Oracle® Communications Service Broker

Installation Guide

Release 6.0

E23522-02

March 2012

ORACLE®

Oracle Communications Service Broker Installation Guide, Release 6.0

E23522-02

Contents

4 Installing the Software

5 Setting Up a Domain

6 Next Steps

Preface

This document describes how to install Oracle Communications Service Broker. This document is relevant to all Service Broker products.

Audience

This document is intended for system administrators, developers, and system integrators who want to run Service Broker.

This documentation is based on the assumption that you are already familiar with Service Broker concepts. See *Oracle Communications Service Broker Concepts Guide* for more information.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents in the Service Broker Release 6.0 documentation set:

- *Oracle Communications Service Broker Release 6.0 Release Notes*
- *Oracle Communications Service Broker Release 6.0 Concepts Guide*
- *Oracle Communications Service Broker Release 6.0 System Administrator's Guide*
- *Oracle Communications Service Broker Release 6.0 Subscriber Store User's Guide*
- *Oracle Communications Service Broker Release 6.0 Online Mediation Controller Implementation Guide*
- *Oracle Communications Service Broker Release 6.0 Policy Controller Implementation Guide*
- *Oracle Communications Service Broker Release 6.0 VPN Implementation Guide*

- *Oracle Communications Service Broker Release 6.0 SVC Implementation Guide*

Downloading Oracle Communications Documentation

Oracle Communication Service Broker documentation is available from the Oracle software delivery Web site:

http://edelivery.oracle.com/

Additional Oracle Communication documentation is available from Oracle Technology Network:

http://www.oracle.com/technetwork/index.html

1

About Service Broker Deployments

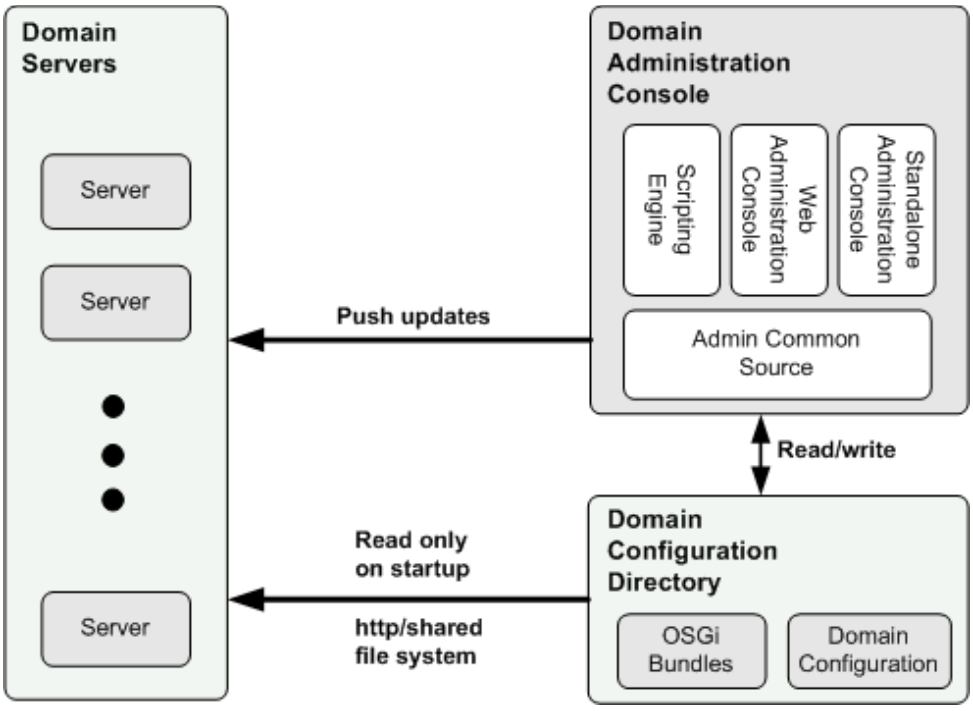
This chapter describes basic concepts of Oracle Communications Service Broker deployments and the different types of Service Broker deployments.

Understanding the Domain-Based Administration Model

Service Broker deployments are implemented and administered using domains. A domain is a logically related group of servers.

Figure 1–1 shows domain components, which are explained in the following sections.

Figure 1–1 A Service Broker Domain



About Domain Servers

Each server runs on its own Java Virtual Machine (JVM), and the term "server" reflects a JVM rather than a physical host.

Servers within a domain are symmetrical, which means that they all have the same OSGi software bundles installed and started.

Servers can be added to and removed from the domain while the system is running, without service interruption.

About the Domain Configuration Directory

A domain has an associated domain configuration, which is stored in a domain configuration directory. The domain configuration directory contains configuration for all servers in the domain and the OSGi software bundles that the servers run.

Individual servers do not store configuration locally except for security related details that they need to access the domain configuration directory.

All servers in the domain have read-only access to the domain configuration directory. When a server starts, it retrieves configuration data from the domain configuration directory and loads it into memory for use at runtime. The domain configuration directory is accessed using a shared file system or via the Domain Configuration Web server.

Domain servers can access the domain directory in one of the following ways:

- Through a shared file system
- Through HTTP

About the Domain Administration Console

The Administration Console provides an interface for the domain configuration. With the Administration Console you manage the domain servers, the OSGi software bundles installed in the domain, and the data stored in the domain configuration directory. The Administration Console gives you read and write access to the domain configuration directory.

When running the Administration Console, it provides a Java Management Extension (JMX) interface for the domain configuration. It also provides an additional interface, whose type depends on Administration Console execution mode.

You can run the Administration Console in the following modes:

- Web access mode (Web Administration Console)

  Starting the Administration Console in Web access mode enables its graphical browser-based user interface. It allows administrators to configure the domain from any computer with a browser and network access to the Web Administration Console server.

- Standalone mode (Standalone Administration Console)

  In standalone mode, the Administration Console runs as a graphical Java client application. You can run the standalone console only on the physical computer on which the Administration Console is installed.

- Scripting mode (Scripting Engine)

  In the scripting mode, the Administration Console runs a scripting engine accepting XML-based scripts. The scripts express JMX operations and attributes that you invoke to change Service Broker configurations.

When running the Administration Console, you access the domain configuration either through the front-end interface enabled by the specific mode chosen, or through the JMX interface in the back-end.

The Administration Console can be installed and run from any host that has access to the domain configuration directory. It can run on the same physical computer hosting

one or more of the servers in the domain, however, Oracle recommends a dedicated physical computer for the Administration Console.

Domain Types

The domain type reflects what kind of software bundles are started on the domain servers and the role that the servers perform.

There are three domain types:

- Signaling domain

  Servers in the signaling domain run the artifacts associated with the Service Broker signaling tier only, that is SSUs, which primarily serves for network connectivity. Servers in the signaling domain are often referred to as signaling servers.

- Processing domain

  Servers in the processing domain run the artifacts associated with the Service Broker processing tier, that is OE, IMs, SMs, applications and mediators, which primarily serve a traffic processing and mediation function. Servers in the processing domain are often referred to as processing servers.

- Unified domain

  This domain combines the processing and signaling tier functions. Servers in the unified domain run the artifacts associated with both the signaling tier and processing tier.

Note: If your signaling domain or unified domain implements SS7 connectivity, then the maximum number of servers in the domain is as follows:

- For SIGTRAN connectivity, the domain consists of up to sixteen servers.
- For TDM connectivity, the domain consists of up to two servers.

Domain Topologies

This section describes the domain topologies that you can use for Service Broker deployments. Domain topologies differ by the level of performance, high availability and service availability that they provide.

Multi-Domain Topology

A multi-domain topology implements two types of domains:

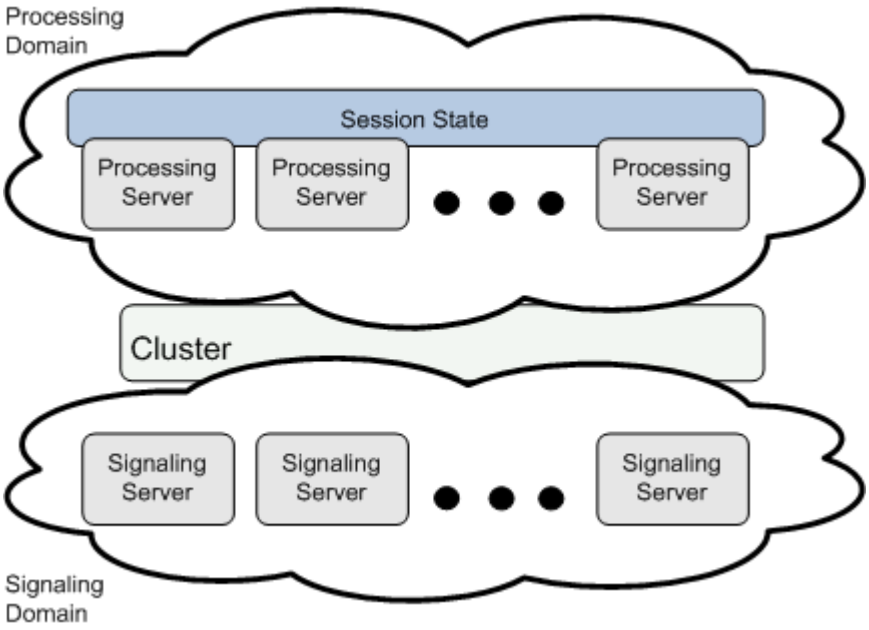
- A signaling domain, running SSUs and serving for network connectivity.
- A processing domain, running stateful components such as IMs, OE, mediators and applications, and serving for traffic processing and mediation function.

The signaling domain includes two or more servers, providing availability, therefore if one server fails, other servers handle the new calls.

Components running on the processing servers are stateful, maintaining and distributing state information across servers in the domain, therefore services are continuously delivered without disruption. See the discussion about service continuity in "Service Mode" for more information.

Figure 1–2 shows a multi-domain topology with one signaling domain and one processing domain, and the session state distributed across the processing servers.

Figure 1–2 A Multi-Domain Topology



The signaling domain and processing domain interact, and propagate protocol events across the tier and domain boundaries.

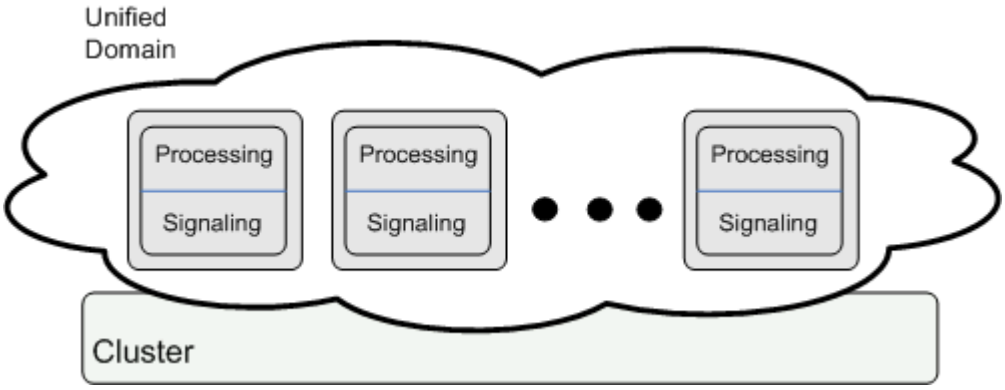
Note: If your system connects to a number of different networks, consider administrating the connectivity to each network separately, by implementing a different signaling domain to connect each of the networks.

Single-Domain Topology

The single-domain topology implements only one domain, a unified domain, with multiple servers, wherein each server performs the role of both the signaling tier and processing tier. Multiple servers in the domain serve for availability.

Figure 1–3 shows a single-domain topology with one unified domain, wherein servers perform the role of both the processing and signaling tiers.

Figure 1–3 A Single-Domain Topology



When implementing a single-domain deployment and creating a unified domain, you need to set the domain's service mode. See "Service Mode" for more information.

Service Mode

The components running in the processing tier are stateful, that is they maintain session information. They retrieve and store session state in an in-memory storage.

When state information is maintained and distributed across the domain servers, service continuity is assured: on server failure, functioning servers continue to retrieve and process all messages, including those stored in the in-memory state of the failed server. Therefore, if a server fails, another server continues to handle existing calls, providing service continuity.

Whereas a processing domain always implements service continuity, in a unified domain you can choose whether to turn service continuity on or off. Turning service continuity off improves performance significantly. When creating a unified domain, select the service mode:

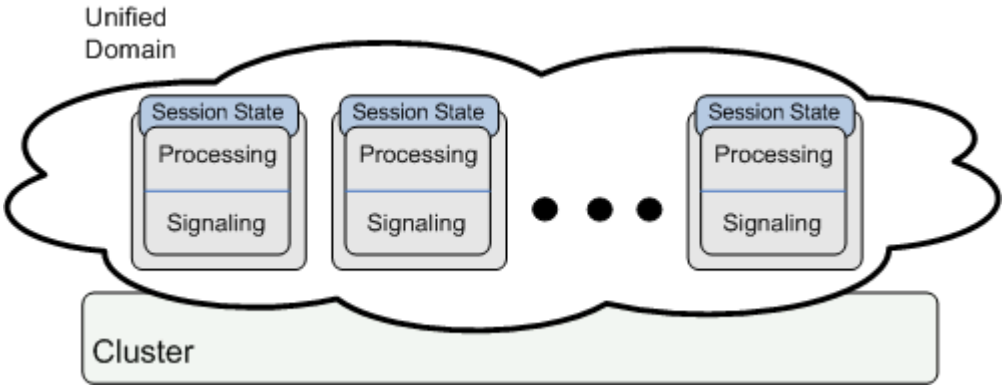
- Service availability: service continuity is turned off
- Service availability and continuity: service continuity is turned on

A multi-domain deployment with separate processing and signaling domains always provides service continuity. A single-domain deployment with one unified domain may be configured to either provide service continuity or not.

About Production Systems

In production systems, for best performance, Oracle recommends using the single-domain topology (described in "Single-Domain Topology") wherein service continuity is turned off, as shown in Figure 1–4. In this deployment, servers maintain state information, however the state information is not distributed across servers.

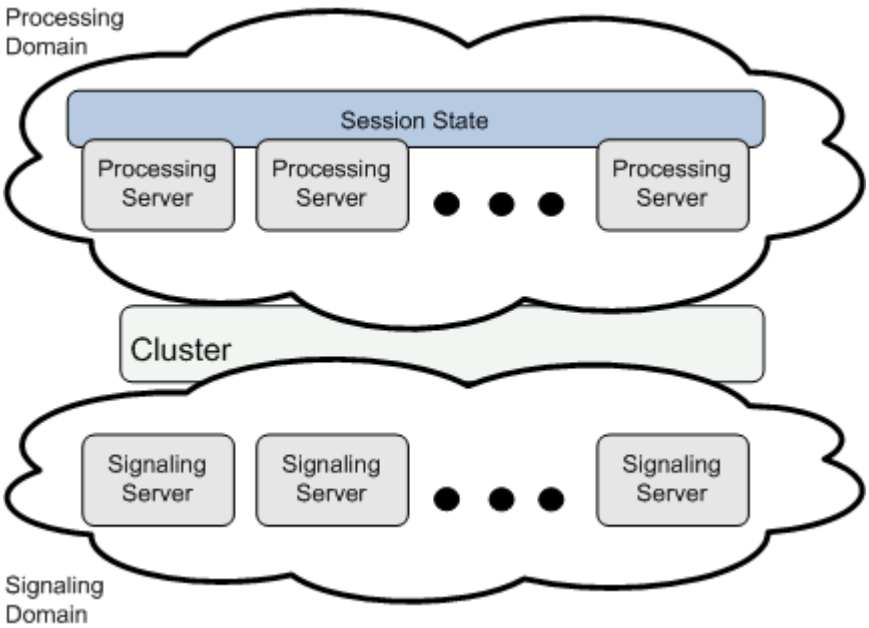
Figure 1–4 A Production Single-Domain Deployment



Another alternative is to implement the multi-domain topology (described in "Multi-Domain Topology"), that implements service continuity out of the box, as shown in Figure 1–5.

In a multi-domain deployment you can create multiple signaling domains (each connecting to a different network), and multiple processing domains, each supporting a different processing functionality. In complex Service Broker deployments, you may find administering different network connections and different functionality in separate domains easier.

Figure 1–5 A Production Multi-Domain Deployment



Note: In a production system, each domain that you install should include at least two servers, for redundancy and availability purposes.

About Test and Evaluation Systems

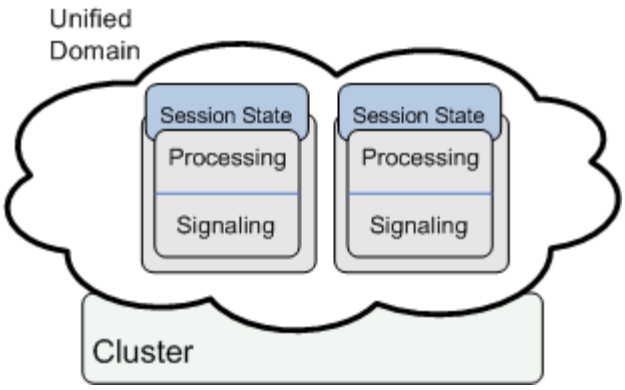
You can use a test and evaluation system to familiarize yourself with Service Broker, for integrations, for proof of concept labs, and for training purposes. A test and evaluation system does not have the same requirements for availability, service continuity and performance that a production system has.

A test and evaluation system is a basic, single-domain deployment. A test and evaluation deployment comprises one unified domain with a single server, or two servers at most, each preforming the role of both signaling tier and processing tier.

WARNING: Never use a test and evaluation system in production.

Figure 1–6 shows a test and evaluation deployment with one domain, wherein one server perform the role of both the processing and signaling tiers. In this figure service continuity is turned off, however, you can also deploy a system with the service continuity turned on.

Figure 1–6 Test and Evaluation Deployment



Scaling the Deployment

Scalability is the ability of a system to provide throughput in proportion to, and limited only by, available hardware resources. A scalable system is one that can handle increasing numbers of requests without adversely affecting response time and throughput.

The growth of computational power within one operating environment is called vertical scaling. Horizontal scaling is leveraging multiple systems to work together on a common problem in parallel.

Service Broker scales both vertically and horizontally. Scaling options differ according to whether you are scaling the processing tier or the signaling tier.

Scaling the Processing Tier

The processing tier of a Service Broker deployment includes two or more servers, employing an N+1 high availability schema, where several processing servers are grouped together to share the workload. The Service Broker processing tier can increase its throughput by adding a new processing servers to the processing domain. You can add a new processing server on either of the following:

- A new physical host computer (horizontal scaling)

- A physical computer that already host another processing servers (vertical scaling)

Scaling the Signaling Tier

As in the processing tier, the signaling tier should include at least two servers, and more servers can be added as needed.

However, there are a few exceptions when implementing connectivity to SS7 networks, as follows:

- When implementing SIGTRAN-based SS7 connectivity, the signaling tier can include up to sixteen servers, whether the servers are administered in a unified domain or in a separate signaling domain. Note that in the case of a unified domain, your processing tier is also limited to the same number of servers.
- When implementing TDM-based SS7 connectivity, the signaling tier can include up to two servers, whether the servers are administered in a unified domain or in a separate signaling domain. Note that in the case of a unified domain, your processing tier is also limited to the same number of servers.

To work around the limitations implied by the SS7 connectivity, Oracle recommends that you deploy a separate signaling domain to implement connectivity to the SS7 network. This way connectivity to other signaling networks is not affected, and servers can be added to the other signaling domain as needed.

To increase the throughput of SS7 connectivity in the case of TDM, you can also add another pair of signaling servers in a separate signaling domain.

Planning a Service Broker Deployment

Before you start an installation, you have to plan your Service Broker deployment.

Planning your deployment includes choosing a domain topology and defining each of the domains. That is the number of servers in the domain and additional domain considerations as described in "Setting Up a Domain".

2

System Requirements

This chapter describes the hardware and software requirements for Oracle Communications Service Broker.

Hardware Requirements

There are different hardware requirements on machines running servers and machines running the Administration Console. See "Servers" and "Administration Console" for details.

Servers

Servers can be deployed on the following CPU architectures:

- x86-64 bit
- Oracle Sun SPARC 64 bit

Each server should have redundant network interface cards.

The following are the recommendations for each physical server in a production system. These are guidelines only:

- Dual quad core processors with 16 GB of RAM
- 2 GB of free disk space for the installation

The following requirements apply to specific server types:

- Processing servers

  Extra disk space is required for processing servers if you store Service Detail Records (SDRs) in files on the server. In general, 10 GB is sufficient for handling SDRs. However, the actual disk space you need depends on the server's capacity, the amount of traffic, and how many SDR files are stored on the server.

  See information about configuring SDR logging in *Oracle Communications Service Broker System Administrator's Guide*.

- Signaling servers

  A Dialogic TDM signaling board and Dialogic software are required for signaling servers that run in a TDM-based network.

  See "Installing Dialogic Hardware and Software for TDM Networks" for more information.

Administration Console

The stand-alone Administration Console and Web Administration Console can be deployed on the following CPU architectures:

- x86-64 bit
- Oracle Sun SPARC 64 bit

The following are the recommendations for each physical computer that hosts an Administration Console. These are guidelines only:

- Quad core processors with 8 GB of RAM
- 300 MB of free disk space for the installation

Software Requirements

Service Broker requires the following software:

- Network Time Protocol (NTP) client

  Service Broker requires that all servers accurately synchronize their system clocks to a common time source, to within one or two milliseconds. Oracle recommends using an NTP client or daemon and synchronizing all Service Broker servers to a common NTP server.

  Caution: Synchronizing system clocks to within one or two milliseconds is critical for proper functioning of the system. If you do not synchronize server clocks, the system may display unexpected behavior.

- Service Broker relies upon a shared library of the underlying operating system to provide Stream Control Transmission Protocol (SCTP) support.

  If you are implementing a solution that uses Diameter over SCTP, you need to ensure that the operating system includes the following library:

  libsctp.so.1

  If needed, install the shared library on the operating system of each managed server host machine.

- The Oracle Universal Installer imposes additional system requirements and library dependencies. For example, it requires 500 MB of swap space to be available on the installation target system.

  For more information about the Oracle Universal Installer, see *Oracle Universal Installer and OPatch User's Guide*.

Supported Configurations

Service Broker can be used with the following operating systems:

- Oracle Enterprise Linux 5.5
- Red Hat Enterprise Linux 5.4 or 5.5
- Oracle Solaris 10

The JDKs are bundled with Service Broker. The Java version used is 1.6.0_30 and JRockit R28.2.0.

The Web Administration Console can be used with the following Web Browsers:

- Microsoft Internet Explorer
- Mozilla Firefox

Table 2–1 lists the supported combinations of CPU type and JVM, along with the name of the installer archive file for the combination.

Table 2–1 Supported Configurations and Corresponding Installation Archive

CPU	JVM	Installer Archive
x86-64 bit	Java JDK 64 bit and JRockit 64 bit	**`ocsb600-linux_x86-64.zip`**
Oracle Sun SPARC 64 bits	Java JDK 64 bit and JRockit 64 bit	**`ocsb600-sparc64.zip`**

3

Pre-Installation Tasks

This chapter describes the tasks you need to perform prior to installing Oracle Communications Service Broker.

Installing Dialogic Hardware and Software for TDM Networks

This task is required only if you intend to run Service Broker in a TDM-based network.

To use Service Broker in a TDM-based network, you must install the following components on any Signaling Server that will run a Service Broker SS7 Signaling Server Unit (SSU):

- Dialogic TDM signaling board (with a license button) and the card driver
- The following Software related to the Dialogic Distributed Signaling Interface (DSI):
  - DSI DevPak
  - DSI MTP3 (needed only if the MTP3 stack is required on the host rather than on the board)
  - DSI SCCP

Information about Dialogic products is on the Dialogic Web site:

`http://www.dialogic.com`

See the associated installation guides for information on installing and using the board and software.

Downloading the Service Broker Software

You download the Service Broker software from the Oracle software delivery Web site:

`http://edelivery.oracle.com`

To download the Service Broker software:

1. In a web browser, navigate to the Oracle software delivery Web site and follow the instructions.
2. When prompted to select a product pack, select **Oracle Communications Applications** and select the platform for your system configuration.

   See "System Requirements" for a description of supported configurations.
3. Select **Oracle Communications Service Broker** media pack and click **Continue**.

4. Follow the instructions to complete the download of the Service Broker media pack.
5. When the download completes, extract the contents of the media pack.

The Service Broker media pack contains installation archives that are named after the system configuration for which it is intended. See "System Requirements" for a list of the archive file names.

Unpacking the Installer

The installation files for each supported operating system are delivered in their own archive file. To unpack the archive:

1. Open a command line shell and navigate to the directory to which you extracted the Service Broker media pack.
2. Unpack the installation archive appropriate for your target system, in the form:

   `unzip` *install_file*

   Where *install_file* is the path and name of the installation archive file. Example on an Intel x86 Linux system:

   ```
   unzip ./ocsb600-linux_x86.zip
   ```

The installer setup files are unpacked to the following directory:

installer_directory`/`*operating_system*`/Disk1/install`

where *operating_system* is your operating system.

Example of unpacked install directory for Solaris 10 64-bit SPARC:

installer_directory`/sparc64/Disk1/install`

Setting Socket Buffer Sizes

To help minimize packet loss, the operating system socket buffers need to be large enough to handle the garbage collection of incoming network traffic.

Service Broker uses Oracle Coherence as its in-memory data grid caching manager. By default, Coherence attempts to allocate a socket buffer of 2 MB. If your operating system is not configured to allow buffers this large, Coherence will utilize smaller buffers. Most versions of Unix have a very low default buffer limit, which you should increase to at least 2 MB.

Coherence will display the following warning if the buffer size is not set to at least 2 MB:

```
UnicastUdpSocket failed to set receive buffer size to 1428 packets
(2096304 bytes); actual size is 89 packets (131071 bytes). Consult your OS
documentation regarding increasing the maximum socket buffer size.
Proceeding with the actual value may cause sub-optimal performance.
```

It is safe to operate with the smaller socket buffer size, but Oracle recommends that you increase it to 2 MB.

To change the socket buffer size to 2 MB on a Linux or Solaris system:

1. Log in as root.
2. Open a command shell, if necessary.
3. Run the command appropriate for your operating system:

- On Linux:

  ```
  sysctl -w net.core.rmem_max=2096304
  ```

  ```
  sysctl -w net.core.wmem_max=2096304
  ```

  The first command sets the read memory size, and the second sets the write memory size.

- On Solaris:

  ```
  ndd -set /dev/udp udp_max_buf 2096304
  ```

4. In Linux, add the **rmem_max** and **wmem_max** name-value pairs as shown in the previous step to **/etc/sysctl.conf** to make the configuration change permanent.

4

Installing the Software

The Service Broker installer is an Oracle Universal Installer. This chapter describes how to use the Oracle Universal Installer in graphical mode or from the command line to install the Oracle Communications Service Broker software.

About the Service Broker Installer

Running the installer installs a domain server software and the Administration Console software. It installs all components necessary to run the standalone Administration Console or Web Administration Console server.

The installer can also install a JDK with a JVM if you choose this option.

The server software installed can be used to run a signaling server, a processing server or a unified server. The role of a server is defined only when you start the server.

The Service Broker software is installed in the Oracle home directory. See "Choosing an Oracle Home" for more information.

When you run the installer, you choose a Service Broker product type to install, such as the Service Controller or Policy Controller. To install multiple products, run the installer multiple times. Choose the same Oracle home directory for the additional installations - this will retain a single instance of the Administration Console and server software. The software associated with the additional products is added to the existing installation.

The Oracle Universal Installer can run in either graphical mode or console mode. To run the installer, you must be logged in to the operating system as a non-root user. For general information about using the Oracle Universal Installer, see *Oracle Universal Installer and OPatch User's Guide*.

Choosing an Oracle Home

An Oracle home is the parent directory on a host computer into which the Service Broker software is installed.

The Oracle home location is defined in an environment variable.

Each Oracle home has a name and a directory path associated with it, which you specify during installation:

- The name identifies the program group associated with a particular home and the Oracle services installed on this home. The Oracle home name must be between 1 to 127 characters long, and can include only alphanumeric characters and underscores.

- The path is the full path to an Oracle home directory where products are installed.

  In the Service Broker documentation, the Oracle home directory is referred to as *Oracle_home*.

If you run the installer for the first time on a host, Oracle recommends that you designate an *Oracle_home* location that is either an empty directory or a non-existing directory for the installer to create.

Service Broker uses the Oracle Universal Installer, which supports the installation of multiple Oracle home directories on the same host. However, if you run the installer multiple times, Oracle recommends that you choose the same Oracle home directory for the additional installations.

An *Oracle_home* inventory or local inventory directory contains only information relevant to a particular *Oracle_home*. Each *Oracle_home* directory contains an **inventory** subdirectory.

Choosing an Oracle Central Inventory

The Oracle central inventory directory contains installation logs corresponding to all installations performed on a particular host. The installation logs for an installation are identified by the timestamp in the log file names.

When you install the first Oracle product in a host, the installer prompts you to specify the location of the central inventory directory. The location of the directory is defined in the following files:

The location of the Oracle Central Inventory is defined in the following files:

- Linux:

  /etc/oraInst.loc

- Solaris:

  /var/opt/oracle/oraInst.loc

Choosing an Installation Mode

You can run the Service Broker installer by using a graphical user interface (graphical mode), using the console interface (console mode), or by using a command-shell script with response files (silent mode).

You can use the graphical mode to walk through the installation by providing information in the dialog boxes when prompted. This method is useful when performing a small number of installations.

If your operating system cannot display the graphical installer, you must use the installer in console mode or silent mode. This method is most useful when installing a large number of servers on multiple hosts. By using the response files, you can also automate the installation of a server or an Administration Console.

Choosing a JDK

A JDK must be installed on every host that runs a server, a stand-alone Administration Console, or the Web Administration Console.

The installation package comes with two JDKs:

- Oracle JRockit
- Sun Java JDK (with the HotSpot JVM)

You can also choose to reuse a supported JDK that is already installed on your host.

See "Supported Configurations" for information about supported JDKs.

Running the Installer in Graphical Mode

To install Service Broker using the Oracle Universal installer in graphical mode:

1. In a command shell, navigate to the installer directory:

   installer_directory`/`*operating_system*`/Disk1/install`

   Where:

   - *installer_directory* is the directory to which you extracted the product installation archive.
   - *operating_system* is the identifier for the operating system you are using, such as **sparc** or **linux**.

2. Enter the following command to launch the installer:

   ```
   ./runInstaller
   ```

   The Oracle Universal Installer is launched in graphical mode.

3. The installer displays a series of windows that prompt you for the information described in Table 4–1. Follow the instructions in the table.

Table 4–1 Installer Windows in Graphical Mode

In this window	Perform the following action
Welcome	Click **Next** to proceed with the installation. You may cancel the installation at any time by clicking **Cancel**.
Specify Inventory directory and credentials	This window is displayed only during the first installation of Oracle products on a system. Specify the full path of the Oracle inventory directory. Choose the operating system group that has write permissions to the inventory directory. Then, click **Next**.
Select a Product to Install	Choose the Service Broker product to install from these options: ■ Online Mediation Controller 6.0.0.1.0 ■ Policy Controller 6.0.0.1.0 ■ Service Controller 6.0.0.1.0 ■ Social Voice Communicator 6.0.0.1.0 ■ Virtual Private Network 6.0.0.1.0 ■ Co-deployed Service Controller and Online Mediation Controller 6.0.0.1.0 ■ Co-deployed Policy Controller and Online Mediation Controller 6.0.0.1.0 ■ Co-deployed Virtual Private Network and Social Voice Communicator 6.0.0.1.0 See *Oracle Communications Service Broker Concepts Guide* for information about each product option.

Table 4–1 (Cont.) Installer Windows in Graphical Mode

In this window	Perform the following action
Specify Home Details	Specify the directory that will serve as the parent directory for all Service Broker products installed on the target system. In this documentation, the home directory is referred to as *Oracle_home*. See "Choosing an Oracle Home" for details about the home directory. If you already have a home directory on your system, you can select that directory (recommended) or create a new home directory. To select an existing home directory: ■ Select the *Oracle_home* name from the **Name** list. The path to the *Oracle_home* directory is automatically entered in the **Path** field. To create a new *Oracle_home* directory: ■ Enter a name for the installation in the **Name** field. ■ Enter the path to home directory in the **Path** field. The installer will create the directory for you. You can alternatively click **Browse** to select a directory from the Choose Directory dialog box. Click **Next** to continue.
Product update registration page	(Optional) Register your installation with My Oracle Support. By registering, Oracle Support notifies you immediately of any security updates that are specific to your installation. To register your installation: ■ In the **Email** field, enter the email address where you wish to be notified of updates. ■ Select the **I wish to receive security updates via My Oracle Support** check box. ■ In the **My Oracle Support Password** field, enter the password for your Oracle support account. If you have not registered with Oracle Support, go to the following My Oracle Support Web site and register to obtain a My Oracle Support account: `https://support.oracle.com/CSP/ui/flash.html` To decline registration: ■ Deselect **I wish to receive security updates via My Oracle Support**. For more information about the advantages of registering your installation with My Oracle Support, see *Oracle Configuration Manager Installation and Administration Guide*. Click **Next** to continue.

Table 4–1 (Cont.) Installer Windows in Graphical Mode

In this window	Perform the following action
Java Installation Option	Specify whether to install a JDK. If you do not install a JDK, you are prompted for the path to an existing JDK. In a production system, you should install one of the certified versions of the JDK provided by the installer. For a test and development system, you can use a different, supported JDK. Also for test and development environments, multiple Service Broker installations can share a single JDK if they reside on the same machine. See "System Requirements" for details about supported JDKs. To install a JDK: ■ Click **Yes**. ■ Click **Next**. The JVM Selection window opens. To use an already installed JDK: ■ Click **No**. ■ Click **Next**. The Java Installation window opens.
Java Selection	This window is only displayed if you selected to install a JDK. Specify which JVM to install: ■ Click **Java JDK** to install the Oracle Java JDK (with the HotSpot JVM). ■ Click **Oracle JRockit JDK** to install the Oracle JRockit JDK. Click **Next** to continue.
Java Installation	This window is only displayed if you selected not to install a JDK. Specify the path to an existing Java installation: ■ In the **Java Base Directory** field, enter the path to the directory where the JDK is installed or click **Browse** to select a directory from the Choose Directory dialog box. The path should be the same as the environment variable `JAVA_HOME` for your Java installation. See "System Requirements" for details about supported JDKs.
Summary	This window displays a summary of the installation, including general information about the installation type, directories, the product components to be installed, the approximate installed size of each component, and the total size of all components to be installed. Click **Install**.
Install	This window is displayed while the components are being installed. The window displays the status of the installation. You can cancel the installation at any time by clicking the **Stop installation** button.
Execute Configuration scripts	This window is displayed only during the first installation of Oracle products on a system. As indicated, run the configuration script, **orainstRoot.sh**, in a new console as the root user. The script configures permissions for the Oracle inventory directory. When the script completes the configuration, click the **OK** button in the Execute Configuration scripts window.

Table 4–1 (Cont.) Installer Windows in Graphical Mode

In this window	Perform the following action
End of Installation	This window describes the outcome of the installation. Click the **Exit** to exit the installation program. Click **Yes** to confirm your choice.

When the installation is completed, the directory structure described in "Directory Structure After Installation" is created on your server.

Running the Installer in Console Mode

The console installer enables you to install the product from a command line. When using the console installer, you can specify the installation parameters in one of two ways:

- By entering the parameters in response to installer prompts. See "Passing Console Installer Parameters Interactively" for instructions.
- By entering all parameters at the command line when invoking the console installer script. See "Passing Console Installer Parameters at the Command Line" for instructions.

The following sections provide more information about each console installer operating mode.

Passing Console Installer Parameters Interactively

In interactive mode, you supply installation parameters to the console installer by responding to on-screen prompts.

To install Service Broker using the console-based installer in interactive mode:

1. In a command shell, navigate to the installer directory:

   installer_directory/*operating_system*/**Disk1/install**

2. Enter the following command to launch the installer:

   ```
   ./consoleInstaller.sh
   ```

   The Oracle Universal Installer is launched in console mode.

3. At the **Please select a product to install** prompt, enter the number corresponding to the Service Broker product you want to install from the following options:
   - **1** to install the Online Mediation Controller
   - **2** to install the Policy Controller
   - **3** to install the Service Controller
   - **4** to install the Social Voice Controller (SVC) only
   - **5** to install the Virtual Private Network (VPN) only
   - **6** to install the co-deployed Policy Controller and Online Mediation Controller
   - **7** to install the co-deployed Service Controller and Online Mediation Controller
   - **8** to install the co-deployed VPN and SVC

4. Enter the name of the Oracle home directory in which you want to install the software. The Oracle home name serves as the alias for the parent directory of the Service Broker products. For example:

   ```
   Please enter an Oracle Home name for the installation: OHOME1
   ```

   In this documentation, the home directory is referred to as *Oracle_home*. See "Choosing an Oracle Home" for details about the home directory.

   If you already have a home directory on your system, you can select that directory or create a new home directory.

5. Enter the full directory path to the *Oracle_home*. For example:

   ```
   Please enter the full installation path: /home/oracle
   ```

6. At the next prompt, enter your My Oracle Support email address or other email address at which you want to receive information on security issues.

7. Choose whether you want to receive security updates through My Oracle Support by entering **Y** for yes, or **N** for no.

8. At the Java Installation Option prompt, enter **Y** to install the JDK as part of the installation, or **N** to choose an existing JDK. If you do not install a JDK, you are prompted for the path to an existing JDK.

   In a production system, you should install one of the certified versions of the JDK provided by the installer.

   For a test and development system, you can use a different, supported JDK. Also for test and development environments, multiple Service Broker installations can share a single JDK if they reside on the same machine.

   See "System Requirements" for details about supported JDKs.

9. If you did not elect to install the JDK, specify the full directory path to the existing JDK on the system.

10. If you elected to install the JDK, choose the JDK type by entering **1** to install the Java JDK (with the HotSpot JVM), or **2** to install Oracle JRockit JDK.

    The installer verifies that there are sufficient system resources for the installation and, if so, proceeds with the installation.

    The installer displays the progress of installation and status messages on screen. When complete, it writes the details of the installation procedure to the log file indicated in the on-screen output. If the installation succeeded, the directory structure described in "Directory Structure After Installation" is created on your server.

11. If this is the first Oracle product installation on this machine, when prompted, follow the on-screen instructions to run the **orainstRoot.sh** script located in your Oracle inventory directory. You must run the script as the root user.

Passing Console Installer Parameters at the Command Line

Instead of supplying parameters to the installer interactively (as described in "Passing Console Installer Parameters Interactively"), you can pass them as command-line options when invoking the installer script.

Run the installer by using the following syntax:

```
./consoleInstaller.sh -silent INSTALL_TYPE INSTALL_JVM JVM_TYPE ORA_HOME_NAME
INSTALL_PATH SECURITY_UPDATES_VIA_MYORACLESUPPORT DECLINE_SECURITY_UPDATES
```

JAVA_HOME MYORACLESUPPORT_USERNAME MYORACLESUPPORT_PASSWORD

For information on the parameters, see table Table 4–2.

For example:

```
./consoleInstaller.sh -silent 1 1 1 OHOME1 ~/OHOME1 false true
```

The example command installs the Service Controller solution with Oracle JRockit in the OHOME1 directory. It declines security updates. Note that it excludes the *JAVA_HOME*, *MYORACLESUPPORT_USERNAME*, *MYORACLESUPPORT_PASSWORD* arguments, which are not needed given the other argument values provided.

If specified, optional arguments must appear in the order shown.

If this is the first Oracle product installation on this machine, after you run the installer script, you are prompted to run the **orainstRoot.sh** script as the root user. The script is located in your Oracle inventory directory. Follow the on-screen instructions to run the script.

Running the Installer in Silent Mode

You can install Service Broker using Oracle Universal Installer in silent mode.

You can specify the necessary installation parameters, such as what to install, the home directory, and so on, in one of two ways:

- By editing the parameters in the **silentInstaller.sh** installer file. Typically, this is the method you will use. The file includes all the mandatory parameters and the most frequently used parameters. Default values are given.

  See "Installing in Silent Mode Without a Response File" for instructions.

- By editing the installation response file. This file includes the same parameters as the **silentInstaller.sh** file, plus additional parameters that you can set. The response file, along with the **-silent** parameter, is given as input to the **runInstaller** utility.

  See "Installing in Silent Mode With a Response File" for instructions.

In either case, you use the parameters described in "Installation Parameters" to specify parameters of the installation.

Installing in Silent Mode Without a Response File

To install Service Broker in silent mode without using a response file:

1. In a command shell, navigate to the following directory:

   installer_directory/*operating_system*/**Disk1/install/**

2. Open the **silentInstaller.sh** file for editing, and modify the installation parameters as required. Table 4–2 lists the installation parameters.

3. Enter the following command to launch the installer:

   ./silentInstaller.sh

   The installation progress is output to the console.

4. If this is the first Oracle product installation on this machine, when prompted, follow the on-screen instructions to run the **orainstRoot.sh** script located in your Oracle inventory directory. You must run the script as the root user.

The directory structure described in "Directory Structure After Installation" is created on your server.

Installing in Silent Mode With a Response File

The installer response file provides parameters that correspond to those in **silentInstaller.sh** file, along with additional parameters. The installer response file is used with the **runInstaller** utility.

To install Service Broker in silent mode using a response file:

1. In a command shell, navigate to the following directory:

   installer_directory`/`*operating_system*`/Disk1/stage/Response/`

2. Edit the installer response file appropriate for your Service Broker implementation, from the following choices:

   - **`codeploy_pcomc.Complete.rsp`** to install both the Policy Controller and Online Mediation Controller
   - **`codeploy_scomc.Complete.rsp`** to install co-deployed Service Controller and Online Mediation Controller
   - **`codeploy_vpnsvc.Complete.rsp`** to install co-deployed VPN and Social Voice Communicator (SVC)
   - **`onlinemediation_controller.Complete.rsp`** to install the Online Mediation Controller
   - **`policy_controller.Complete.rsp`** to install the Policy Controller
   - **`service_controller.Complete.rsp`** to install the Service Controller
   - **`socialvoice_communicator.Complete.rsp`** to install SVC
   - **`virtualprivate_network.Complete.rsp`** to install VPN

   Table 4–2 describes the basic parameters in the installer response file. For information on additional parameters in the response file, see the comments contained within the response file.

   Note that the **TOPLEVEL_COMPONENT** setting in the files, which determines the product type of the installation, is preset to the value corresponding to the file's product type, as listed above.

3. Save the file after making your changes.

4. Navigate to the installer directory:

   installer_directory`/`*operating_system*`/Disk1/install`

5. Enter the following command to launch the installer:

   `./runInstaller -silent -responseFile` *full_path_to_response_file*

   For example, to install the Policy Controller solution:

   `./runInstaller -silent -responseFile` *installer_directory*`/linux/Disk1/stage/Response/rcc_policy_controller.Custom.rsp`

   The installation progress is output to the console.

6. If this is the first Oracle product installation on this machine, when prompted, follow the on-screen instructions to run the **orainstRoot.sh** script located in your Oracle inventory directory. You must run the script as the root user.

The directory structure described in "Directory Structure After Installation" is created on your server.

Installation Parameters

Table 4–2 lists the installation parameters accepted by the silent installer.

Table 4–2 Installation Parameters

Parameter	Description

`TOPLEVEL_COMPONENT`	Specifies the Service Broker product to install.
	Mandatory
	The values for the parameter differ by installation method. In the **silentInstaller.sh** script, use one of the following values:
	■ `1` to install the Online Mediation Controller
	■ `2` to install the Policy Controller
	■ `3` to install the Service Controller
	■ `4` to install the Social Voice Communicator (SVC)
	■ `5` to install the Virtual Private Network (VPN)
	■ `6` to install the co-deployed Policy Controller and Online Mediation Controller
	■ `7` to install the co-deployed Service Controller and Online Mediation Controller
	■ `8` to install the co-deployed VPN and SVC
	Example:
	`TOPLEVEL_COMPONENT=1`
	In a response file, specify the component using the syntax:
	{"*component*","*version*"}
	where *component* is the Service Broker product to install and *version* is its version. The *component* may be one of the following:
	■ **onlinemediation_controller** for the Online Mediation Controller
	■ **policy_controller** for the Policy Controller
	■ **service_controller** for the Service Controller
	■ **socialvoice_communicator** for the Social Voice Communicator (SVC)
	■ **virtualprivate_network** for the Virtual Private Network (VPN)
	■ **codeploy_pcomc** for the co-deployed Policy Controller and Online Mediation Controller
	■ **codeploy_scomc** for the co-deployed Service Controller and Online Mediation Controller
	■ **codeploy_vpnsvc** for the co-deployed VPN and SVC
	Example:
	`TOPLEVEL_COMPONENT={"service_controller","6.0.0.1.0"}`

Table 4–2 (Cont.) Installation Parameters

Parameter	Description
INSTALL_JVM	Specifies whether to install one of the bundled JDKs or to use an already installed JDK. Mandatory In a production system, you should install one of the certified versions of the JDK provided by the installer. For test environments, you can use an already installed, supported JDK. Also for test environments, multiple Service Broker installations can share a single JDK if they reside on the same machine. Use: ■ **1** to install the JVM (this is the default) ■ **0** to not install the JVM (if you already have one installed) Example: **INSTALL_JVM=1**
JVM_TYPE	The type of JVM to install. Mandatory if the value of **INSTALL_JVM** is **0**. Use: ■ **1** to install the Java JDK (with the HotSpot JVM) ■ **2** to install the Oracle JRockit JDK Example: **JVM_TYPE=1**
JAVA_HOME	The path to the directory of your Java JDK. Mandatory if the value of **INSTALL_JVM** is **0**. The path should be the same as the environment variable **JAVA_HOME** for your Java installation. Example: **JAVA_HOME="/usr/local/java/jdk"**
ORACLE_HOME_NAME	The name of the Oracle home directory. Mandatory If you have previously installed an Oracle product using the Oracle Universal Installer, you already have an Oracle home defined. If you want to reuse the existing Oracle home, enter its name here. When you use an existing Oracle home name, the value you specify for ORACLE_HOME is overridden with the existing Oracle home directory path. Example: **ORACLE_HOME_NAME="ORACLE_HOME"** The default Oracle home is **OHome1**
ORACLE_HOME	The path to use as your *Oracle_home* directory. Mandatory Specify the directory that will serve as the central support directory for all Oracle products installed on the target system. Example: **ORACLE_HOME="/usr/local/oracle_home"** The default path is **~/OHome1**

Table 4–2 (Cont.) Installation Parameters

Parameter	Description
DECLINE_SECURITY_UPDATES	Indicates whether you want to receive security updates related to Oracle products. Mandatory Enter **true** to decline security updates, or **false** to receive security updates. Example: **DECLINE_SECURITY_UPDATES="true"**
SECURITY_UPDATES_VIA_ MYORACLESUPPORT	Indicates whether you want to receive security updates through your Oracle My Oracle Support account. Mandatory Enter **true** to receive security updates via My Oracle Support, or **false** to decline security updates. If you have not registered with Oracle Support, go to the following My Oracle Support Web site and register to obtain a My Oracle Support account: https://support.oracle.com/CSP/ui/flash.html Example: **SECURITY_UPDATES_VIA_MYORACLESUPPORT="false"**
MYORACLESUPPORT_USERNAME	Specifies your My Oracle Support email address or other email address where you wish to be notified of updates. Mandatory if **DECLINE_SECURITY_UPDATES** is **false** or **SECURITY_ UPDATES_VIA _MYORACLESUPPORT** is **true**. Example: **MYORACLESUPPORT_USERNAME="user@example.com"**
MYORACLESUPPORT_PASSWORD	Specifies the password for your Oracle support account. Mandatory if **DECLINE_SECURITY_UPDATES** is **false** or **SECURITY_ UPDATES_VIA _MYORACLESUPPORT** is **true**. Example: **MYORACLESUPPORT_PASSWORD="***password***"**

Directory Structure After Installation

Figure 4–1 illustrates the high-level directory structure that is created during installation.

Figure 4–1 Installed Directory Hierarchy

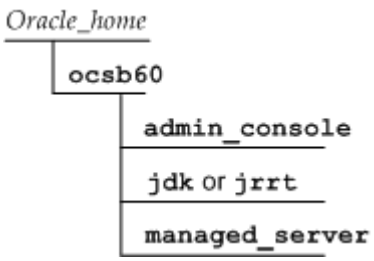


Table 4–3 describes each directory and its contents.

Table 4–3 Installed Directories and Content Descriptions

Directory	Description
Oracle_home`/ocsb60`	Base directory for your installation. *Oracle_home* is specified during the installation.
Oracle_home`/ocsb60/admin_console`	Contains binaries for the Administration Console, and the start scripts and modules related to the Administration Console.
Oracle_home`/ocsb60/managed_server`	Contains binaries for the Processing Server or Signaling Server, and the start scripts and modules related to the server.
Oracle_home`/ocsb60/jdk`	Contains the Java JDK (with the HotSpot JVM). This directory is created only if you specified to install the Java JDK during the installation.
Oracle_home`/ocsb60/jrrt`	Contains the Oracle JRockit JDK. This directory is created only if you specified to install the Oracle JRockit JDK during the installation.

Deinstalling Service Broker

You can deinstall Service Broker or any of its components by running the Oracle Universal Installer. When you deinstall, the installer removes all files and folders that it originally installed, but does not remove any files and folders that you created or modified, such a domain directories and configuration files.

See "Deinstalling Using Graphical Mode" for instructions on deinstalling Service Broker or one of its components using Oracle Universal Installer in graphical mode.

See "Deinstalling Using Silent Mode" for instructions on deinstalling Service Broker or one of its components using Oracle Universal Installer in silent mode.

Deinstalling Using Graphical Mode

To deinstall Service Broker or any of its components using the Oracle Universal Installer in graphical mode:

1. In a command shell, navigate to the installer directory:

   installer_directory`/`*operating_system*`/Disk1/install`

2. Enter the following command to launch the installer:

   ```
   ./runInstaller
   ```

   The Oracle Universal Installer is launched in graphical mode.

3. In the Welcome window, click **Deinstall Products**.

   The Inventory window displays all Oracle homes on the host.

4. In the **Contents** tab, select the Oracle home or product within the Oracle home that you want to deinstall.

   Selecting the Oracle home removes all products within that Oracle home. Alternatively, expand the Oracle home and select individual components within that home to deinstall.

Tip: To save a text copy of the inventory tree and selected components before you remove components, click **Save As**. Navigate to a directory where you want to save the tree, enter the file name to use, and click **Save**.

5. Click **Remove**.

   The Confirmation window opens.

6. Click **Yes** to confirm your selection.

   The installer deinstalls the selected components.

7. Click **Close** to dismiss the Inventory window.

Deinstalling Using Silent Mode

To deinstall Service Broker or one of its components in silent mode, you run the **runInstaller** utility, specifying the **-deinstall** option.

The simplest way to specify the components to deinstall is on the command line. However, you can alternatively specify the components to deinstall in an installer response file. The installer response files are located in the installation directory for your platform:

installer_directory/*operating_system*`/Disk1/stage/Response/`

See "Installing in Silent Mode With a Response File" for more information about the installer response files.

Table 4–4 describes the parameter-value pairs that you set, either on the command line or in the installer response file.

Table 4–4 Parameters for Deinstalling Components

Name	Description
`DEINSTALL_LIST`	Specify the component to deinstall. Mandatory Use the following syntax: {"*component*","*version*"} where *component* is the Service Broker product to remove and *version* is its version. The *component* value may be one of the following: ■ **policy_controller** for the Policy Controller ■ **service_controller** for the Service Controller ■ **onlinemediation_controller** for the Service Controller ■ **socialvoice_communicator** for SVC ■ **virtualprivate_network** for VPN ■ **codeploy_pcomc** for the co-deployed Policy Controller and Online Mediation Controller ■ **codeploy_ccomc** for the co-deployed Service Controller and Online Mediation Controller ■ **codeploy_vpnsvc** for the co-deployed VPN and SVC Examples: `DEINSTALL_LIST={"policy_controller","6.0.0.1.0"}`

Table 4–4 (Cont.) Parameters for Deinstalling Components

Name	Description
REMOVE_HOMES	Specify the full path to the Oracle home directory to remove. Optional Use the following syntax: {*path_to_Oracle_home*} Example: **REMOVE_HOMES={/usr/local/oracle_home}**

You can use the **runInstaller** script to get a list of the installed components and their versions:

1. In a command shell, navigate to the installer directory:

   installer_directory/*operating_system*/**Disk1/install**

2. Enter the following command:

   ./runInstaller -deinstall

   The installed components are listed in the output.

To deinstall Service Broker or one of its components:

1. In a command shell, navigate to the installer directory:

   installer_directory/*operating_system*/**Disk1/install**

2. Enter one of the following commands to launch the installer.

   - To specify the components to deinstall on the command line, enter:

     ./runInstaller -deinstall -silent DEINSTALL_LIST={"*component***","***version***"} REMOVE_HOMES=**{*path_to_Oracle_home*}

     where *component*, *version*, and *path_to_Oracle_home* are specified as described in Table 4–4.

     For example:

     ./runInstaller -deinstall -silent DEINSTALL_LIST={"service_controller","6.0.0.1.0"} REMOVE_HOMES={/home/oracle/OHome1}

   - To use an installer response file in which the components to deinstall are specified, enter:

     ./runInstaller -deinstall -silent -responseFile *full_path_to_response_file*

5

Setting Up a Domain

This chapter describes how to set up an Oracle Communications Service Broker domain.

About Domain Setup

Whether a domain is a signaling domain, a processing domain, or a unified domain, setting up a domain includes:

1. Considering the domain configuration directory. See "Domain Configuration Considerations" for more information.
2. Installing the domain servers and a single instance of the Administration Console.
3. Creating the domain
4. Starting the domain Web server
5. Starting the Administration Console
6. Adding servers to the domain
7. Deploying a distributed cluster
8. Configuring data storage
9. Starting the domain servers

Domain Configuration Considerations

A domain has an associated domain configuration, which is stored in a domain configuration directory. All configuration data and OSGi software bundles are located in the domain configuration directory. See "Understanding the Domain-Based Administration Model" for more information about the domain configuration directory.

The domain configuration directory is referred to as *domain_home* in this document. The domain configuration directory needs to be accessible to the Administration Console and all servers in the domain.

The Administration Console requires read and write access to the domain configurations. The servers require only read access to the domain configurations.

The Administration Console and the domain servers can access the domain configuration directory on a shared file system or, for a hosted domain, using a Web server that serves the files:

- If you use the shared file-system, set up all servers so that they can access the domain configuration directory.
- If you use a Web server to serve the domain files, set up the Web server with read access to the domain configuration directory and map the directory to a URL.

The domain configuration directory is created only when you run the domain creation script and create the domain. You specify the manner of access when creating the domain (as described in "Creating the Domain").

For general information about sharing or hosting files, refer to your operating system or Web server documentation.

Installing the Administration Console and Domain Servers

The Service Broker installation process installs the software of a domain server and the Administration Console. See "About the Service Broker Installer" for more information.

The Administration Console can be installed and run from any host that has access to the other domain servers and to the domain configuration directory.

Because the domain configuration directory is independent of the Administration Console instance used to access it, you can have as many Administration Consoles installed as you wish. Run the installer on each host where you want to run the standalone Administration Console or the Web Administration Console server. Note however that only one Administration Console can manage a specific domain at any given time. Running multiple instances of the Administration Console that access the same domain simultaneously will result in an error.

You run the installation process to install the Administration Console and a single server instance. To install more server instances, run the installer again for each server that you want to add.

If you install multiple server instances on the same physical host, do not run the installer multiple times, because this will install multiple instances of the Administration Console on the same host as well. While having multiple Administration Console instances installed in a domain does not interfere with the system, they are unneeded. Instead, run the installer once and then copy the contents of the **managed_server** directory to a new directory for each server you want to add.

To create multiple servers on a single physical host:

1. Run the installer on the physical host. See "Installing the Software" for instructions.

   The default installed Administration Console directory is:

   Oracle_home`/ocsb60/admin_console`

   The default installed server directory is:

   Oracle_home`/ocsb60/managed_server`

2. Create a new server directory for each additional server.

   To simplify your directory structure, you may choose to create the directories alongside the existing server directory in an existing Oracle home. For example:

   Oracle_home`/ocsb60/`*new_server_directory*

3. Copy the contents of the installed server directory to each new server directory that you added.

To create additional servers on another host repeat this procedure on a different host.

At a later stage, after creating the domain and starting the Administration Console, you will have to define each server in the domain configuration by specifying its name and listening ports. See "Adding a Server to the Domain" for more information.

Table 5–1 and other tables in the following sections illustrate a sample domain with four servers. Table 5–1 outlines the directory structure for the components of the example domain.

Table 5–1 Example Directory Structure for Multiple Servers on a Single Physical Host

Physical Host	Directory Path	Description
sb_01	*oracle_home*`/ocsb60/admin_console`	Administration Console
sb_01	*oracle_home*`/ocsb60/managed_server`	Default server installation directory Use as the source when creating the directories for the new servers on the sb_01 host.
sb_01	*oracle_home*`/ocsb60/managed_1`	First server instance directory
sb_01	*oracle_home*`/ocsb60/managed_2`	Second server instance directory
sb_02	*oracle_home*`/ocsb60/admin_console`	Administration Console
sb_02	*oracle_home*`/ocsb60/managed_server`	Default server installation directory Use as the source when creating the directories for the new servers on the sb_01 host.
sb_02	*oracle_home*`/ocsb60/managed_3`	Third server instance directory
sb_02	*oracle_home*`/ocsb60/managed_4`	Fourth server instance directory

After installing the domain servers and Administration Console you can move on to the next step which is creating a domain.

Creating the Domain

By now, you have installed the domain servers and Administration Console, but you did not define them as a related group of servers.

Creating the domain generates the domain configuration directory that contains the server configuration settings and artifacts that the domain servers run. When you start a server, you pass it the domain archive file. This file controls all aspects of the server's operating behavior.

The contents of the domain reflect a specific Service Broker product type, such as Service Controller, Policy Controller, or VPN. When you create a domain, you are prompted for the product type of the domain. Only those products types that have been specifically installed in a given Oracle home are available. See "Installing the Software" for more information.

Domain types reflect the role that servers will perform in the domain. See "Understanding the Domain-Based Administration Model" for information about domain types.

Servers can access the contents of a domain configuration directory in the following ways:

- Through the local or shared file system
- Through a Web server that hosts the domain configuration files

You choose the access mode when creating the domain. For most deployments, file system access to the domain configuration is sufficient. However, you may choose to use hosted access if your local network topology requires it.

The following instructions assume the use of file system access mode. For more information on hosted domain access, see *Oracle Communications Service Broker System Administrator's Guide*.

Securing the Domain

The domain creation script creates domains based on the settings in property files in the Administration Console directory. By default, the setting that controls security in the domain is enabled. This setting applies to the connections between components in the domain; that is, between the Administration Console and servers.

Before you run the script, you must either disable the setting or, if you want to retain security for the domain, set up the client and server certificate keystores on the host system.

For a non-production system intended for test, instructional, or evaluation purposes only, you may choose to disable security between the components, as follows:

1. Open the following document for editing:

   Oracle_home/**ocsb60/admin_console/properties/common.properties**

2. Set the axia.ssl property to false:

   axia.ssl=false

   For more information on the properties in the file, see *Oracle Communications Service Broker System Administrator's Guide*.

3. Save and close the file.

For information on setting up a secure domain, see the discussion on configuring security between Service Broker components in *Oracle Communications Service Broker System Administrator's Guide*.

Creating the Domain in Interactive Mode

The following instructions describe how to create a domain using the domain creation script.

If you kept SSL security enabled, you must create the client and server security keystores on the target system before attempting to run the domain creation script. Otherwise, the domain creation script will not complete successfully. See "Securing the Domain" for information about disabling SSL security.

The domain creation script can operate interactively or in silent mode. The following instructions describe how to use the script interactively. See "Creating the Domain in Silent Mode" for information about running the script in silent mode.

To create a domain using the domain creation script in interactive mode:

1. In a command shell, navigate to the following directory:

   Oracle_home/**oracle/ocsb60/admin_console**

2. Run the create domain script, as follows:

   ./create_domain.sh

3. If you retained the default value (**true**) for the SSL security setting, enter **Y** at the following prompt:

   ```
   The SSL security flag for the Administration Console is enabled.
   Note that the client and server keystores must be created for the domain
   creation to succeed.
   Continue?
   [Y]es, [N]o:
   ```

   The prompt does not appear if you disabled SSL security for the domain. See "Securing the Domain" for information about disabling SSL security.

   If you still need to create the security keystores, enter **N** to cancel the domain creation script.

4. When prompted for the index of the product for which to create the domain, enter the number corresponding to the product domain type you want to create.

   Options are:

   - **Online-Mediation-Controller**
   - **Policy-Controller**
   - **Service-Controller**
   - **Virtual-Private-Network**
   - **Social-Voice-Communicator**
   - **Co-Deployed-Policy-Controller-and-Online-Mediation-Controller**
   - **Co-Deployed-Service-Controller-and-Online-Mediation-Controller**
   - **Co-Deployed-Virtual-Private-Network-and-Social-Voice-Communicator**

   Only the product types installed in the Oracle home directory are shown.

5. Enter the number corresponding to the domain type you want to create. The options are: **1** for a unified domain, **2** for a processing domain, or **3** for a signaling domain.

6. If you selected the unified (**1**) for the domain type, you will be prompted for **Service Mode**. At the prompt, enter **1** for service availability (if you create a domain for a single-domain production deployment), or **2** for service availability and continuity (if you create a domain for a single-domain test and evaluation deployment).

7. When prompted for the domain path, enter the full path to the directory where the domain configuration will be created. For example:

   /home/oracle/domains/mydomain

   This directory is referred to as *domain_home* in this document.

8. At the **Enable domain SSL** prompt, enter **true** to enable SSL or **false** to disable SSL for the domain connections. This setting controls whether Service Broker requires a secure connection between deployed domain components.

9. Enter the IP multicast address, which is the address on which servers and the Administration Console exchange certain types of internal communication within the domain. The address must be within the range 224.0.0.0 to 239.255.255.255. If you plan to set up additional domains that are part of the same cluster, keep a record of the value entered, and enter the same value when you create the other domains.

10. Enter the port number on which the servers and the Administration Console listen for multicast messages. The port number must be **1024** or greater. If you plan to set up additional domains that are part of the same cluster, keep a record of the value entered, and enter the same value when you create the other domains.
11. Enter the time-to-live for multicast messages. The value must be **1** or greater.
12. Indicate whether this domain should be a hosted domain:
    - **true** if you want servers to access the domain configuration over a web connection.
    - **false** if you want servers to access the domain files from either the local file system or a shared remote file system.
13. If you specified **true** for the hosted domain, enter the host name and port for the Web server that will provide access to the domain configuration. For example:

    http://localhost:9001/

    The port is defined in the **org.eclipse.equinox.http.jetty.http.port** property in the **hosting.properties** file.

    Note that if security is enabled then the URL should start with https:// and the port is defined in the **org.eclipse.equinox.http.jetty.https.port** property in the **hosting.properties** file.

    Note: Hosted domains are intended for test and evaluation environments only. Oracle recommends that you do not use a hosted domain in a production deployment.

14. For VPN or SVC domains, enter the password of the built-in **admin** user. The **admin** user is the administrator for the RESTful Provisioning APIs.

The following is an example of the domain creation script command and its prompts, with user input in bold:

```
./create_domain.sh
--------------------------------------------------------------------------
Welcome, please follow the steps below to create a domain.

--------------------------------------------------------------------------
Available Domains:
--------------------------------------------------------------------------
[1] Online-Mediation-Controller
[2] Policy-Controller
[3] Service-Controller
Please enter the index of the domain to create: 1

--------------------------------------------------------------------------
Available Online-Mediation Controller Domain Types:
--------------------------------------------------------------------------
[1] Unified
[2] PN
[3] SSU
Please enter the index of the domain type to create: 1

--------------------------------------------------------------------------
Available Service Modes:
--------------------------------------------------------------------------
[1] Service Availability
```

```
[2] Service Availability and Continuity
Please select the service mode: 1
Please enter the domain path: /home/oracle/orahome/mc/domain
Enable domain SSL?
Please enter one of the following: true false > false
Please enter the multicast address: 239.255.255.255
Please enter the multicast port: 1036
Please enter the multicast time-to-live (ttl): 3
Should this domain be a hosted domain?
Please enter one of the following: true false > false
```

The script creates the domain according to your selections. When the script finishes, **END SCRIPT** appears. You can now start the Administration Console and add servers to the domain.

Creating the Domain in Silent Mode

In silent mode, you run the domain creation script with the **-silent** flag, passing domain parameters as command line input to the script. Run the installer by using the following syntax:

`./create_domain.sh -silent` *product_domain_type domain_type domain_path domain_ssl multicast_address multicast_port multicast_ttl hosted service_mode hosted_url*

Where:

- *product_domain_type* is the solution type of the domain, from these options:
  - **Online-Mediation-Controller**
  - **Policy-Controller**
  - **Service-Controller**
  - **Virtual-Private-Network**
  - **Social-Voice-Communicator**
  - **Co-Deployed-Policy-Controller-and-Online-Mediation-Controller**
  - **Co-Deployed-Service-Controller-and-Online-Mediation-Controller**
  - **Co-Deployed-Virtual-Private-Network-and-Social-Voice-Communicator**
- *domain_type* is the type of the domain. The options are: **Unified**, **PN**, **SSU**, which correspond to a unified domain, processing domain, and signaling domain.
- *domain_path* is the path to the new domain. This would be the domain configuration directory, for example: /home/oracle/domains/mydomain
- *domain_ssl* specifies whether to enable SSL in the new domain: **true** or **false**.
- *multicast_address* is the multicast address on which the servers and Administration Console exchange certain types of internal messages within the domain. The address must be within the range 224.0.0.0 to 239.255.255.255.
- *multicast_port* is the multicast port for the domain.
- *multicast_ttl* is the time-to-live (TTL) heading value for multicast messages on the domain.
- *hosted* indicates whether the domain files will be hosted on a Web server: **true** or **false**.
- *service_mode* specifies whether the domain should support service continuity in addition to service availability: **true** or **false**.

- *hosted_url* is the URL of the hosted domain, if you entered **true** for the *hosted* value. If you entered **false** for the *hosted* value, this parameter is not required.

For example:

```
./create_domain.sh -silent Policy-Controller Unified
/home/oracle/orahome/domain false 239.255.255.255 1034 2 false true
```

For VPN or SVC domains, enter the password of the built-in **admin** user when prompted. The **admin** user is the administrator for the RESTful Provisioning APIs.

When the script finishes, **END SCRIPT** appears. You can now start the Administration Console and add servers to the domain.

Starting the Domain Web Server

If you created a hosted domain, that is the domain servers access the domain configuration over a web connection, then you need to start the domain web server.

The domain web server provides servers with HTTP access to the domain configuration and the OSGi bundles for the domain. The web server's sole functionality is to allow HTTP access to the domain configuration.

The Domain web server must be started by a user who has read privileges to the Domain Configuration directory. See the discussion about configuring security between Service Broker components for more information about setting up user privileges.

To start the domain web server:

1. Open a command line shell.

   Note: You must be logged in as a user who has read privileges on the file system where the domain configuration resides.

2. Change the directory to:

   *Oracle_home***/ocsb60/admin_console**

   Oracle_home is the Oracle Home directory you defined when you installed the product.

3. Enter:

   ./host.sh *directory*

   Replace *directory* with the path to the domain configuration directory.

4. If HTTPS is enabled, enter the keystore password.

5. If Basic Authentication is enabled, enter the user name and password combination to use for HTTP Basic Authentication.

When the domain web server is started, servers can access the domain configuration over HTTP or HTTPS.

The port is defined in the property **org.eclipse.equinox.http.jetty.http.port**. The default value is 9000. See the system administrator's reference appendix in *Oracle Communications System Administrator's Guide*.

Starting the Administration Console

The Administration Console provides the graphical interface for the domain configuration. After creating the domain, you can start configuring it from the Administration Console interface.

You can run the Administration Console in Web access mode or standalone mode, as described in the following sections.

Starting the Administration Console in Web Access Mode

Starting the Administration Console in Web access mode enables its browser-based user interface. It allows administrators to configure the Service Broker domain from any computer with a browser and network access to the Web Administration Console server.

The Administration Console may serve the user interface over HTTP or Secure HTTP (HTTPS). The following file contains the properties that control web access, including security for the connection and the Administration Console Web interface port:

*Oracle_home***/ocsb60/admin_console/properties/web.properties**

If you retain the default value (**true**) for the HTTP Basic Authentication requirement setting (**axia.basic.auth**), you will need to specify the username and password for accessing the user interface when starting the Web Administration Console server. Set the value to **false** to disable authentication requirements for the Administration Console interface.

By default, the Administration Console is served on the HTTPS port, as specified by the following setting:

org.eclipse.equinox.http.jetty.https.enabled=true

To use this port, you need to configure the SSL keystore for the Administration Console server.

Set this value to **false** to disable the HTTPS. You will also need to enable the HTTP port, as in the following sample portion of the file:

```
axia.basic.auth=false

org.eclipse.equinox.http.jetty.http.enabled=true
org.eclipse.equinox.http.jetty.http.port=9000

org.eclipse.equinox.http.jetty.https.enabled=false
org.eclipse.equinox.http.jetty.https.port=9000
```

For more information on **web.properties** and security, see *Oracle Communications Service Broker System Administrator's Guide*.

To start the Administration Console for Web access:

1. Open a command line shell.
2. Change to the *Oracle_home***/ocsb60/admin_console** directory.
3. Enter the following command to start the Web Administration Console server:

   ./web.sh *domain_home*

   where *domain_home* is the path to the domain configuration directory.

4. If HTTP Basic Auth is enabled in the **web.properties** file (**axia.basic.auth=true**), enter the username and password that clients must use to access the Web Administration Console interface from a browser.

   The Administration Console enforces the password complexity requirements specified in **common.properties**. By default, the password must be at least six characters in length and must contain at least one upper-case letter, lower-case letter, and digit. You can modify the requirements by editing the values of the **axia.console.password.validation** settings in the file.

5. If starting the Web Administration Console for a Policy Controller domain with HTTP Basic Auth enabled, enter a second user name and password combination when prompted. The second set of credentials are used to control access to the Policy Designer GUI.

After starting the Web Administration Console server, you access the user interface from a browser by opening the following URL:

https://*ipadress***:9000/console**

See *Oracle Communications Service Broker System Administrator's Guide* for more information about starting and using the Administration Console. See *Oracle Communications Service Broker Policy Controller Implementation Guide* for more information about the Policy Designer GUI.

Starting the Administration Console in Standalone Mode

In standalone mode, the Administration Console runs as a Java client application. You can run the standalone console only on the physical computer on which the Administration Console is installed. Web-access mode, on the other hand, allows console users to access the Web Administration Console interface from a browser on a remote client computer.

To start the standalone Administration Console:

1. Open a command line shell.
2. Change to the *Oracle_home*`/ocsb60/admin_console` directory.
3. Enter the following command to start the Administration Console:

   ./start.sh *domain_home*

   where *domain_home* is the path to the domain configuration directory.

The standalone Administration Console interface appears.

Adding a Server to the Domain

After creating the domain, you add the servers you installed to the domain. You can add servers to a domain by using either JMX MBean operations or the Administration Console. After you add a server to the domain, you will be able to start the server, passing it the domain configuration. See "Starting the Server" for more information.

The following steps provide an overview of adding a server to the domain. For details on servers and on using the Administration Console interface, see *Oracle Communications Service Broker System Administrator's Guide*.

To add a server to the domain:

1. Open the Administration Console interface.

See "Starting the Administration Console" for more information on starting the Administration Console.

2. Click the **Switch to OFFLINE mode** icon at the top of the page.

   Offline mode enables you to make changes to the configuration before adding servers to the domain or when the servers are not running. In online mode, changes are pushed to the managed servers when you commit the changes.

3. In the navigation tree, expand the **OCSB** node.
4. Expand the **Domain Management** node.
5. Click **Servers**.
6. Click the **Add Server** icon.
7. Set the following fields in the **Add Server** dialog:

Table 5–2 Server Configuration Fields

Field	Description
Name	The name of the server. This name will be passed in the start command for the server. It must be unique across all domains. In a multi-domain deployment (that is, when using separate processing and signaling domains), server names should follow this format: ▪ **pn**_*number* for processing domain servers, where *number* is an integer. For example, **pn_1**, **pn_2**, and so on. ▪ **ssu**_*number* for signaling domain servers, where *number* is an integer. For example, **ssu_1**, **ssu_2**, and so on. It is possible to use custom server names, but then you must configure a mapping between the custom names to the expected names. See the discussion about mapping custom server names in *Oracle Communications Service Broker System Administrator's Guide* for more information. Format: Alpha-numeric characters. Case-sensitive. No white spaces.
Address	The host name or IP-address of the physical computer where the server runs. Format: alpha-numeric. IP-address format or DNS name format.
Port	The port number is deprecated and no longer used. Set this parameter to -1.
Admin Port	The IP port used by the server to communicate with the Administration Console. Format: numeric
JMX JRMP port	The port to use for Java Remote Method Protocol (JRMP) invocations to the server. Format: numeric
JMX Registry port	The port to use for the MBean Server on the server. Format: numeric

8. Click **OK**.

   The new server definition appears in the server list.

Repeat this procedure for each server you installed.

Table 5–3 outlines the corresponding server configurations for the example domain servers whose installation is demonstrated in Table 5–1.

Table 5–3 Example Server Configuration Settings

Server Name	Host	Port	Admin port	JMX Port	JMX Registry
pn_1	sb_01.telco.com	-1	8901	10003	10103
pn_2	sb_01.telco.com	-1	8901	10003	10104
pn_3	sb_02.telco.com	-1	8902	10004	10105
pn_4	sb_02.telco.com	-1	8902	10004	10106

You can now start the server, passing it the identity you configured in the domain, as described in "Starting the Server".

If you want to make configuration changes while the server is running, you can switch the Administration Console to online editing mode by clicking the **Switch to ONLINE mode** icon. If you intend to continue configuring the domain before starting the server (for example, to configure data store connectivity), you can keep the Administration Console interface in offline editing mode.

Deploying a Distributed Cluster

Servers within a domain and servers in different domains work together closely. Servers distribute events among each other inside the domain, and also across the domain boundaries. Therefore, all servers are grouped in a cluster.

Service Broker deploys Oracle Coherence for a distributed cluster. By default, servers in the cluster use IP multicasting for intra-server communication. IP multicasting configuration is set during domain creation, when you run the domain creation script. See "Creating the Domain" for more information.

For cluster deployments, if not using IP multicasting for intra-server communication, group the domains into clusters using well-known addresses. See the discussion on managing clusters in *Oracle Communications Service Broker System Administrator's Guide*.

Configuring Data Storage

Certain Service Broker features generate data that can be stored in persistent storage. These features include the Degraded Mode service and Subscriber Profile store.

Service Broker works with the following types of storage:

- Oracle Database 11g
- Oracle Berkeley DB (BDB) file-based storage

> **Note:** The Social Voice Communicator and VPN web applications only support Oracle Database storage for persistence.

Configuring data storage requires these steps:

1. Configure the database connection or local file storage location for each server.

   See "Using Oracle Berkeley DB File-Based Storage" or "Using Oracle Database 11g Storage" for more information.

2. Specify the persistence mechanism each feature should use by installing the appropriate persistence package.

   See "Specifying the Persistence Mechanism by Feature" for more information.

The following sections contain more information on these steps.

Using Oracle Berkeley DB File-Based Storage

Oracle Berkeley DB is a file-based storage mechanism. When configured to use Berkeley DB, each processing server has its own data directory in which it stores and accesses application data. Stored data is automatically synchronized across the servers in the cluster.

> **Note:** Multiple processing servers should not be configured to share a single data directory, even if the servers reside on the same physical host computer.

Additional deployment design considerations exist for using Berkeley DB in a cluster of domain servers. When the servers in a domain cluster start, they nominate a single server to act as the master node. The data on the master node serves as the primary copy of the data, from which the other servers in the domain cluster are synchronized. The nomination and synchronization processes among the nodes occur automatically.

However, if a server that is acting as the master node subsequently becomes unavailable, the remaining nodes must nominate a new node to serve as the master. To nominate a new master node, a majority of the servers in the domain cluster must be available. If less than a majority remain, a new master cannot be nominated and data storage and synchronization will fail. Note that this scenario will always occur in a cluster of only two servers.

You can avoid this issue by increasing the number of servers in your domain cluster. This reduces the likelihood that most servers in a cluster would be unavailable at a given time. For example, the cluster should use 3 servers instead of 2, 5 servers instead of 4, and so on.

To use Berkeley DB file-based storage, follow these steps:

1. Create or identify the directory on the server host in which you want Berkeley DB to store data. Oracle recommends that you use a local disk for better performance.
2. In the Administration Console, click the **Switch to OFFLINE mode** icon at the top of the page to switch the configuration mode to offline mode, if necessary.
3. Click the **Lock & Edit** icon.
4. In the navigation tree, expand the **OCSB** node and then **Domain Management**.
5. Expand the **Data Store** node.
6. Click the **Persistent Stores** node.
7. Click the **Berkeley DB Store** tab.
8. Click the **New** button.
9. Set the following fields in the dialog box:

Table 5–4 Berkeley DB Store Configuration Fields

Name	Description
Managed Server Name	The name of the domain server as specified in the domain configuration, such as **pn_1** or **pn_2**.
BDB Enabled	Set to **true** to direct the server to use Berkeley DB file-based storage for data persistence, or **false** to disable file-based storage.

Table 5–4 (Cont.) Berkeley DB Store Configuration Fields

Name	Description
BDB Environment Directory	The location of the data directory where the server should store persistent data. Use the full path to the directory, such as: **/home/oracle/OHOME1/bdbfiles**
Address (Host:Port)	The local IP address or host name and port of the server on which Berkeley DB service exchanges data synchronization messages with other servers in the domain cluster. By default, the port is 5001. For example: **192.168.1.10:5001**

10. Click **OK** to close the dialog box.
11. Repeat steps 8 through 10 for each managed server in your domain.
12. Click the **Commit** icon.

You may choose to continue working in offline edit mode, or switch to online mode if starting the managed servers. If the managed servers are running when you perform the configuration, you need to restart the servers to have the persistence changes take effect.

Using Oracle Database 11g Storage

To use Oracle Database 11g for Service Broker persistent storage, you must first prepare the database. You then configure Service Broker connections to the Oracle Database, as described in the following sections.

Preparing the Database

To prepare the database for Service Broker application storage:

1. In the database management system, create a user that Service Broker will use to access the database. The user should have permissions to create tables and add and modify data.

   You will enter this username and password when configuring the JDBC connection information (see "Configuring the Database Connection").

2. Run the SQL script needed for the Service Broker applications you are using.

   The SQL scripts create the database tables as required by the application. The scripts applicable to the Online Mediation Controller and Policy Controller are in the following directory:

   Oracle_home/**ocsb60/admin_console/scripts/database**

   The scripts are:

   - **degraded_mode_cdr_store.sql**: Configures the database for storing CDRs that are generated by Service Broker when it is operating in degraded mode. Use this script to configure the database for the Online Mediation Controller.
   - **subscriber_store.sql**: Configures the database for the subscriber profile storage. Use this script to configure the database for the Online Mediation Controller or Policy Controller.

For information on preparing the databases for the Service Broker VPN and Service Broker SVC products, see *Oracle Communications Service Broker VPN Implementation Guide* and *Oracle Communications Service Broker SVC Implementation Guide*.

After preparing the database, you can configure the database connection in the Service Broker domain.

Configuring the Database Connection

Service Broker connects to external databases using the settings defined in the JDBC driver connections in the domain configuration.

The credentials used to connect to the database are stored in the Service Broker credential store. The credential store is a secure credential storage mechanism used by Service Broker applications. For complete information on the Credential Store, see *Oracle Communications Service Broker System Administrator's Guide*.

Creating the Database Connection

To create the database connection by using the Administration Console:

1. If the Administration Console is not already in offline mode, click the **Switch to OFFLINE mode** icon at the top of the page.
2. Click the **Lock & Edit** icon.
3. Expand the **OCSB** node and then **Domain Management**.
4. Expand the **Data Store** node.
5. Click the **Credential Store** node.
6. In the **Credential Store** tab, use the fields in the **Password** group to add the database user credentials to the Credential Store, as follows
   a. Enter a name for the credential in the **Key** field. This value serves as an internal alias for the database user credential in the credential store.
   b. In the **Password** field, enter the password associated with the database user you created for Service Broker.
   c. Deselect the **One-way** check box so that the credential is created as a two-way credential. This allows Service Broker to retrieve the password from the credential store for submission to the external database management system.
   d. Click the **Set Password** button.
7. Click the **Persistent Stores** node.
8. Click the **JDBC Store** tab.
9. Click the **New** button.
10. Set the following fields in the dialog box:

Table 5–5 Database Connection Configuration Fields

Field	Description
URL	Connection URL to the database. For example: jdbc:oracle:thin:@//dbhost.example.com:1521/orcl
User	The username of the database user you created that Service Broker uses to access the Oracle Database.

Table 5–5 (Cont.) Database Connection Configuration Fields

Field	Description
Password Credential Key	The identifier for the database credential in the credential store. This value should match the value you entered in the **Key** field in the Password area of the Credential Store tab when adding the database credential to the Credential Store.
Connection Factory	The Java class used to create connections to the database. In most cases, this should be the default, **oracle.jdbc.pool.OracleDataSource**. If you have installed a custom or third party connection class, specify its name.
Initial Pool Size	The initial number of connections in the database connection pool.
Min Pool Size	Minimum number of connections in the connection pool.
Max Pool Size	Maximum number of connections in the connection pool.
Validation SQL	A SQL command used to verify that an open connection to the database remains valid over time. A connection can become invalid, or stale, when it is open for long time without activity. Typically, the command simply reads from an empty table. For example: `"select * from dual"`
Connection Pool Name	The name of the JDBC connection. By default, the Subscriber Profile feature attempts to access storage associated with a driver named **oracle_driver**. If you do not use **oracle_driver** for this value, you need to configure the Subscriber Profile feature to use the connection identified by your custom driver name. For better control of database access, Oracle recommends that you use different driver names for the Subscriber Profile and Degraded Mode features.

11. Click **OK** to save the configuration.

    The new connection instance appears in the list of drivers.

12. Repeat steps 9 through 11 for each Processing Server in your domain.

13. Click the **Commit** icon.

You may choose to continue working in offline edit mode, or switch to online mode if you intend to start the managed server. If the managed servers are running when you perform the configuration, you need to restart the servers to have the changes take effect.

Specifying the Persistence Mechanism by Feature

The particular storage mechanism used by a feature is determined by the persistence package that is installed in the domain. By default, the Degraded Mode and Subscriber Store features use Berkeley DB storage. (The VPN and SVC products can only use Oracle DB.)

To modify the storage mechanism for each feature, you need to replace its persistence package in the domain.

Note: Only a single persistence package for a given feature should be installed in the domain at a time. If you install a new persistence package, be sure to remove the existing package for that feature before you attempt to restart the servers in the domain.

The following instructions describe how to install and remove packages from the domain. If servers are running, they will need to be restarted for the changes to take effect. For details on package management, see the *Oracle Communications Service Broker System Administrator's Guide*.

Also, these instructions reflect the interface presented by the Web Administration Console. The steps are slightly different in the standalone Administration Console interface.

To apply a storage mechanism by Service Broker feature:

1. In the Administration Console, switch to offline mode.
2. Click the **Lock & Edit** icon
3. Expand the **OCSB** node.
4. Expand **Domain Management**.
5. Click **Packages**.
6. Click the **Install** button.
7. In the Install Bundle dialog box, click the **Browse** button.
8. In the Upload File dialog box, click **Browse** and navigate to the package directory:

   *oracle_home***/ocsb60/admin_console/modules**
9. Use the file chooser dialog box to select one of the following package files, depending on the persistence mechanism you want to use.

   For Subscriber Store persistence, select one of the following files:

   - oracle.ocsb.app.rcc.service.subscriber_store.providers.store.config_bdb.jar: Select this file to use Oracle Berkeley DB file-based storage. This is the default package for Subscriber Store persistence.
   - oracle.ocsb.app.rcc.service.subscriber_store.providers.store.config_db.jar: Select this file to use Oracle Database.

   For Degraded Mode persistence, select one of the following files:

   - oracle.ocsb.app.rcc.service.degraded_mode.persistence.bdb.jar: Select this file to use Oracle Berkeley DB file-based storage. This is the default package for Degraded Mode persistence.
   - oracle.ocsb.app.rcc.service.degraded_mode.persistence.database.jar: Select this file to use Oracle Database.
   - oracle.ocsb.app.rcc.service.degraded_mode.persistence.in_memory.jar: Select this file to use Oracle Coherence cache storage.
10. With the file you selected showing in the Upload File dialog box file field, click **Upload**.
11. Click the **OK** button in the File Uploaded dialog box.
12. Click **OK** in the Upload File dialog box.

13. Click **Install** in the Install Bundle dialog box.

    The package you selected appears in the package list.

14. Change the start level of the package you loaded to match the level of the following packages, depending on whether it is a Subscriber Profile or a Degraded Mode package:

    - oracle.ocsb.app.rcc.service.degraded_mode.core for the Degraded Mode persistence package
    - ocsb.app.rcc.service.subscriber_store.core for the Subscriber Profile persistence package

    To change the start level:

    a. Select the package in the list.

    b. Click the **Start Level** button.

    c. In the **Enter the start level** field, enter the new start level number. Enter 190 for the Subscriber Profile package, or 295 for the Degraded Mode package.

    d. Click **OK**.

15. Remove the previously installed persistence package (that is, the package that you are replacing) as follows:

    a. Select the package in the list.

    b. Click the **Uninstall** button.

Starting the Server

After you add a server to the domain and configured data storage, you can start the server. If a domain contains several servers, you need to start each server individually.

For Service Broker controller types that rely on persistent storage, you should configure data storage before attempting to start the server. See "Configuring Data Storage" for information on setting up data storage.

For hosted domains (that is, if you configured the domain files to be exposed by a Web server), you need to start the Web server process first using the **host.sh** script. For more information on hosted domains, see the *Oracle Communications Service Broker System Administrator's Guide*.

To start the server:

1. In a command line shell, change directories to the server directory. By default, the server is installed to the following directory:

   *Oracle_home***/ocsb60/managed_server**

2. Enter:

   ./start.sh *server_name domain_home***/initial.zip**

   where:

   server_name is the name you gave the server when you added the server to the domain.

   domain_home is the full path to the domain configuration directory where the **initial.zip** file is. The path may be in the form of a file location or URL. Specify the protocol as follows:

- For a Web-hosted domain, add the protocol prefix **http://** or **https://** to the path, depending on your security settings.
- If your domain configuration is accessed on a shared file system, add the protocol prefix **file://** to the path.

The following is an example of how to start a server, passing it the **pn_1** identity and the configuration bundle located on the local file system:

```
./start.sh pn_1 file:///home/oracle/sc_basic_domain/initial.zip
```

3. If using HTTPS, enter the keystore password when prompted.

Table 5–6 shows the start command and parameters for each server in the example domain introduced in Table 5–1. The start script is located in the server directory.

Note that the name of the server in the domain configuration, as shown in Table 5–3, corresponds to the name given as a parameter when starting the server.

Table 5–6 Example Server Start Commands

Server Directory	Server Start Command for a Non-hosted Domain	Server Start Command for a Hosted Domain
managed_1	**start.sh pn_1 file:../domain/initial.zip**	**start.sh pn_1 http://***ipaddress***:9001/initial.zip**
managed_2	**start.sh pn_2 file:../domain/initial.zip**	**start.sh pn_2 http://***ipaddress***:9001/initial.zip**
managed_3	**start.sh pn_3 file:../domain/initial.zip**	**start.sh pn_3 http://***ipaddress***:9001/initial.zip**
managed_4	**start.sh pn_4 file:../domain/initial.zip**	**start.sh pn_4 http://***ipaddress***:9001/initial.zip**

When you start multiple servers on a single physical host, Oracle recommends use of a separate command shell for each server.

See *Oracle Communications Service Broker System Administrator's Guide* for more information about starting servers.

6

Next Steps

This chapter provides a quick-step guide for the steps that you need to perform after successfully setting up an Oracle Communications Service Broker deployment.

Next Steps Overview

After completing all procedures in this document, the initial configuration of the deployment is complete. You can now configure the functional and operating behavior of the Service Broker implementation, along with additional features and advanced Service Broker topologies.

The following are the general steps that you perform after installing Service Broker:

- Start the SS7 process if your deployment requires connectivity to the SS7 network
- Perform additional setup tasks required for your particular Service Broker product implementation
- Complete the security configuration for the system
- Configure SNMP

Starting the SS7 Process

If your deployment requires connectivity to an SS7 network, then you need to start the SS7 process on the signaling servers.

After you added a signaling server to a domain, you can start the SS7 process on this server. If a domain contains several servers, you need to start the SS7 process on each server individually.

See the discussion about starting the SS7 process in *Oracle Communications Service Broker System Administrator's Guide* for more information.

Performing Additional Setup Tasks for Particular Service Broker Products

You may need to perform additional setup tasks required for your particular Service Broker product implementation. For example, for VPN and Social Voice Communicator implementations, you need to set up custom attribute-value pair (AVP) definitions used for charging purposes.

See the Service Broker implementation guide applicable to your system for specific setup steps.

When you deploy and configure components in the signaling tier and processing tier as required for the sub-product that you installed and for your specific solution, use

Oracle Communications Service Broker Signaling Domain Configuration Guide for information about configuring SSUs, and *Oracle Communications Service Broker Processing Domain Configuration Guide* for information about deploying and configuring IMs, SMs, and the OE.

Completing Security Configuration for the System

It is recommended that you complete the security configuration for the system, for example, by importing the security certificates that allow secure communications with remote telecommunication network.

See the discussion on configuring security in *Oracle Communications Service Broker System Administrator's Guide*.

Configuring SNMP

Configure SNMP settings for sending SNMP traps to external management systems.

See SNMP and other monitoring information in *Oracle Communications Service Broker System Administrator's Guide*.