
```

The following command can be used to configure the default time quota threshold.  This is the time margin at which the Application Server requests new credits before the current credits are exhausted or the validity is reached.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> set
defaultTimeQuotaThresholdSeconds 30
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> get
  enabled = true
  simultaneousOnlineOfflineBilling = false
  defaultTimeQuotaThresholdSeconds = 30
  defaultCreditControlFailureHandling = Retry And Terminate
  unitDetermination = Centralized
  timeSliceSeconds = 300
  warningToneThresholdSeconds = 30
  enableBroadsoftAvps = true
  routingMode = Direct
  txTimerSeconds = 5
```

To control the behavior of credit failure, set *defaultCreditControlFailureHandling* to one of the following values:

- "continue" – The call is allowed if the connection to the OCF is lost.
- "terminate" – The call is released.
- "retryAndTerminate" – A connection to backup OCF(s) is attempted if failover is allowed.  If the backup servers all fail, the call is released.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> set
defaultCreditControlFailureHandling continue
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> get
  enabled = true
  simultaneousOnlineOfflineBilling = false
  defaultTimeQuotaThresholdSeconds = 30
  defaultCreditControlFailureHandling = Continue
  unitDetermination = Centralized
  timeSliceSeconds = 300
  warningToneThresholdSeconds = 30
  enableBroadsoftAvps = true
  routingMode = Direct
  txTimerSeconds = 5
```

To have the OCF determine how many credits (seconds) to allocate, set the following parameter to “centralized”. If set to “decentralized”, the Application Server requests *timeSliceSeconds* to determine the number of credits.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> set
unitDetermination decentralized
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> get
  enabled = true
  simultaneousOnlineOfflineBilling = false
  defaultTimeQuotaThresholdSeconds = 30
  defaultCreditControlFailureHandling = Continue
  unitDetermination = Decentralized
  timeSliceSeconds = 300
  warningToneThresholdSeconds = 30
  enableBroadsoftAvps = true
  routingMode = Direct
  txTimerSeconds = 5
```

To set the amount of time requested by the Application Server for a credit control request (CCR) Initial or a CCR Update in the Requested-Service-Unit AVP, set the following parameter. This is only applicable when *unitDetermination* is set to “decentralized”.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> set
timeSliceSeconds 300
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> get
  enabled = true
  simultaneousOnlineOfflineBilling = false
  defaultTimeQuotaThresholdSeconds = 30
  defaultCreditControlFailureHandling = Continue
  unitDetermination = Decentralized
  timeSliceSeconds = 300
  warningToneThresholdSeconds = 30
  enableBroadsoftAvps = true
  routingMode = Direct
  txTimerSeconds = 5
```

To set the number of seconds before credits run out at which the Application Server plays the low credits warning tone, set the following parameter.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> set
warningToneThresholdSeconds 30
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> get
  enabled = true
  simultaneousOnlineOfflineBilling = false
  defaultTimeQuotaThresholdSeconds = 30
  defaultCreditControlFailureHandling = Continue
  unitDetermination = Decentralized
  timeSliceSeconds = 300
  warningToneThresholdSeconds = 30
  enableBroadsoftAvps = true
  routingMode = Direct
  txTimerSeconds = 5
```

The routing mode controls how the peers and realms are used.

- When set to “direct”, this indicates that the Execution Server shall have a direct connection to the OCFs specified in the *P-CFA* header. For a given Ro accounting session, the CDFs are tried in the order received in the P-CFA; if no Diameter connection exists for a given OCF, the next OCF from the P-CFA is tried.
- When set to “dynamic”, it is similar to direct, except that if no direct connection exists to a given OCF, the Execution Server tries to dynamically establish a Diameter connection to the OCF using the information available in the P-CFA. This results in a dynamic entry to be created in the Peer Table (this does not impact the Realm Routing Table). Note that the entry is added even if the connection fails.
- When set to “realm”, this indicates that realm routing procedures are applied in case the Execution Server does not have direct connection to the OCF. The realm is extracted from the User-Name AVP present in the ACR.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> set
routingMode dynamic
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> get
  enabled = true
  simultaneousOnlineOfflineBilling = false
  defaultTimeQuotaThresholdSeconds = 30
  defaultCreditControlFailureHandling = Retry And Terminate
  unitDetermination = Centralized
  timeSliceSeconds = 300
  warningToneThresholdSeconds = 30
  enableBroadsoftAvps = true
  routingMode = Dynamic
  txTimerSeconds = 5
```

The Tx timer controls how long call setup is allowed to be suspended before credit control failure handling is applied. It is not the actual time allocated to the OCF to answer to a CCR, which can be longer (and multiple servers can be attempted). For more information, see the *BroadWorks Rf/Ro Interface Specification* [6].

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> set txTimerSeconds 3
...Done

AS_CLI/Interface/Accounting/BroadWorksCDRInterface/Diameter/Online> get
  enabled = false
  simultaneousOnlineOfflineBilling = false
  defaultTimeQuotaThresholdSeconds = 30
  defaultCreditControlFailureHandling = Retry And Terminate
  unitDetermination = Centralized
  timeSliceSeconds = 300
  warningToneThresholdSeconds = 30
  enableBroadsoftAvps = true
  routingMode = Dynamic
  txTimerSeconds = 3
```

There are other parameters that affect the behavior of the accounting subsystem in the same menu.

```
AS_CLI/Interface/Accounting/BroadWorksCDRInterface>
```

These parameters are described in the *BroadWorks Accounting Call Detail Record Interface Specification* [5].

# 6 Sh Interface

The steps in this section assume that the Application Server is running and the Sh Interface is licensed. In addition, there are certain requirements on the HSS regarding implicitly-registered identity sets. For more information, see section *6.6 HSS Requirements.*

Configuring the Sh interface consists of the following steps:

1) Enabling Public Identity Reporting.
2) Configuring Diameter stack, HSS peers, and realms.
3) Enabling ShInterface Input Channel.
4) Configuring HSS Realm.
5) Configuring DNS SRV Records.

## 6.1 Enable IMS Public Identity Reporting

To synchronize user data over the Sh interface when public identities are created, public identity reporting must be enabled.

```
AS_CLI/Interface/OCI/Reporting> set enablePublicIdentityReporting true
A restart is required for changes to the following attributes to take
effect:
{ enablePublicIdentityReporting }
...Done
```

A restart is required for this change to take effect.

## 6.2 Diameter Configuration

For the Sh interface to be able to send and receive messages, the Diameter stack, peers, and realms must be configured, as documented in section *3 Diameter Stack*. Add as many peers and realms (the routing mode is always realm for the Sh interface) as required to connect to the HSS servers.

## 6.3 Enable ShInterface Input Channel

The *ShInterface* input channel can be enabled on the Provisioning Server as follows.

With the default severity of "Info", the key contents of the Sh interface messages sent and received by the Provisioning Server are logged in human-readable form.

```
AS_CLI/Applications/ExecutionAndProvisioning/PS/Logging/InputChannels>
set ShInterface enabled true severity Info
...Done

AS_CLI/Applications/ExecutionAndProvisioning/PS/Logging/InputChannels>
get
                                            Name  Enabled  Severity
===================================================================
                                         Generic     true     Debug
                                           Timer     true
                                           Audit     true
                               CommonPersistency     true      Info
           BroadsoftCommonCommunicationTransport     true      Info
  BroadsoftCommonCommunicationTransportKeepAlive     true    Notice
                                       ServiceOS     true
                                            SMAP    false
```

```
                                     InstantConf     true
                                      FileSystem    false
                                      HttpClient    false
                                     NameService     true
                                     ShInterface     true       Info
                                DeviceManagement     true
                                        Diameter    false
                                      CallCenter     true
                        BroadWorksMobileManager    false       Info
                                           OCI-P     true       Info

18 entries found.
```

The specific Diameter stack messaging and logs are output on the *Diameter* input channel on both the Provisioning Server and Execution Server, similar to how it is done for the other interfaces.

## 6.4 Configure HSS Realm

The Application Server can interface directly with an HSS farm or through an SLF.  The *hssRealm* attribute must be configured, as this value is always used with the Sh interface for Diameter realm routing.

Following is a configuration example in which the Application Server interfaces with an HSS farm whose realm is *hss.ims.core.telco.com*.

```
AS_CLI/Interface/Sh> set hssRealm hss.ims.core.telco.com
...Done
```

## 6.5 Configure DNS SRV Records

Depending on the desired deployment rules, the DNS server can be configured with SRV records that resolve the hssRealm to the peer identities of the HSS nodes.

The Application Server Sh interface implementation examines the priority fields returned during an SRV query.  For even load balancing, the priority field must be identical for all SRV records that correspond to the hssRealm.  For an active-standby scheme, the priority field of the hosts must be different.

**NOTE**: The Application Server does not use the weight or port returned in the DNS SRV query.

For example, if hssRealm hss.ims.core.telco.com represents an HSS farm that consists of nodes hss1.hss.ims.core.telco.com through hss4.hss.ims.core.telco.com, then DNS SRV records for _diameter._tcp.hss.ims.core.telco.com resolving to hostnames hss1.hss.ims.core.telco.com through hss4.hss.ims.core.telco.com must be added.  An example from a DNS server configuration file is included for illustrative purposes.  Exact syntax, format, and content requirements vary by DNS server implementation.  For the full details on administering SRV records, see your DNS server's documentation.

```
_diameter._tcp.hss.ims.core.telco.com     IN     SRV     1     0     3868
hss1.hss.ims.core.telco.com
_diameter._tcp.hss.ims.core.telco.com     IN     SRV     1     0     3868
hss2.hss.ims.core.telco.com
_diameter._tcp.hss.ims.core.telco.com     IN     SRV     1     0     3868
hss3.hss.ims.core.telco.com
_diameter._tcp.hss.ims.core.telco.com     IN     SRV     1     0     3868
hss4.hss.ims.core.telco.com
```

## 6.6 HSS Requirements

To provide proper functionality in an IMS network, the operator must properly configure implicitly-registered identity sets at the HSS. For each BroadWorks user, the following sets must be defined:

- A set consisting of the user's Session Initiation Protocol (SIP) uniform resource locator (URL) public identity, primary directory number (DN) TEL URL, and alternate DN TEL URLs. This ensures that the HSS would push S-CSCFName and IMSUserState against the user's primary SIP URL public identity if the device performs IMS registration with TEL-URI so that the Application Server has an S-CSCFName to direct out-of-the-blue requests to for such devices that register with TEL-URI.
- For each secondary SIP URI public identity, a one-member set consisting of the secondary SIP URL public identity.
- For the Video Add-On SIP URI public identity, a one-member set consisting of the Video Add-On SIP URL public identity.

For further information on the IMS public identities defined in the Application Server, see the *BroadWorks IMS Public Identities Feature Description, Release 14.0* [1].

## 6.7 Other Sh Interface-related Configuration via CLI

- *AS_CLI/Interface/Sh> publicIdentityRefreshDelayInSeconds*: This specifies the amount of time the Application Server waits when there is an addition of an IMS public identity before triggering the corresponding public identity refresh procedure. This capability provides provisioning systems that listen to Open Client Interface (OCI) reports with enough time to add the public identity to the HSS, so that the public identity refresh procedure can be executed against a known public identity in the HSS.
- *AS_CLI/Interface/Sh/NonTransparentData>* get/set/clear/refresh: Examine or modify the data retrieved from the HSS for a specific identity.
- *AS_CLI/Interface/Sh/NonTransparentData/SystemRefresh>* query/start/terminate: Initiate, stop, and monitor a refresh of the HSS data for all identities in the system.
- *AS_CLI/System/StartupParam>* bw.sh.systemRefreshPublicIdentitiesPerHour: This specifies the rate at which constituent public identity refresh procedures are triggered during a system refresh.

**NOTE**: A restart is required for changes to bw.sh.systemRefreshPublicIdentitiesPerHour to take effect.

## 6.8 Disable Sh Interface

Sh interface signaling can be disabled by clearing the *hssRealm* attribute at the *AS_CLI/Interface/Sh* level.

## 6.9 Supported Application Server Deployment Topologies (Relay)

In BroadWorks, the Sh interface is implemented in the Provisioning Server while the Rf and Ro interfaces are implemented in the Execution Server. The operator has the choice of either configuring two distinct Diameter interfaces (dual Diameter interfaces, that is, one for the Provisioning Server and one for the Execution Server) or only one. In the latter case, all Sh traffic is relayed via the Execution Server using an internal Diameter interface between the Provisioning Server and the Execution Server.

When operating in dual Diameter interfaces mode, Diameter stack instances are assigned distinct Diameter identities, listening IP addresses, ports, Peer Tables, Realm Routing Tables, and so on.

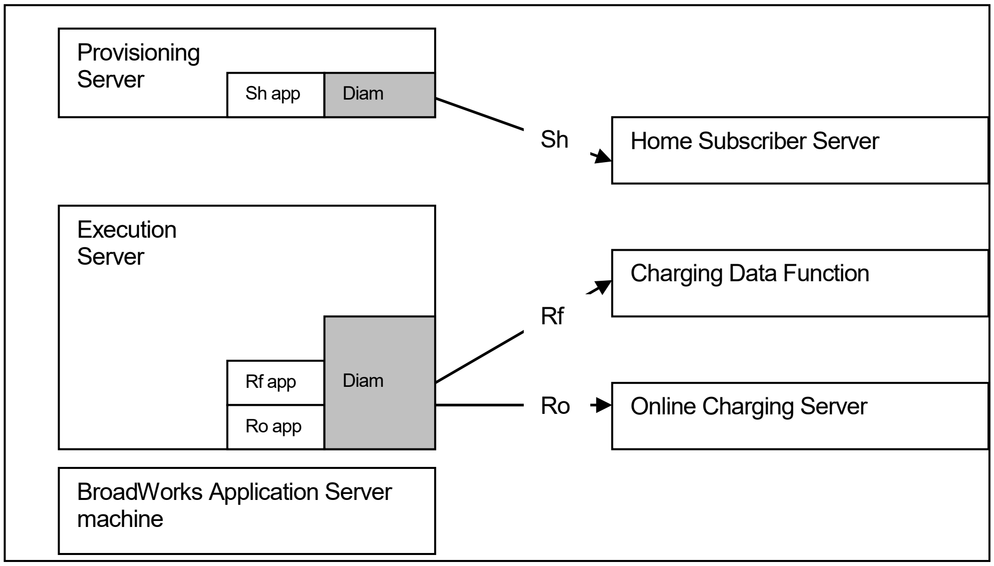

Figure 2  Dual Diameter Interfaces

Conversely, when the Execution Server relays the Sh Diameter traffic, outgoing request forwarding/routing is done based only on the configured Execution Server Peer And Realm Routing Tables (that is, the Provisioning Server blindly forwards all outgoing traffic to the Execution Server, regardless of the configured Peer and Realm Routing Tables on the Provisioning Server, and then the Execution Server takes care of routing outgoing Sh requests to the HSS or next hop Diameter agent).  Diameter message relay between the Provisioning Server and the Execution Server is done through an internal Diameter link, bound on the local host interface, that is, 127.0.0.1.

When using relay, a single Diameter identity is required (although listening ports must be configured for the internal Diameter stack instances).

Note that the internal Diameter stack instance on the Execution Server is active only when relay is enabled and the Sh interface is enabled on the Provisioning Server.

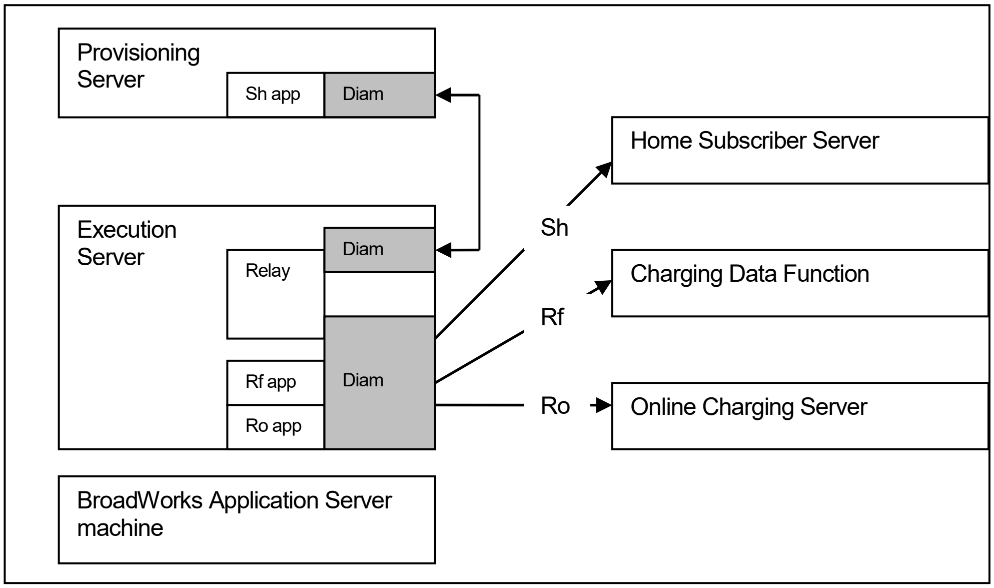

Figure 3 Relayed Diameter Interfaces

Choosing between dual Diameter interfaces or relayed Diameter interfaces depends on the network configuration and the deployed Diameter applications. For example, dual Diameter interfaces are typically used if only Sh is deployed.

Note that when using relayed Diameter interfaces, the relay function implemented in the Execution Server is internal only to a single Application Server node; that is, the Execution Server cannot relay Rf/Ro/Sh traffic from other Application Server nodes. Thus, the relay is not a formal Diameter relay or proxy agent as defined in *RFC 3588* [3].

## Acronyms and Abbreviations

This section lists the acronyms and abbreviations found in this document. The acronyms and abbreviations are listed in alphabetical order along with their meanings.

| | |
|---|---|
| ABNF | Augmented Backus-Naur Format |
| ACR | Accounting Request |
| AS | Application Server |
| AVP | Attribute-value Pair |
| BCCT | BroadWorks Common Communications Transport |
| BW | BroadWorks |
| CCF | Charging Collection Function |
| CCR | Credit Control Request |
| CDF | Charging Data Function |
| CDR | Call Detail Record |
| CEA | Capabilities-Exchange-Answer |
| CER | Capabilities-Exchange-Request |
| CLI | Command Line Interface |
| DN | Directory Number |
| DNS | Domain Name System |
| EMS | Element Management System |
| FQDN | Fully Qualified Domain Name |
| HSS | Home Subscriber Server |
| HTTP | Hypertext Transfer Protocol |
| IMS | IP Multimedia Subsystem |
| IP | Internet Protocol |
| OCF | Online Charging Function |
| OCI | Open Client Interface |
| OCI-P | Open Client Interface-Provisioning |
| P-CFA | P-Charging-Function-Addresses |
| PS | Provisioning Server |
| S-CSCF | Serving Call Session Control Function |
| SIP | Session Initiation Protocol |
| SLF | Subscriber Location Function |
| SNMP | Simple Network Management Protocol |
| SRV | Service Locator |
| TCP | Transmission Control Protocol |

| | |
|---|---|
| UDP | User Datagram Protocol |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| XS | Execution Server |

## References

[1] BroadSoft, Inc. 2009. *BroadWorks IMS Public Identities Feature Description, Release 14.0.* Available from BroadSoft at xchange.broadsoft.com.

[2] BroadSoft, Inc. 2007. *Sh Interface Using Condor Feature Description, Release 14.sp4*. Available from BroadSoft at xchange.broadsoft.com.

[3] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and Arkko, J., "Diameter Base Protocol", RFC 3588, Internet Engineering Task Force, September 2003. Available from http://www.ietf.org/.

[4] Hakala, H., Mattila, L., Koskinen, J-P., Stura, M., Loughney, J., "Diameter Credit-Control Application", RFC 4006, Internet Engineering Task Force, August 2005. Available from http://www.ietf.org/.

[5] BroadSoft, Inc. 2018. *BroadWorks Accounting Call Detail Record Interface Specification, Release 23.0.* Available from BroadSoft at xchange.broadsoft.com.

[6] BroadSoft, Inc. 2018. *BroadWorks Rf/Ro Interface Specification, Release 23.0.* Available from BroadSoft at xchange.broadsoft.com.

[7] BroadSoft, Inc. 2018. *BroadWorks Sh Interface Specification, Release 23.0.* Available from BroadSoft at xchange.broadsoft.com.

[8] BroadSoft, Inc. 2009. *Advice of Charge Feature Description, Release 16.0*. Available from BroadSoft at xchange.broadsoft.com.

[9] BroadSoft, Inc. 2009. *IMS Accounting Ro Interface Feature Description, Release 16.0*. Available from BroadSoft at xchange.broadsoft.com.

[10] BroadSoft, Inc. 2010. *BroadWorks Diameter Stack Feature Description, Release 17.0*. Available from BroadSoft at xchange.broadsoft.com.

[11] BroadSoft, Inc. 2018. *Support Government Emergency Telecommunications Service (GETS) Feature Description, Release 23.0.* Available from BroadSoft at xchange.broadsoft.com.