Sun GlassFish Communications Server Diameter Adapter 1.0 Installation and Quick Start Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 821–1268–10
December 2009

091223@23031

Contents

Preface

This guide provides information about the Diameter configuration in the Sun GlassFish™ Communications Server configuration file, `domain.xml`.

This preface contains information about and conventions for the entire Sun GlassFish Communications Server documentation set.

Communications Server Documentation Set

The Uniform Resource Locator (URL) for Communications Server documentation is `http://docs.sun.com/coll/1343.10`. For an introduction to Communications Server, refer to the books in the order in which they are listed in the following table.

TABLE P–1 Books in the Communications Server Documentation Set

Book Title	Description
Documentation Center	Communications Server documentation topics organized by task and subject.
Release Notes	Late-breaking information about the software and the documentation. Includes a comprehensive, table-based summary of the supported hardware, operating system, Java™ Development Kit (JDK™), and database drivers.
Quick Start Guide	How to get started with the Communications Server product.
Installation Guide	Installing the software and its components.
Application Deployment Guide	Deployment of applications and application components to the Communications Server. Includes information about deployment descriptors.
Developer's Guide	Creating and implementing Java Platform, Enterprise Edition (Java EE platform) applications intended to run on the Communications Server that follow the open Java standards model for Java EE components and APIs. Includes information about developer tools, security, debugging, and creating lifecycle modules.
Java EE 5 Tutorial	Using Java EE 5 platform technologies and APIs to develop Java EE applications.

TABLE P–1 Books in the Communications Server Documentation Set *(Continued)*

Book Title	Description
Java WSIT Tutorial	Developing web applications using the Web Service Interoperability Technologies (WSIT). Describes how, when, and why to use the WSIT technologies and the features and options that each technology supports.
Administration Guide	System administration for the Communications Server, including configuration, monitoring, security, resource management, and web services management.
High Availability Administration Guide	Setting up clusters, working with node agents, and using load balancers.
Administration Reference	Editing the Communications Server configuration file, `domain.xml`.
Performance Tuning Guide	Tuning the Communications Server to improve performance.
Reference Manual	Utility commands available with the Communications Server; written in man page style. Includes the `asadmin` command line interface.

Related Documentation

For documentation about other stand-alone Sun GlassFish server products, go to the following:

- Message Queue documentation (`http://docs.sun.com/coll/1343.4`)
- Identity Server documentation (`http://docs.sun.com/app/docs/prod/ident.mgmt#hic`)
- Directory Server documentation (`http://docs.sun.com/coll/1224.1`)
- Web Server documentation (`http://docs.sun.com/coll/1308.3`)

A Javadoc™ tool reference for packages provided with the Communications Server is located at `http://glassfish.dev.java.net/nonav/javaee5/api/index.html`. Additionally, the following resources might be useful:

- The Java EE 5 Specifications (`http://java.sun.com/javaee/5/javatech.html`)
- The Java EE Blueprints (`http://java.sun.com/reference/blueprints/index.html`)

For information on creating enterprise applications in the NetBeans™ Integrated Development Environment (IDE), see `http://www.netbeans.org/kb/55/index.html`.

For information about the Java DB database included with the Communications Server, see `http://developers.sun.com/javadb/`.

The GlassFish Samples project is a collection of sample applications that demonstrate a broad range of Java EE technologies. The GlassFish Samples are bundled with the Java EE Software Development Kit (SDK), and are also available from the GlassFish Samples project page at `https://glassfish-samples.dev.java.net/`.

Default Paths and File Names

The following table describes the default paths and file names that are used in this book.

TABLE P–2 Default Paths and File Names

Placeholder	Description	Default Value
as-install	Represents the base installation directory for Communications Server.	Solaris™ and Linux installations, non-root user: *user's-home-directory*`/SUNWappserver` Solaris and Linux installations, root user: `/opt/SUNWappserver` Windows, all installations: *SystemDrive*`:\Sun\AppServer`
domain-root-dir	Represents the directory containing all domains.	All installations: *as-install*`/domains/`
domain-dir	Represents the directory for a domain. In configuration files, you might see *domain-dir* represented as follows: `${com.sun.aas.instanceRoot}`	*domain-root-dir*`/`*domain-dir*
instance-dir	Represents the directory for a server instance.	*domain-dir*`/`*instance-dir*
samples-dir	Represents the directory containing sample applications.	*as-install*`/samples`
docs-dir	Represents the directory containing documentation.	*as-install*`/docs`

Typographic Conventions

The following table describes the typographic changes that are used in this book.

TABLE P–3 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

TABLE P–3 Typographic Conventions *(Continued)*

Typeface	Meaning	Example
AaBbCc123	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	A placeholder to be replaced with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized (note that some emphasized items appear bold online)	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file.

Symbol Conventions

The following table explains symbols that might be used in this book.

TABLE P–4 Symbol Conventions

Symbol	Description	Example	Meaning
`[ ]`	Contains optional arguments and command options.	`ls [-l]`	The `-l` option is not required.
`{ | }`	Contains a set of choices for a required command option.	`-d {y|n}`	The `-d` option requires that you use either the `y` argument or the `n` argument.
`${ }`	Indicates a variable reference.	`${com.sun.javaRoot}`	References the value of the `com.sun.javaRoot` variable.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
→	Indicates menu item selection in a graphical user interface.	File → New → Templates	From the File menu, choose New. From the New submenu, choose Templates.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (http://www.sun.com/documentation/)
- Support (http://www.sun.com/support/)
- Training (http://www.sun.com/training/)

Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions.

To share your comments, go to http://docs.sun.com and click Feedback. In the online form, provide the document title and part number. The part number is a seven-digit or nine-digit number that can be found on the title page of the book or at the top of the document.

◆ ◆ ◆ CHAPTER 1

Installing the Diameter Adapter

Diameter is a networking protocol for authentication, authorization, and accounting (AAA). Many internet providers use AAA before allowing users to connect to their networks. Sun GlassFish™ Communications Server Diameter Adapter 1.0 integrates the Diameter stack into Communications Server.

Diameter Adapter is available to Sun customers as a patch on SunSolve. A Diameter installer and a README file are included in the patch.

This chapter provides instructions for installing Diameter Adapter to an existing installation of Communications Server. The following topics are addressed:

For information on requirements, support, and known issues, see *Sun GlassFish Communications Server Diameter Adapter 1.0 Release Notes*.

Installing Diameter Adapter on a Single Instance

This section provides instructions for installing Diameter Adapter on a single instance of Communications Server.

▼ To Install Diameter Adapter on a Communications Server Instance

Before You Begin Communications Server must be installed before attempting to install the Diameter Adapter software.

1 **Download the Diameter Patch from SunSolve.**

You must be a registered Sun customer to access SunSolve.

2. **Extract the `.zip` file to a location of your choice.**
   The `.zip` file includes an installer JAR file and a README file.

3. **Navigate to the `bin` directory of your Communications Server installation.**
   `cd` *as-install*`/bin`

4. **Stop the Communications Server domain if it is running.**
   `asadmin>` **`stop-domain`** *domain_name*

5. **Run the Diameter Adapter installer.**
   `asadmin>` **`install-addon`** *path_to_installer_jar*
   The Licensing page is displayed.

6. **Accept the license to start the installation.**
   When installation is complete, a message similar to the following is displayed:

```
Installation of the addon SunDiameter completed.
CLI207 The installation of the addon /home/user1234/
SunDiameter_installer.jar completed.
```

7. **Start the domain.**
   `asadmin>` **`start-domain`** *domain_name*

Next Steps For configuration instructions, proceed to "Configuring Diameter for Communications Server" on page 17.

Installing Diameter Adapter on Multiple Instances

This section provides instructions for installing Diameter Adapter on multiple instances of Communications Server.

▼ To Install Diameter Adapter on Multiple Communications Server Instances

Before You Begin If you are using an existing domain, that domain must support clustering.

1. **Create the domain if not already created.**
   You can use the `create-domain(1)` command. Be sure the domain supports clustering.

2. **Download the Diameter Patch from SunSolve.**
   You must be a registered Sun customer to access SunSolve.

3. **Extract the `.zip` file to a location of your choice.**

   The `.zip` file includes an installer JAR file and a README file.

4. **Navigate to the `bin` directory of your Communications Server installation.**

   `cd` *as-install*`/bin`

5. **Stop the Communications Server domain if it is running.**

   ```
   asadmin> stop-domain domain_name
   ```

6. **Run the Diameter Adapter installer.**

   ```
   asadmin> install-addon path_to_installer_jar
   ```

   The Licensing page is displayed.

7. **Accept the license to start the installation.**

   When installation is complete, a message similar to the following is displayed:

   ```
   Installation of the addon SunDiameter completed.
   CLI207 The installation of the addon /home/user1234/
   SunDiameter_installer.jar completed.
   ```

8. **Start the domain.**

   ```
   asadmin> start-domain domain_name
   ```

9. **Start the database.**

   ```
   asadmin> start-database
   ```

10. **Create new instances.**

    The new instances can be located on the same machine as the DAS or on different machines.

11. **To apply the Diameter service to a cluster, do the following:**

    a. **If not already created, create a cluster by using the `create-cluster`(1) command.**

       For example:

       ```
       asadmin> create-cluster diameter-cluster
       Command create-cluster executed successfully.
       ```

    b. **Create the Diameter service for that cluster by using the `create-diameter-service`(1) command.**

       For example:

       ```
       asadmin> create-diameter-service diameter-cluster
       Command create-diameter-service executed successfully.
       ```

c. Create instances for the cluster by using the `create-instance(1)` command.

Because the Diameter service is enabled at the cluster config level, instances automatically inherit the Diameter service feature. However, some custom properties (required for Diameter stack initialization) pertaining to the instance host address must be created explicitly while creating the instances. The *hostname* can be specified in IP address format as well.

Specify `DIAMETER_TCP_ADDRESS` and `DIAMETER_SSL_ADDRESS` as the *instance_hostname* by using the `--systemproperties` option.

For example:

```
asadmin> create-instance --nodeagent nodeagent_name --cluster cluster_name
--systemproperties DIAMETER_TCP_ADDRESS=instance_hostname
:DIAMETER_SSL_ADDRESS=instance_hostname instance_name
```

Note – If you did not specify the system properties during instance creation, use the following command:

```
asadmin> create-system-properties --target instance_name
DIAMETER_TCP_ADDRESS=instance hostname:DIAMETER_SSL_ADDRESS=instance hostname
```

---

d. Restart the cluster.

Use the `stop-cluster(1)` and `start-cluster(1)` commands.

Next Steps For configuration instructions, proceed to "Configuring Diameter for Communications Server" on page 17.

◆ ◆ ◆ CHAPTER 2

Getting Started Using Diameter

The Diameter stack configuration is provided through the Diameter service element in the Sun GlassFish™ Communications Server `domain.xml` file. The Admin Console and `asadmin` Diameter commands can be used to administer the various configuration parameters in the Diameter service.

This chapter describes how to configure and get started with Diameter. The following topics are addressed:

About Diameter in the Communications Server Environment

The *Diameter service* consists of connection information for an individual Diameter peer. A *peer configuration* contains details of a listener, peers, and a set of one or more Diameter applications.

A *Diameter listener* is a local Diameter peer which defines details such as the following:

- Vendor ID
- Product name
- Firmware version
- Mode, such as, client, server, relay, or proxy
- Address/port/transport of the local peer (`bindto`)
- Flag indicating if unknown peer connections will be accepted or not (`accept-unknown-peer`)
- Time interval for retrying peer connections (`tctimer`)
- Realm name used by the local peer

A peer configuration can also include a list of known peers. Each peer points to the application, including the host/port/transport of the peer, and also to the realm. A peer configuration includes a list of Diameter applications to run.

Applications can be either authentication or accounting applications supported by a particular vendor.

Other configurables of the Diameter service are `max-threads`, which denotes the maximum number of threads in a Grizzly pool configuration, `msg-buffer-size`, which indicates the size of the byte buffers in the pool used to store and parse incoming messages, and `enable-msg-persistence` for enabling and disabling message persistence.

Diameter Tools

After installing Diameter Adapter 1.0, you must configure Diameter for Sun GlassFish Communications Server before you can use the Diameter tools.

The following topics are addressed here:

Admin Console

The Communications Server Admin Console is a browser-based utility that features a graphical interface for performing administrative tasks. The format for starting the Administration Console in a web browser is `http://`*hostname:port*. For example:

```
http://localhost:4848
```

After you create the Diameter service, the Diameter node appears in the Admin Console. You can then use the Admin Console pages to perform Diameter administrative tasks. Navigate to the Diameter pages as follows: Configurations→*config-name*→Diameter Service. You can display the help text for a page in the Admin Console by clicking the Help button on the page.

Command-line Interface

Alternatively, you can use the `asadmin` commands to configure and administer the Diameter service from the command line. You can run the `asadmin` commands either from a command prompt or from a script. The format for starting the `asadmin` utility on the command line is *as-install*`/bin/asadmin` *command --option*. A multimode example:

```
asadmin> monitor --type httplistener
```

For a list of all the `asadmin` commands for Diameter, see Appendix A, "Diameter Commands for the asadmin Utility."

You can display help information for a command by typing `help` followed by the command. For example:

```
asadmin> help create-diameter-application
```

For a collection of the help pages for all Communications Server `asadmin` commands, see *Sun GlassFish Communications Server 2.0 Reference Manual*.

Code Generation Tool

The Diameter Adapter provides a tool for developers to add or edit attribute value pairs (AVPs) that are available in the current API. The tool can be used to generate Enums, grouped AVP interfaces, implementation classes, and factory methods to simplify application code development. For information and instructions on using this tool, see Chapter 3, "Generating Diameter Code for Communications Server."

Configuring Diameter for Communications Server

After installation, you are ready to configure the Diameter service. This section provides instructions for configuring the Diameter service by using the Communications Server `asadmin` Diameter commands.

The following topics are addressed here:

▼ To Configure the Initial Diameter Service

To create the Diameter service, you must use the command line. If you do not specify the *target* operand, the default instance is used. If the task applies to a cluster, *target* must specify the cluster name.

After you create the Diameter service by using the `create-diameter-service` command, the Diameter node appears in the Communications Server Admin Console.

Before You Begin The Diameter Adapter must be installed before Diameter can be configured.

1. **Create the Diameter service by using the `create-diameter-service(1)` command.**

   ```
   asadmin> create-diameter-service target-name
   ```

   Default *target-name* is `server`.

2. **Repeat Step 1 for each additional configuration.**

   This is a one-time task per configuration.

3. **Create a Diameter application by using the `create-diameter-application(1)` command.**
   For additional usage information, see "To Create a Diameter Application" on page 21.

4. **Create a Diameter peer by using the `create-diameter-peer(1)` command.**
   For additional usage information, see "To Create a Diameter Peer" on page 25.

5. **Restart the server for changes to take effect.**
   Use the `stop-domain(1)` and `start-domain(1)` commands.

▼ To Create a Diameter Resource

After configuring the initial Diameter service, you are ready to create a Diameter resource. This section provides instructions for creating a resource for Diameter by using the Communications Server Admin Console.

If you want to work from the command line, you can use the `create-admin-object(1)` command. Be sure to set the required properties as described in this procedure. For an example command, see Example 2–1.

Before You Begin The Diameter service, a Diameter application, and a Diameter peer must be created before a Diameter resource can be created.

1. **Open the Admin Console.**
2. **Navigate to Resources→Connectors→Admin Object Resource.**
3. **Click New.**
4. **Select the `sundiameter` resource adapter.**
5. **Select one of the following connection definitions:**
   - For Sh Application
     `com.sun.diameter.application.sh.api.UserProfileServer`
   - For Online Charging Application
     `com.sun.diameter.application.credit.ro.api.RoChargingFactory`
   - For Offline Charging Application
     `com.sun.diameter.application.credit.rf.api.RfChargingFactory`
6. **Enter a JNDI name of your choice and click Next.**
   The Additional Properties section displays the following Diameter peer properties:

- PeerId — The ID of the peer
- Host — Host name or IP address of the peer
- Port — Port number of the peer
- Realm — Peer realm
- SecondaryPeerId — ID of peer to be used when the primary peer is not available

7 **Set PeerId or Host-Port-Realm (others are optional).**

The Host-Port-Realm combination is used to identify unknown peers that are accepted by the Diameter stack.

8 **Configure the `PeerId` property.**

Specify the peer ID used when creating the peer, such as PeerId: `peer_1`

9 **Click Finish.**

Example 2–1 Creating a Diameter Resource on the Command Line

This example creates a Diameter administered object with JNDI name `rf_so`.

```
asadmin> create-admin-object --raname sundiameter --target server
--property PeerId=charging-peer --restype com.sun.diameter.application.credit.
rf.api.RfChargingFactory diameter/rf_ao
Command create-admin-object executed successfully.
```

Using the Diameter Commands

This section provides instructions for using the Diameter `asadmin` commands. The following topics are addressed:

- "Administering the Diameter Service" on page 19
- "Administering Diameter Applications" on page 21
- "Administering Diameter Peers" on page 25

These tasks can also be accomplished by using the Admin Console, with the exception of creating or deleting the Diameter service which can only be done on the command line. Instructions for using the Admin Console are provided in the online help.

Administering the Diameter Service

The Diameter service itself can only be administered on the command line. You cannot create or delete the Diameter service by using the Admin Console. However, all the other Diameter tasks can be performed either on the command line or in the Admin Console.

The following topics are addressed here:

To Create the Diameter Service

Use the `create-diameter-service` command in remote mode to create the Diameter service on an instance or cluster. The default Diameter stack configuration is applied. If you do not specify the *target* operand, the default instance is used. If the task applies to a cluster, *target* must specify the cluster name.

The Diameter service can only be created from the command line. After you create the Diameter service by using the `create-diameter-service` command, the Diameter node appears in the Communications Server Admin Console. Then you can use the Admin Console for other Diameter tasks if you prefer.

Before You Begin The Diameter Adapter must be installed before the Diameter service can be created.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Create the Diameter service by using the `create-diameter-service`(1) command.**

Example 2–2 Creating the Diameter Service on a Server Instance

This example creates the Diameter service on the default instance.

```
asadmin> create-diameter-service
Command create-diameter-service executed successfully.
```

See Also You can also view the full syntax and options of the command by typing `asadmin help create-diameter-service` at the command line.

To Delete the Diameter Service

Use the `delete-diameter-service` command in remote mode to delete the Diameter service configuration on a cluster or server instance. The Diameter service can only be deleted from the command line. This command disables the Diameter service and removes the Diameter node in the Admin Console until you recreate the Diameter service. As child elements of the Diameter service, all Diameter configuration elements will be deleted.

If you do not specify the `--target` option, the default instance is used. If the task applies to a cluster, the `--target` option must specify the cluster name.

1. **Ensure that the server is running.**
   Remote commands require a running server.
2. **Delete the Diameter service by using the `delete-diameter-service`(1) command.**

Example 2–3 Deleting the Diameter Service

This example disables the Diameter service on the default instance.

```
asadmin> delete-diameter-service
Command delete-diameter-service executed successfully.
```

See Also You can also view the full syntax and options of the command by typing `asadmin help delete-diameter-service` at the command line.

Administering Diameter Applications

The following topics are addressed here:

▼ To Create a Diameter Application

Use the `create-diameter-application` command in remote mode to create a vendor-specific Diameter application. If you do not specify the `--target` option, the default instance is used. If the task applies to a cluster, the `--target` option must specify the cluster name.

Before You Begin The Diameter service must be created before a Diameter application can be created.

1. **Ensure that the server is running.**
   Remote commands require a running server.
2. **Create a Diameter application by using the `create-diameter-application`(1) command.**
   Information about the options and properties of the command is included in this help page.

Example 2–4 Creating a Diameter Application

This example creates a vendor-specific Diameter accounting application that is added to the application set referenced by the ID `apps_1`. The vendor ID of this application is 10415.

```
asadmin> create-diameter-application --applicationsid apps_1 --acctappid 300 10415
Command create-diameter-application executed successfully.
```

See Also You can also view the full syntax and options of the command by typing `asadmin help create-diameter-application` at the command line.

▼ To Delete a Diameter Application

Use the `delete-diameter-application` command in remote mode to remove a Diameter application for a specific vendor. If you do not specify the `--target` option, the default instance is used. If the task applies to a cluster, the `--target` option must specify the cluster name.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **List the Diameter applications in a by using the `list-diameter-applications`(1) command.**

3. **Delete a Diameter application by using the `delete-diameter-application`(1) command.**

Example 2–5 Deleting a Diameter Application

```
asadmin> delete-diameter-application --applicationsid apps_2 2000
Command delete-diameter-application executed successfully.
```

See Also You can also view the full syntax and options of the command by typing `asadmin help delete-diameter-application` at the command line.

▼ To Delete a Diameter Application Set

An *application set* is a place holder that is used to group a set of applications that have different vendor IDs. This helps peers to refer to applications easily. Use the `delete-diameter-applications` command in remote mode to remove a Diameter application set. If you do not specify the `--target` option, the default instance is used. If the task applies to a cluster, the `--target` option must specify the cluster name.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Delete a Diameter application set by using the `delete-diameter-applications`(1) command.**

Example 2–6 Deleting a Diameter Application Set

This examples deletes the application set named `apps_2`.

```
asadmin> delete-diameter-applications apps_2
Command delete-diameter-applications executed successfully.
```

See Also You can also view the full syntax and options of the command by typing `asadmin help delete-diameter-applications` at the command line.

▼ To Delete Diameter Accounting Application

Use the `delete-diameter-acctapp` command in remote mode to delete a Diameter accounting application. If you do not specify the `--target` option, the default instance is used. If the task applies to a cluster, the `--target` option must specify the cluster name.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Delete a Diameter accounting application by using the `delete-diameter-acctapp(1)` command.**

3. **Restart the server for changes to take effect.**

   Use the `stop-domain(1)` and `start-domain(1)` commands.

Example 2–7 Deleting a Diameter Accounting Application

This example deletes the accounting application 300.

```
asadmin> delete-diameter-acctapp --applicationsid apps_1 --vendorid 10415 300
Command delete-diameter-acctapp executed successfully
```

See Also You can also view the full syntax and options of the command by typing `asadmin help delete-diameter-acctapp` at the command line.

▼ To Delete a Diameter Authentication Application

Use the `delete-diameter-authapp` command in remote mode to delete a Diameter authentication application. If you do not specify the `--target` option, the default instance is used. If the task applies to a cluster, the `--target` option must specify the cluster name.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Delete a Diameter authentication application by using the `delete-diameter-authapp`(1) command.**

3. **Restart the server for changes to take effect.**

   Use the `stop-domain`(1) and `start-domain`(1) commands.

Example 2–8 Deleting an Authentication Application

This example deletes authentication application 200.

```
asadmin> delete-diameter-authapp --applicationsid apps_1 --vendorid 10415 200
Command delete-diameter-authapp executed successfully.
```

See Also You can also view the full syntax and options of the command by typing `asadmin help delete-diameter-authapp` at the command line.

▼ To List Vendors for a Diameter Application Set

Use the `list-diameter-application` command in remote mode to list the vendor IDs for a specific Diameter application set. If you do not specify the `--target` option, the default instance is used. If the task applies to a cluster, the `--target` option must specify the cluster name.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **List vendor IDs for a Diameter application set by using the `list-diameter-application`(1) command.**

Example 2–9 Listing Vendor IDs for a Diameter Application Set

This example lists the vendor IDs for application set `apps_1`.

```
asadmin> list-diameter-application apps_1
10415
Command create-diameter-service executed successfully.
```

See Also You can also view the full syntax and options of the command by typing `asadmin help list-diameter-application` at the command line.

▼ To List Diameter Applications

Use the `list-diameter-applications` command in remote mode to list the Diameter applications in a specific cluster or server instance. If you do not specify the `--target` option, the default instance is used. If the task applies to a cluster, the `--target` option must specify the cluster name.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **List the Diameter applications in a particular server or cluster by using the `list-diameter-applications`(1) command.**

Example 2–10 Listing Diameter Applications

This example list the Diameter applications on the default instance.

```
asadmin> list-diameter-applications
apps_1
Command list-diameter-applications executed successfully.
```

See Also You can also view the full syntax and options of the command by typing `asadmin help list-diameter-applications` at the command line.

Administering Diameter Peers

The following topics are addressed here:

▼ To Create a Diameter Peer

Use the `create-diameter-peer` command in remote mode to add peer host information to the known peer list in the specified cluster or server instance. If you do not specify the `--target` option, the default instance is used. If the task applies to a cluster, the `--target` option must specify the cluster name.

Before You Begin A Diameter application must be created before a Diameter peer can be created.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Add peer host information to the Diameter peer list by using the `create-diameter-peer`(1) command.**

   Information about the options and properties of the command is included in this help page.

Example 2–11 Adding Peer Information to a Diameter Application Set

This example creates a peer named `peer_1` on the default instance.

```
asadmin> create-diameter-peer --applicationsref apps_1 --realm Sun.com --peerhost nyx
--peerport 3878 --transport tls peer_1
Command create-diameter-peer executed successfully.
```

See Also You can also view the full syntax and options of the command by typing `asadmin help create-diameter-peer` at the command line.

▼ To Update a Diameter Peer

Use the `set` command to update an existing Diameter peer.

1. **List Diameter peers by using the `list-diameter-peers`(1) command.**
2. **Update a Diameter peer by using the `set`(1) command.**

   This example changes the listening port of the peer with ID `p1`.

   ```
   asadmin> set server-config.diameter-service.peer-configuration.peers.peer.p1.port=3737
   server-config.diameter-service.peer-configuration.peers.peer.p1.port = 3737
   ```

   where `p1` is the peer ID which can be changed.
3. **Restart the server for changes to take effect.**

   Use the `stop-domain`(1) and `start-domain`(1) commands.

▼ To Delete a Diameter Peer

Use the `delete-diameter-peer` command in remote mode to remove a peer in the Diameter service from the specified cluster or server instance. If you do not specify the `--target` option, the default instance is used. If the task applies to a cluster, the `--target` option must specify the cluster name.

1. **Ensure that the server is running.**

   Remote commands require a running server.
2. **List Diameter peers by using the `list-diameter-peers`(1) command.**
3. **Delete a Diameter peer by using the `delete-diameter-peer`(1) command.**

Example 2–12 Deleting a Diameter Peer

This example deletes `peer_1` from the default instance.

```
asadmin> delete-diameter-peer peer_1
Command delete-diameter-service executed successfully.
```

See Also You can also view the full syntax and options of the command by typing `asadmin help delete-diameter-peer` at the command line.

▼ To List Diameter Peers

Use the `list-diameter-peers` command in remote mode to list all peers on the specified cluster or server instance. If you do not specify the `--target` option, the default instance is used. If the task applies to a cluster, the `--target` option must specify the cluster name.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **List Diameter peers by using the `list-diameter-peers`(1) command.**

Example 2–13 Listing Diameter Peers

This example lists the Diameter peers on the default instance.

```
asadmin> list-diameter-peers
peer_1
Command list-diameter-peers executed successfully.
```

See Also You can also view the full syntax and options of the command by typing `asadmin help list-diameter-peers` at the command line.

◆ ◆ ◆ CHAPTER 3

Generating Diameter Code for Communications Server

The attribute value pair (AVP) code generation tool is based on data present in the `dictionary.xml` file. The schema for the Diameter dictionary is defined by the `dictionary.xsd` file. All files are available under the *as-install*`/addons/sundiameter/tools` directory.

Generating Diameter Code

For in-depth understanding, refer to the schema for the `dictionary.xsd` file. The schema defines the following elements:

- **Application.** Identifies the application being supported by the Diameter stack. Includes a list of messages and supported AVPs.
- **AVP.** Holds information that applies to a particular AVP, such as, datatype, vendor information, document reference.
- **Message.** Defines the messages supported by the application, and lists the supported AVPs .

▼ To Use the AVP Code Generation Tool

Input to the code generation tool is done by editing the `dictionary.xml` file (recommended) or the interfaces that are decorated with required annotations. After editing the `dictinary.xml` file, you must validate `dictionary.xml` against `dictionary.xsd`.

You can directly write the AVP interface decorated with appropriate annotations (example below) and use the tool to generate implementation class.

In the following examples, you can specify AVP attributes such as `protected`, `vendor-bit`, `mandatory`, `may-encrypt`, or `vendor-id` of choice. ID and code value should be the same, and unique for each AVP in the `dictionary.xml` file. The `doc-ref-ID` always points to a value such as `code` or `code:vendor-id` (if `vendor-id` is present).

Before You Begin The Diameter service must be created before the Diameter code generation tool can be used.

1 (Optional) Edit the `dictionary.xml` file to add a new application or edit an existing application.

Note – The application ID should be unique for the `dictionary.xml` file. This step is not mandatory. You can also edit or add AVPs under an existing application.

Example of adding a new application:

```
<application id="0" package-name="com.sun.diameter.base.api">
.... set of AVPs goes here...
</application>
```

2 Edit the `dictionary.xml` file to add new AVPs under a specific application.

Example of adding enumeration AVP:

```
<avp name="My-Enum-AVP" protected="may" vendor-bit="mustnot" code="899"
description="" mandatory="must" may-encrypt="no" ID="899" doc-ref-ID="899">
<type type-name="Enumerated"></type>
<enum name="STATE_MAINTAINED" code="0"></enum>
<enum name="NO_STATE_MAINTAINED" code="1"></enum>
</avp>
```

Example of adding base datatypes AVP:

```
<avp name="My-Unsigned32-AVP" protected="may" vendor-bit="mustnot"
code="900" mandatory="must" may-encrypt="no" ID="900" doc-ref-ID="900">
<type type-name="Unsigned32"></type>
</avp>
```

Example of adding grouped AVP:

```
<avp name="My-Grouped-AVP" protected="may" vendor-bit="mustnot" code="901"
mandatory="must" may-encrypt="no" ID="901" doc-ref-ID="900">
<grouped>
<gavp name="My-Unsigned32-AVP" max="-1"/>
<gavp name="My-OctetString-AVP"/>
</grouped>
</avp>

<avp name="My-Unsigned32-AVP" protected="may" vendor-bit="mustnot" code="900"
mandatory="must" may-encrypt="no" ID="900" doc-ref-ID="900">
<type type-name="Unsigned32"></type>
</avp>

<avp name="My-OctetString-AVP" protected="may" vendor-bit="must" code="902"
mandatory="must" may-encrypt="no" ID="902" vendor-id="10415" doc-ref-ID="902:10415">
```

```
<type type-name="OctetString"></type>
</avp>
```

Note – In the preceding examples, you can specify AVP attributes such as `protected`, `vendor-bit`, `mandatory`, `may-encrypt`, `vendor-id` of your choice. ID and code value should be same and unique for each AVP in the `dictionary.xml` file. The `doc-ref-ID` always points to a value such as `'code'` or `code:vendor-id` (if `vendor-id` is present).

3 Edit the `dictionary.xml` file to add document reference at the end of `dictionary.xml` file.

Example of adding document reference for Java Docs:

```
<docs doc-ID="287">
<![CDATA[
<pre>
My Document for AVP goes here.
</pre>]]>
</docs>
```

4 Generate AVP classes by running the following command:

```
% cd as-install/addons/diametertools
% ../../lib/ant/bin/ant all
```

If you have made changes in the `dictionary.xml` file as mentioned in Step 2, the following files are created:

a. `EnumMyEnumAVP.java`

This is the Enum class for AVP code 899 under given package name (*as-install*`/addons/sundiameter/tools/generated/src/`*package-name*).

Code snapshot:

```
package com.sun.diameter.application.credit.api;

            import com.sun.diameter.annotation.AVPInfo;

            @AVPInfo(code = 899)
            public enum EnumMyEnumAVP {

                STATE_MAINTAINED (0),
                NO_STATE_MAINTAINED (1);

                private int value;

                private EnumMyEnumAVP(int value) {
                    this.value = value;
                }
```

```
                public int getValue() {
                    return value;
                }

                public static EnumMyEnumAVP toValue(int mValue) {
                    if (mValue == 0) {
                        return EnumMyEnumAVP.STATE_MAINTAINED;
                    }
                    if (mValue == 1) {
                        return EnumMyEnumAVP.NO_STATE_MAINTAINED;
                    }
                    return null;
                }

            }
```

b. `MyGroupedAVP.java`

This is the Java class for AVP code 901 under given package name (*as-install*`/addons/sundiameter/tools/generated/src/`*package-name*).

Code snapshot:

```
package com.sun.diameter.application.credit.api;

        import java.util.List;
        import com.sun.diameter.annotation.AVPInfo;
        import com.sun.diameter.base.api.AVP;

        @AVPInfo(code = 901, mandatory = "must", mayEncrypt = "no", protect = "may")
        public interface MyGroupedAVP
            extends AVP
        {


            @AVPInfo(code = 900, type = "Unsigned32")
            public List<Long> getMyUnsigned32AVPList();

            @AVPInfo(code = 900, type = "Unsigned32")
            public void setMyUnsigned32AVPList(List<Long> data);

            @AVPInfo(code = 902, vendorId = 10415, type = "OctetString")
            public String getMyOctetStringAVP();

            @AVPInfo(code = 902, vendorId = 10415, type = "OctetString")
            public void setMyOctetStringAVP(String data);

        }
```

c. `MyGroupedAVPImpl.java`

This is the Impl Java class for AVP code 901 under given package name (*as-install*`/addons/sundiameter/tools/generated/src/`*package-name*`/impl`).

Code snapshot:

```
package com.sun.diameter.application.credit.api.impl;

            import java.util.ArrayList;
            import java.util.List;
            import com.sun.diameter.annotation.AVPInfo;
            import com.sun.diameter.application.GeneratedApplicationAVPFactory;
            import com.sun.diameter.application.credit.api.MyGroupedAVP;
            import com.sun.diameter.base.api.AVP;
            import com.sun.diameter.base.datatypes.OctetString;
            import com.sun.diameter.base.datatypes.Unsigned32;
            import com.sun.diameter.base.impl.AVPImpl;
            import com.sun.diameter.base.impl.GroupedAVPImpl;

            @AVPInfo(code = 901)
            public class MyGroupedAVPImpl
                extends GroupedAVPImpl
                implements MyGroupedAVP
            {

                private AVPImpl _t;
                private List<Long> myUnsigned32AVPList = new ArrayList<Long>();
                private String myOctetStringAVP;

                public MyGroupedAVPImpl() {
                    super();
                    this.setCode(901);
                    this.setMandatory(true);
                    this.setVendorSpecific(false);
                    this.setSecured(false);
                }

                public void addAVP(AVP avp) {
                    switch ((avp.getCode())) {
                        case (900):
                            myUnsigned32AVPList.add(((Unsigned32) avp.getDataAsType()).getValueByObject());
                            break;
                        case (902):
                            myOctetStringAVP = ((OctetString) avp.getDataAsType()).getValue();
                            break;
                    }
                    super.addAVP(avp);
                }
```

```
    public boolean isGenerated() {
        return (true);
    }

    public List<Long> getMyUnsigned32AVPList() {
        return myUnsigned32AVPList;
    }MyGroupedAVP.java

    public void setMyUnsigned32AVPList(List<Long> value) {
        this.myUnsigned32AVPList=value;
        this.removeAllAVP(900,-1);
        for (Long v: value) {
            AVP avp;
            avp = GeneratedApplicationAVPFactory.createMyUnsigned32AVP(v);
            this.getNestedAVPs().add(avp);
        }
    }

    public String getMyOctetStringAVP() {
        return myOctetStringAVP;
    }

    public void setMyOctetStringAVP(String value) {
        this.myOctetStringAVP=value;
        this.removeAVP(902,10415);
        AVP avp;
        avp = GeneratedApplicationAVPFactory.createMyOctetStringAVP10415(value);
        this.getNestedAVPs().add(avp);
    }

}
```

d. Makes an entry for Base data type AVPs under:

as-install/addons/sundiameter/tools/generated/src/com/sun/diameter/application/GeneratedApplicationAVPFactory.java

Code snapshot:

```
@AVPInfo(code = 900)
            public static AVP createMyUnsigned32AVP(long data) {
                AVPImpl avpImpl = new AVPImpl();
                avpImpl.setData(new Unsigned32(data));
                avpImpl.setCode(900);
                avpImpl.setMandatory(true);
                avpImpl.setVendorSpecific(false);
                avpImpl.setSecured(false);
                return avpImpl;
            }
```

```
@AVPInfo(code = 902, vendorId = 10415)
public static AVP createMyOctetStringAVP10415(String data) {
    AVPImpl avpImppublic static MyGroupedAVP createMyGroupedAVP() {
    return new MyGroupedAVPImpl();
}l = new AVPImpl();
    avpImpl.setData(new OctetString(data));
    avpImpl.setCode(902);
    avpImpl.setMandatory(true);
    avpImpl.setVendorSpecific(true);
    avpImpl.setVendorId(10415);
    avpImpl.setSecured(false);
    return avpImpl;
}

@AVPInfo(code = 899)
public static AVP createMyEnumAVP(EnumMyEnumAVP data) {
    AVPImpl avpImpl = new AVPImpl();
    avpImpl.setData(new Enumerated((data.getValue())));
    avpImpl.setCode(899);
    avpImpl.setMandatory(true);
    avpImpl.setVendorSpecific(false);
    avpImpl.setSecured(false);
    return avpImpl;
}
```

e. Makes an entry for Grouped type AVPs under:

as-install/addons/sundiameter/tools/generated/src/com/sun/diameter/application/ApplicationAVPFactory.java

Code snapshot:

```
public static MyGroupedAVP createMyGroupedAVP() {
                return new MyGroupedAVPImpl();
            }
```

Here, *package-name* points to the attribute value present at application level. For example:

```
<application id="0" package-name="com.sun.diameter.base.api">
```

All files are compiled and available under *as-install*/addons/sundiameter/tools.

5 Use newly generated classes in Diameter.

```
% cd as-install/addons/sundiameter/tools/build/
% jar -cvf jar-name .
% cp jar-name as-install/domains/domain-name/applications/j2ee-modules/sundiameter/
% cd as-install/domains/domain-name/applications/j2ee-modules/sundiameter/
% jar -xvf jar-name
```

Note – For a cluster scenario, you must run this sequence manually on each instance, then restart the cluster. Use the `stop-cluster(1)` and `start-cluster(1)` commands to restart the instances.

6. **Restart the domain.**

   Use the `stop-domain(1)` and `start-domain(1)` commands.

7. **If you have given your own *package-name* at the application level, you need to make the same entry as a JVM Option.**

   a. **Open Admin Console.**

   b. **Go to Application Server -> JVM Settings -> JVM Options**

   c. **Click Add JVM Option and add the following entry:**

      ```
      -Dsun.diameter.avp.packagelist=<package-name>
      ```

   d. **Click Save.**

   e. **Restart the domain.**

      Use the `stop-domain(1)` and `start-domain(1)` commands.

◆ ◆ ◆ APPENDIX A

Diameter Commands for the asadmin Utility

This appendix lists the `asadmin` Diameter commands that apply to this release of the Sun GlassFish™ Communications Server. For a complete list of `asadmin` commands, see *Sun GlassFish Communications Server 2.0 Reference Manual*.

`create-diameter-application(1)`
: Creates a vendor-specific Diameter application. Supported in remote mode only. For usage instructions, see "To Create a Diameter Application" on page 21.

`create-diameter-peer(1)`
: Adds peer host information to specific Diameter applications in the specified cluster or server instance. Supported in remote mode only. For usage instructions, see "To Create a Diameter Peer" on page 25.

`create-diameter-service(1)`
: Creates the Diameter service with the default Diameter stack configuration. Supported in remote mode only. For usage instructions, see "To Create the Diameter Service" on page 20.

`delete-diameter-acctapp(1)`
: Deletes a Diameter accounting application. Supported in remote mode only. For usage instructions, see "To Delete Diameter Accounting Application" on page 23.

`delete-diameter-application(1)`
: Deletes a Diameter application of a specific vendor. Supported in remote mode only. For usage instructions, see "To Delete a Diameter Application Set" on page 22.

`delete-diameter-applications(1)`
: Deletes a Diameter application set. Supported in remote mode only. For usage instructions, see "To List Vendors for a Diameter Application Set" on page 24.

`delete-diameter-authapp(1)`
: Deletes a Diameter authentication application. Supported in remote mode only. For usage instructions, see "To Delete a Diameter Authentication Application" on page 23.

`delete-diameter-peer(1)`
: Deletes a peer in the Diameter service from the specified cluster or server instance. Supported in remote mode only. For usage instructions, see "To Delete a Diameter Peer" on page 26.

`delete-diameter-service(1)`
: Deletes the Diameter service configuration on a cluster or server instance. Supported in remote mode only. For usage instructions, see "To Delete the Diameter Service" on page 20.

`list-diameter-application(1)`
: Lists vendors for a specific Diameter application set. Supported in remote mode only. For usage instructions, see "To List Vendors for a Diameter Application Set" on page 24.

`list-diameter-applications(1)`
: Lists the Diameter applications in a specific cluster or server instance. Supported in remote mode only. For usage instructions, see "To List Diameter Applications" on page 24.

`list-diameter-peers(1)`
: List all peers on the specified cluster or server instance. Supported in remote mode only. For usage instructions, see "To List Diameter Peers" on page 27.

Index