Oracle® Communication Services Gatekeeper

System Backup and Restoration Guide

Release 4.0

June 2008

ORACLE

Oracle Communication Services Gatekeeper System Backup and Restoration Guide, Release 4.0

Copyright © 2007, 2008, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

CHAPTER 1

Introduction

A variety of events can lead to the failure of a server instance. Often one failure condition leads to another. Loss of power, hardware malfunction, operating system crashes, network partitions, or unexpected application behavior may each contribute to the failure of a server instance.

WebLogic Network Gatekeeper uses a highly clustered architecture as the basis for minimizing the impact of failure events. However, even in a clustered environment it is important to prepare for a sound recovery process in the event that an individual server or server machine fails.

This chapter summarizes WebLogic NetWork Gatekeeper failure prevention and recovery features, and describes the configuration artifacts that are required in order to restore different portions of a WebLogic Network Gatekeeper domain. The remaining sections in this guide describe how to back up WebLogic Network Gatekeeper configuration artifacts, and how to use those artifacts to restore the system in the event of a server failure.

Failure Prevention and Automatic Recovery Features

WebLogic Network Gatekeeper, and the underlying WebLogic Server platform, provide many features that protect against server failures. In a production system, all available features should be used in order to ensure uninterrupted service.

Overload Alarms

Network Gatekeeper's underlying WebLogic Server platform detects increases in system load that can affect deployed application performance and stability. WebLogic Server also allows administrators to configure failure prevention actions that occur automatically at predefined load

thresholds. Automatic overload protection helps you avoid failures that result from unanticipated levels of application traffic or resource utilization as indicated by:

- A workload manager's capacity being exceeded
- The HTTP session count increasing to a predefined threshold value
- Impending out of memory conditions

See Avoiding and Managing Overload in the WebLogic Server 10 documentation for more information.

Note: Each backwards compatible communication service in WebLogic Network Gatekeeper uses a pair of attributes, `OverloadPercentage` and `SevereOverloadPercentage`, that define the amount of load on the software module required to trigger an overload alarm. Always monitor for these alarms, and perform system throttling as necessary to avoid failures that could result from unanticipated levels of application traffic or resource utilization. See the *System Administrator's Guide* for more information on these communication services.

Redundancy and Failover for Clustered Services

Using multiple Access tier and Network tier servers in dedicated clusters increases the reliability and availability of your applications. Access tier clusters maintain no stateful information about applications, so the failure of a server does not result in any data loss. Network Gatekeeper also performs automated failover for servers within the Network tier. Any production installation must use the tiered configuration to protect against individual server failures. See Redundancy, Load Balancing, and High Availability in the *Architecture Overview* for more information.

Automatic Restart for Managed Servers

WebLogic Server self-health monitoring features improve the reliability and availability of server instances in a domain. Selected subsystems within each server instance monitor their health status based on criteria specific to the subsystem. (For example, the JMS subsystem monitors the condition of the JMS thread pool while the core server subsystem monitors default and user-defined execute queue statistics.) If an individual subsystem determines that it can no longer operate in a consistent and reliable manner, it registers its health state as "failed" with the host server.

Each WebLogic Server instance, in turn, checks the health state of its registered subsystems to determine its overall viability. If one or more of its critical subsystems have reached the FAILED

state, the server instance marks its own health state FAILED to indicate that it cannot reliably host an application.

When used in combination with Node Manager, server self-health monitoring enables you to automatically reboot servers that have failed. This improves the overall reliability of a domain, and requires no direct intervention from an administrator. For more information, see Using Node Manager to Control Servers in the WebLogic Server 10 documentation.

Managed Server Independence Mode

Managed Servers maintain a local copy of the domain configuration. When a Managed Server starts, it tries to contact the Administration Server to retrieve any changes to the domain configuration that were made since the Managed Server was last shut down. If a Managed Server cannot connect to the Administration Server during startup, it can use its locally-cached configuration information—this is the configuration that was current at the time of the Managed Server's most recent shutdown. A Managed Server that starts up without contacting its Administration Server to check for configuration updates is running in *Managed Server Independence (MSI)* mode. By default, MSI mode is enabled. See Replicating domain config files for Managed Server Independence in the WebLogic Server 10 documentation.

Automatic Migration of Failed Managed Servers

When using Linux or UNIX operating systems, you can use WebLogic Server's server migration feature to automatically start a candidate (backup) server if a Network tier server's machine fails or becomes partitioned from the network. The server migration feature uses node manager, in conjunction with the `wlsifconfig.sh` script, to automatically boot candidate servers using a floating IP address. Candidate servers are booted only if the primary server hosting a Network tier instance becomes unreachable. See Migration in the WebLogic Server 10 documentation for more information about using the server migration feature.

Geographic Redundancy for Regional Site Failures

In addition to server-level redundancy and failover capabilities, WebLogic Network Gatekeeper enables you to configure peer sites to protect against catastrophic failures, such as power outages, that can affect an entire domain. This enables you to failover from one geographical site to another, avoiding complete service outages. See Geographic Redundancy in the *Architecture Overview* for more information.

Overview of Configuration Artifacts

A WebLogic Network Gatekeeper deployment utilizes two basic categories of configuration information: domain-level configuration, and database configuration. The domain-level configuration consists of the artifacts used by the underlying WebLogic Server platform to configure the behavior of managed servers, clusters, security, and other resources deployed to clusters and servers within the domain. The primary domain-level configuration artifact is the `config.xml` file, stored in the *`domain-home`*`/config` directory. The `config.xml` file generally references additional configuration files beneath the `config` directory to configure additional domain resources such as JDBC and JMS.

In addition to the basic domain-level configuration of the WebLogic Server platform, WebLogic Network Gatekeeper stores some configuration for its core services in the form of database tables. This includes the routing configuration for backward-compatible communication services and PRM integration data. The database tables are shared across clustered instances of WebLogic Network Gatekeeper server instances. The database must be backed up at regular intervals to protect against data loss or corruption. An Oracle RAC deployment is also required for production installations, to provide redundancy and failover for the database configuration.

Both domain-level configuration backups and database backups may be required at different times in order to fully restore servers, or migrate server instances to new server hardware, in a WebLogic Network Gatekeeper installation.

Common Backup and Restoration Tasks

Maintaining system integrity requires that you make use of existing high availability and failover features, perform regular backups of configuration artifacts, and understand how to restore server instances or migrate servers onto viable hardware. These common tasks are summarized in Table 1-1.

Table 1-1 Common Backup and Restoration Tasks

Task	Links
Enable WebLogic Server platform reliability and recovery features.	• Avoiding and Managing Overload (WebLogic Server 10 documentation) • Replicating domain config files for Managed Server Independence (WebLogic Server 10 documentation) • Using Node Manager to Control Servers (WebLogic Server 10 documentation) • Migration (WebLogic Server 10 documentation)
Enable WebLogic Network Gatekeeper reliability and recovery features.	• Redundancy, Load Balancing, and High Availability (*Concepts and Architectural Overview*) • Geographic Redundancy (*Concepts and Architectural Overview*)
Backup WebLogic Server domain configuration.	• "Enabling Automatic Configuration Backups" on page 2-2 • "Storing the Domain Configuration Offline" on page 2-3 • "Backing Up Domain Security Data" on page 2-4 • "Backing Up Additional Configuration Files" on page 2-5

Table 1-1 Common Backup and Restoration Tasks

Task	Links
Backup WebLogic Network Gatekeeper database configuration.	• "Backing Up An Oracle 10g Single Instance Database" on page 4-2 • "Backing Up An Oracle 10g Single Instance Database" on page 4-2 • "Backing Up An Oracle 10g RAC Database" on page 4-4 • "Backing Up a MySQL Database" on page 4-5 • "Restoring the Database from Backup" on page 4-6
Restore a failed Access Tier or Network Tier server instance.	• "Restarting a Failed Administration Server" on page 3-1 • "Restarting Failed Access and Network Tier Servers" on page 3-3 • "Moving an Access or Network Tier Server to a Different Machine" on page 3-4

CHAPTER 2

Backing Up the Domain Configuration

The following sections provide information and procedures for backing up the primary configuration artifacts of a WebLogic Network Gatekeeper domain:

Overview of Domain Configuration Backup

Recovery from the failure of a server instance requires access to the domain's configuration and security data. WebLogic Network Gatekeeper can be configured to perform certain domain backups automatically. The administrator must also perform a manual backup of the domain configuration artifacts and store those backups outside of the actual domain directory.

By default, the Administration Server stores a domain's primary configuration data in a file called *domain_name*`/config/config.xml`, where *domain_name* is the root directory of the domain. The primary configuration file may reference additional configuration files for specific WebLogic Server services, such as JDBC and JMS. The configuration for specific services are stored in additional XML files in subdirectories of the *domain_name*`/config` directory, such as *domain_name*`/config/jms` and *domain_name*`/config/jdbc`.

The Administration Server can automatically archive multiple versions of the domain configuration (the entire *domain-name*/config directory). The configuration archives can be used for system restoration in cases where accidental configuration changes need to be reversed. For example, if an administrator accidentally removes a configured resource, the prior configuration can be restored by using the last automated backup.

The Administration Server stores a finite number of automated backups locally in *domain-name*\config. For this reason, automated domain backups are limited in their ability to guard against data corruption, such as a failed hard disk. Automated backups also do not preserve certain configuration data that are required for full domain restoration, such as LDAP repository data and server start-up scripts. BEA recommends that you also maintain multiple backup copies of the configuration and security offline, in a source control system, as described in "Backing Up Domain Security Data" on page 2-4.

Enabling Automatic Configuration Backups

Follow these steps to enable automatic domain configuration backups on the Administration Server for your domain:

1. Access the Administration Console for your domain.
2. In the left pane of the Administration Console, select the name of the domain.
3. In the right pane, click the Configuration->General tab.
4. In the Advanced Options bar, click Show.
5. In the Archive Configuration Count box, enter the maximum number of configuration file revisions to save.
6. Click Apply.

When you enable configuration archiving, the Administration Server automatically creates a configuration JAR file archive each time the Administrator uses the Activate Changes button in the Administration Console to change the active configuration. The JAR file contains a complete copy of the previous configuration (the complete contents of the *domain-name*\config directory). JAR file archive files are stored in the *domain-name*\configArchive directory. The files use the naming convention config-*number*.jar, where *number* is the sequential number of the archive.

When you save a change to a domain's configuration, the Administration Server saves the previous configuration in *domain-name*\configArchive\config.xml#*n*. Each time the

Administration Server saves a file in the `configArchive` directory, it increments the value of the *#n* suffix, up to a configurable number of copies—5 by default. Thereafter, each time you change the domain configuration:

- The archived files are rotated so that the newest file has a suffix with the highest number,
- The previous archived files are renamed with a lower number, and
- The oldest file is deleted.

Keep in mind that configuration archives are stored locally within the domain directory, and they may be overwritten according to the maximum number of revisions you selected. For these reasons, you must also create your own off-line archives of the domain configuration, as described in "Storing the Domain Configuration Offline" on page 2-3.

Storing the Domain Configuration Offline

Although automatic backups protect against accidental configuration changes, they do not protect against data loss caused by a failure of the hard disk that stores the domain configuration, or accidental deletion of the domain directory. To protect against these failures, you must also store a complete copy of the domain configuration offline, preferably in a source control system.

BEA recommends storing a copy of the domain configuration at regular intervals. For example, back up a new revision of the configuration when:

- you first deploy the production system
- you add or remove deployed applications
- the configuration is tuned for performance
- any other permanent change is made.

The domain configuration backup should contain the complete contents of the *domain_name*`/config` directory. For example:

```
cd ~/bea/user_projects/domains/mydomain
```

```
tar cvf domain-backup-06-17-2007.jar config
```

Store the new archive in a source control system, preserving earlier versions should you need to restore the domain configuration to an earlier point in time.

Backing Up Domain Security Data

The WebLogic Security service stores its configuration data `config.xml` file, and also in an LDAP repository and other files. As with the *`domain_name`*`/config` directory, a copy of the LDAP repository and security files should be stored offline each time you make a change to the security configuration.

Backing Up the WebLogic LDAP Repository

The default Authentication, Authorization, Role Mapper, and Credential Mapper providers that are installed with WebLogic Network Gatekeeper store their data in an LDAP server. Each WebLogic Network Gatekeeper server instance contains an embedded LDAP server. The Administration Server contains the master LDAP server, which is replicated on all Managed Servers. If any of your security realms use these installed providers, you should maintain an up-to-date backup of the following directory tree:

`domain_name``\servers\`*`adminServer`*`\data\ldap`

where *`domain_name`* is the domain's root directory and *`adminServer`* is the directory in which the Administration Server stores runtime and security data.

Each WebLogic Network Gatekeeper server has an LDAP directory, but you only need to back up the LDAP data on the Administration Server—the master LDAP server replicates the LDAP data from each Managed Server when updates to security data are made. WebLogic security providers cannot modify security data while the domain's Administration Server is unavailable. The LDAP repositories on Managed Servers are replicas and cannot be modified.

The `ldap`/`ldapfiles` subdirectory contains the data files for the LDAP server. The files in this directory contain user, group, group membership, policies, and role information. Other subdirectories under the `ldap` directory contain LDAP server message logs and data about replicated LDAP servers.

Do not update the configuration of a security provider while a backup of LDAP data is in progress. If a change is made—for instance, if an administrator adds a user—while you are backing up the `ldap` directory tree, the backups in the `ldapfiles` subdirectory could become inconsistent. If this does occur, consistent, but potentially out-of-date, LDAP backups are available.

Once a day, a server suspends write operations and creates its own backup of the LDAP data. It archives this backup in a `ZIP` file below the `ldap\backup` directory and then resumes write operations. This backup is guaranteed to be consistent, but it might not contain the latest security data.

For information about configuring the LDAP backup, see Configuring Backups for Embedded LDAP Server in the WebLogic Server 100 Documentation.

Backing Up SerializedSystemIni.dat and Security Certificates

All servers create a file named `SerializedSystemIni.dat` and place it in the *`domain_name`*`/security` directory. This file contains encrypted security data that must be present to boot the server. You must back up this file.

If you configured a server to use SSL, also back up the security certificates and keys. The location of these files is user-configurable.

Backing Up Additional Configuration Files

Certain additional files maintained at the operating system level can be helpful when recovering from a system failure. Consider backing up the following information as necessary for your system:

- Load Balancer configuration scripts. For example, any automated scripts used to configure load balancer pools and virtual IP addresses for the engine tier cluster, as well as NAT configuration settings.
- NTP client configuration scripts used to synchronize the system clocks of engine and data tier servers.
- Host configuration files for each WebLogic Network Gatekeeper machine (host names, virtual and real IP addresses for multihomed machines, IP routing table information).
- Managed Server start scripts. In a WebLogic Network Gatekeeper deployment, the start scripts used to boot Access and Network tier servers are generally customized to include configuration information such as JVM Garbage Collection parameters and other tuning parameters.

As with offline backups of the domain configuration, BEA recommends storing multiple copies of the above files in a source control repository.

CHAPTER 3

Restoring Failed Server Instances

The following sections provide information and procedures for restoring and moving failed servers in a WebLogic Network Gatekeeper domain:

Restarting a Failed Administration Server

When you restart a failed Administration Server, no special steps are required. Start the Administration Server as you normally would.

If the Administration Server shuts down while Managed Servers continue to run, you do not need to restart the Managed Servers that are already running in order to recover management of the domain. The procedure for recovering management of an active domain depends upon whether you can restart the Administration Server on the same machine it was running on when the domain was started.

Restarting an Administration Server on the Same Machine

If you restart the WebLogic Administration Server while Managed Servers continue to run, by default the Administration Server can discover the presence of the running Managed Servers.

Note: Make sure that the startup command or startup script does not include `-Dweblogic.management.discover=false`, which disables an Administration Server from discovering its running Managed Servers.

The root directory for the domain contains a file, `running-managed-servers.xml`, which contains a list of the Managed Servers in the domain and describes whether they are running or not. When the Administration Server restarts, it checks this file to determine which Managed Servers were under its control before it stopped running.

When a Managed Server is gracefully or forcefully shut down, its status in `running-managed-servers.xml` is updated to "not-running". When an Administration Server restarts, it does not try to discover Managed Servers with the "not-running" status. A Managed Server that stops running because of a system crash, or that was stopped by killing the JVM or the command prompt (shell) in which it was running, will still have the status "running' in `running-managed-servers.xml`. The Administration Server will attempt to discover them, and will throw an exception when it determines that the Managed Server is no longer running.

Restarting the Administration Server does not cause Managed Servers to update the configuration of static attributes. *Static attributes* are those that a server refers to only during its startup process. Servers instances must be restarted to take account of changes to static configuration attributes. Discovery of the Managed Servers only enables the Administration Server to monitor the Managed Servers or make runtime changes in attributes that can be configured while a server is running (dynamic attributes).

Restarting an Administration Server on Another Machine

If a machine crash prevents you from restarting the Administration Server on the same machine, you can recover management of the running Managed Servers as follows:

1. Install the WebLogic Network Gatekeeper software on the new administration machine (if this has not already been done).
2. Make your application files available to the new Administration Server by copying them from backups or by using a shared disk. Your application files should be available in the same relative location on the new file system as on the file system of the original Administration Server.
3. Make your configuration and security data available to the new administration machine by copying them from backups or by using a shared disk. For more information, refer to "Storing the Domain Configuration Offline" on page 2-3 and "Backing Up Security Data" on page 11-5.

4. Restart the Administration Server on the new machine.

   Make sure that the startup command or startup script does not include `-Dweblogic.management.discover=false`, which disables an Administration Server from discovering its running Managed Servers.

When the Administration Server starts, it communicates with the Managed Servers and informs them that the Administration Server is now running on a different IP address.

Restarting Failed Access and Network Tier Servers

If the machine on which the failed Managed Server runs can contact the Administration Server for the domain, simply restart the Managed Server manually or automatically using Node Manager. Note that you must configure Node Manager and the Managed Server to support automated restarts, as described in Using Node Manager to Control Servers in the WebLogic Server 10 documentation.

If the Managed Server cannot connect to the Administration Server during startup, it can retrieve its configuration by reading locally-cached configuration data. A Managed Server that starts in this way is running in Managed Server Independence (MSI) mode. For a description of MSI mode, and the files that a Managed Server must access to start up in MSI mode, see Replicating domain config files for Managed Server Independence in the WebLogic Server 10 documentation.

To start up a Managed Server in MSI mode:

1. Ensure that the following files are available in the Managed Server's root directory:
   - `msi-config.xml`.
   - `SerializedSystemIni.dat`
   - `boot.properties`

   If the files are not in the Managed Server's root directory:

   a. Copy the `config.xml` and `SerializedSystemIni.dat` file from the Administration Server's root directory (or from a backup) to the Managed Server's root directory.

   b. Rename the configuration file to `msi-config.xml`. When you start the server, it will use the copied configuration files.

   Note: Alternatively, use the `-Dweblogic.RootDirectory=`*`path`* startup option to specify a root directory that already contains these files.

2. Start the Managed Server at the command line or using a script.

The Managed Server will run in MSI mode until it is contacted by its Administration Server. For information about restarting the Administration Server in this scenario, see "Restarting a Failed Administration Server" on page 3-1.

Moving an Access or Network Tier Server to a Different Machine

Sometimes it is necessary to restart an Access Tier or Network tier server instance on a different machine. This situation might occur if a server machine's motherboard fails or if you upgrade to newer server hardware. When restarting a server on a new machine, maintain the same IP address and network configuration (DNS name, port numbers, and so forth) as the previous machine, if possible. If the network configuration remains unchanged, then the server can accept the existing domain configuration and you can restart it using the techniques described in "Restarting Failed Access and Network Tier Servers" on page 3-3.

If the network configuration is changed on the newer machine, then you must update the domain network configuration to match the server machine. Follow these steps before booting either an Access Tier or Network Tier server:

1. Access the Administration Console for the domain.
2. Click the Lock & Edit button to start a configuration session.
3. Select the Environment->Servers tab in the left pane.
4. Select the name of the server to update from the list of servers in the right pane.
5. Use the Configuration->General tab to modify the listen address and port settings to match the new server hardware.
6. Use the Protocols->Channels tab to modify any configured network channels to match the network settings of the new server hardware.
7. Click Activate Changes to apply your configuration changes.
8. Start the Managed Server on the new server hardware, using the instructions in "Restarting Failed Access and Network Tier Servers" on page 3-3.

In addition to the above steps, Network Tier servers require that you update the routing configuration for backwards compatible communication service plug-ins. This is required because the routing configuration uses Network Tier servers' IP addresses in the plug-in ID. Simply remove any old routes that involve the affected Network Tier server, and create new

routes based on the new server machine's network configuration. See Managing and Configuring the Plug-in Manager in the System Administrator's Guide for more information.

CHAPTER 4

Backing Up and Restoring the Network Gatekeeper Database

The following sections describe how to backup and restore WebLogic Network Gatekeeper installations:

Overview of Database Backups

In addition to the domain-level configuration backups described in "Backing Up the Domain Configuration" on page 2-1, you must backup the WebLogic Network Gatekeeper database (generally named `slee_db`). The `slee_db` maintains certain configuration information for Network Gatekeeper core services, such as routes defined for backwards compatible plug-ins, SLAs, and PRM data. The `slee_db` also stores the alarm configuration and generated alarms.

Although your RDBMS server software should maintain a replicated database for a production installation, BEA recommends backing up the full `slee_db` once daily, to an offline location. This provides an extra layer of protection against losing the domain-wide configuration stored in the database. Keep in mind that should you need to restore the `slee_db` from a backup, you will

lose any changes made to the configuration since the last scheduled backup. Because Network tier servers cache certain information from the `slee_db` tables, restoring the system from a backup of `slee_db` requires a full restart of the Network tier cluster to avoid inconsistencies between the cache and database tables.

The procedures for backing up the `slee_db` database are specific to the type of database you have deployed: a MySQL system is backed up differently than an Oracle 10g system. To a certain extent, system backup procedures also vary depending on the type of base system and specific configurations, such as additional equipment connected to the system, and so on. The sections that follow give general recommendations about performing backups, and provide information about the tools used to support these procedures.

Backing Up An Oracle 10g Single Instance Database

The sections that follow describe how to backup a single-instance Oracle 10g database. If you use an Oracle RAC database, see "Backing Up An Oracle 10g RAC Database" on page 4-4.

Configure the Backup Settings

Use Oracle RMAN to perform the backup. Different approaches to making a backup are described in the document *Oracle® Database Backup and Recovery Basics, 10g Release 1 (10.1)*. The backup and recovery scenarios below are based on the section, "Performing Disaster Recovery on Oracle Single instance using RMAN."

To configure Oracle for backup, the user performing the configuration must be logged in with database administrator privileges.

To make an efficient backup, the database must:

- Be running in ARCHIVELOG Mode.
- Use a Flash Recovery Area

In addition, the RMAN option CONTROLFILE AUTOBACKUP should be set to ON. The sections that follow provide more details.

ARCHIVELOG Mode

The database must be running in ARCHIVELOG Mode. This makes it possible to perform on-line backups of the database. See the section "Setting the Initial Database Archiving Mode" in chapter "Managing Archived Redo Logs" in *Oracle® Database Administrator's Guide 10g Release 1 (10.1)*.

Flash Recovery Area

The database should be configured to use the Flash Recovery Area. It will be used to store most of the backup and recovery-related files. See the section “Setting Up a Flash Recovery Area for RMAN” in chapter “Setting Up and Configuring Backup and Recovery” in *Oracle® Database Backup and Recovery Basics 10g Release 1 (10.1)*.

Auto Backup Control File

The RMAN configuration option, CONTROLFILE AUTOBACKUP, should be set to ON. This enables RMAN to make backups of the database Control File and Server Parameter File. See the section ‘Configuring Control File and Server Parameter File Autobackup” in chapter “Setting Up and Configuring Backup and Recovery” in *Oracle® Database Backup and Recovery Basics 10g Release 1 (10.1)*.

Perform the Backup

To perform the Oracle backup:

1. Login to the server executing the database.
2. Start the RMAN executable at the operating system using the `rman` command.
3. At the RMAN prompt, connect to the target database using the `connect target` command.
4. At the RMAN prompt, backup the database to the Flash Recovery Area using the command: `backup database plus archivelog;`

   This command returns a database identifier (DBID).
5. Keep a record of the database identifier, because it is required when performing the recovery procedure.
6. Use operating system-specific tools to copy the Oracle configuration files and password files to your permanent back-up storage location.
7. Use operating system-specific tools to copy the Flash Recovery Area to your permanent back-up storage location.

Backing Up An Oracle 10g RAC Database

The sections that follow describe how to backup an Oracle 10g RAC database. If you use an Oracle single instance database instead, see "Backing Up An Oracle 10g Single Instance Database" on page 4-2.

Configure Oracle RAC for Backup

Use Oracle RMAN to perform the backup. Different approaches to making backups are described in the document, *Oracle® Real Application Clusters Administrator's Guide 10g Release 1 (10.1)*. The backup and recovery scenarios described here are based on *Oracle® Real Application Clusters Administrator's Guide 10g Release 1 (10.1)*.

To configure Oracle for backup, the user performing the configuration must be logged in with database administrator privileges.

To make an efficient backup:

- Define the location of the snapshot control file.
- Configure the auto backup control file.
- Configure the archived redo logs.
- Configure the flash recovery area.

The sections that follow provide more details about these procedures.

Snapshot Control File Location

Configure the location of the snapshot control file using the instructions given in section "Configuring the RMAN Snapshot Control File Location" in chapter "Configuring Recovery Manager and Archiving" in *Oracle® Real Application Clusters Administrator's Guide 10g Release 1 (10.1)*.

Auto Backup Control File

Configure the RMAN Control file Autobackup feature according to section "Configuring the RMAN Control File Autobackup Feature" in chapter "Configuring Recovery Manager and Archiving" in *Oracle® Real Application Clusters Administrator's Guide 10g Release 1 (10.1)*.

Archived Redo Log

Configure the archive redo log according to the following sections in the chapter “Configuring Recovery Manager and Archiving” in Oracle® Real Application Clusters Administrator's Guide 10g Release 1 (10.1):

- “Managing Archived Redo Logs Using RMAN in Real Application Clusters”
- “Archived Redo Log File and Destination Conventions in RAC”
- “RMAN Archiving Configuration Scenarios”
- “Changing the Archiving Mode in Real Application Clusters”

Flash Recovery Area

Configure the database to use a Flash Recovery Area. This area will be used to store most of the backup and recovery-related files. See the section “Using a Flash Recovery Area in Real Application Clusters” in chapter “Managing Backup and Recovery” in *Oracle® Real Application Clusters Administrator's Guide 10g Release 1 (10.1).*

Perform the Backup

Follow the instructions given in section “Instance Recovery in Real Application Clusters” and section “RMAN Backup Scenarios for Real Application Clusters” in chapter “Managing Backup and Recovery” in *Oracle® Real Application Clusters Administrator's Guide 10g Release 1 (10.1).*

Use operating system-specific tools to copy the Oracle configuration files and password files to your permanent back-up storage location.

Keep a record of the database identifier, because it is required for performing the recovery procedure.

Backing Up a MySQL Database

To backup the MySQL database, first prepare the database by:

- Obtaining a read lock on all tables in slee_db, and
- Flushing all cached data to the database files.

After preparing the database, use one of the available MySQL backup scripts, such as mysqldump or mysqlhotcopy, to create an offline copy of the database. The exact tool and procedure you use

depends on the type of database you have created (InnoDB, MyISAM, ISAM). See Backup and Recovery in the MySQL Developer Zone to determine which backup utility and strategy best meets your needs.

You may also consider using the `--log-bin` option when starting MySQL to record database updates in a binary log file. The binary log acts as an incremental backup, and it can be applied to the a full database backup to restore the database to a more recent point in time.

Restoring the Database from Backup

The procedures for database restoration depend on whether the installation uses Oracle or MySQL as a database. If the installation uses Oracle see "Restoring a single instance Oracle 10g database" on page 4-6 for a single instance database or "Restoring an Oracle 10g RAC database" on page 4-7 for a RAC configuration.

Restoring a single instance Oracle 10g database

Although there are a number of options for restoring an Oracle database, RMAN is the preferred option for WebLogic Network Gatekeeper, as it is also the preferred option for making backups.

Note: This section does not describe how to restore the Oracle software. Refer to the Oracle documentation for information on how to re-install the database. It is important that the database be restored to the same host (the same DNS name or IP-address) arranged with the same directory structure as the original.

1. Shutdown all Network tier server instances prior to the restoration procedure, and do not restart them until the procedure has been performed.
2. After you have finished the installation of the Oracle database software, shutdown the database software and delete all data files and redo logs.
3. Delete the Control File and the Server Parameter File.
4. Using operating system specific tools, copy the Oracle configuration files and password files from your permanent backup storage location to the Oracle database.
5. Using operating system specific tools, copy the Flash Recovery Area from your permanent backup storage to the newly installed database.
6. Start the RMAN executable at the operating system command line using the command `rman`
7. At the RMAN prompt, connect to the target database using the command `connect target`

8. At the RMAN prompt, set the Database Identifier (DBID) using the command
   `SET DBID <DBID_RECORDED_DURING_BACKUP>`
   Use the value recorded during the backup procedure.
9. At the RMAN prompt, start the database in NOMOUNT state using the command:
   `STARTUP NOMOUNT`
10. At the RMAN prompt, restore the Server Parameter File from Autobackup using the command
    `RESTORE SPFILE FROM '<PATH_TO_AUTOBACKUP_OF_SPFILE>';`
11. At the RMAN prompt, restore the Control File from Autobackup using the command
    `RESTORE CONTROLFILE FROM AUTOBACKUP;`
12. At the RMAN prompt, shutdown the database using the command `shutdown`
13. At the RMAN prompt, start the database in MOUNT state using the command
    `STARTUP MOUNT`
14. At the RMAN prompt, restore the database using the command `RESTORE DATABASE;`
15. At the RMAN prompt, recover the database using the command `RECOVER DATABASE;`
16. At the RMAN prompt, open the database using the command
    `ALTER DATABASE OPEN RESETLOGS;`

At this point, the database is restored, and you can restart Network tier instances.

Restoring an Oracle 10g RAC database

Although there are a number of options for restoring an Oracle database, RMAN is the preferred option for WebLogic Network Gatekeeper, as it is also the preferred option for making backups. Read the instructions given in chapter “Managing Backup and Recovery” in *Oracle® Real Application Clusters Administrator's Guide 10g Release 1 (10.1)*.

Note: This section does not describe how to restore the Oracle software. Refer to the Oracle documentation for information on how to re-install the database. It is important that the database be restored to the same host (the same DNS name or IP-address) arranged with the same directory structure as the original.

Below is a summary of the steps to take when performing the restoration:

1. Shutdown all Network tier server instances prior to the restoration procedure, and do not restart them until the procedure has been performed.

2. After you have finished the installation of the Oracle RAC database software, shutdown the database software and delete all data files and redo logs.
3. Using operating system specific tools, copy the Oracle configuration files and password files from your permanent back-up storage location to the Oracle database.
4. Using operating system specific tools, copy the Flash Recovery Area from the permanent backup storage to the newly installed database.
5. At the RMAN prompt, connect to the target database using the command `connect target`
6. At the RMAN prompt, set the Database Identifier (DBID) using the command
   `SET DBID <DBID_RECORDED_DURING_BACKUP>`
   Use the value recorded during the backup procedure.
7. At the RMAN prompt, start the database in NOMOUNT state using the command:
   `STARTUP NOMOUNT`
8. At the RMAN prompt, restore the Server Parameter File from Autobackup using the command
   `RESTORE SPFILE FROM '<PATH_TO_AUTOBACKUP_OF_SPFILE>';`
9. At the RMAN prompt, restore the Control File from Autobackup using the command
   `RESTORE CONTROLFILE FROM AUTOBACKUP;`
10. At the RMAN prompt, shutdown the database using the command `shutdown`
11. At the RMAN prompt, start the database in MOUNT state using the command
    `STARTUP MOUNT`
12. At the RMAN prompt, restore the database using the command `RESTORE DATABASE;`
13. At the RMAN prompt, recover the database using the command `RECOVER DATABASE;`
14. At the RMAN prompt, open the database using the command
    `ALTER DATABASE OPEN RESETLOGS;`

At this point, the database is restored, and you can restart Network tier instances.

Restoring a MySQL Database

The tools and procedures used for backing up a MySQL database vary depending on the type of database you installed. For this reason, the instructions below provide only general guidelines for restoring a database. See Backup and Recovery in the MySQL Developer Zone for more detailed instructions and examples.

Before restoring a MySQL database, shut down all Network tier server instances. Do not restart the servers until the procedure has been performed.

Begin the restoration process by deleting any existing data files and reinstalling MySQL, if necessary. A MySQL backup file generally consists of a series of SQL statements used to re-create the database at a specific point in time. Simply start the MySQL process and apply the statements from the backup file. Alternately, your backup may consist of file system-level copies of the MySQL data files. In this case, restore the files to their original location (for example, `/usr/local/mysql/data`) before restarting the server.

After restoring from a backup and restarting the server, apply any incremental backups obtained from binary log files to bring the database up-to-date.

At this point, the database is restored and you can restart Network tier server instances.

APPENDIX A

WebLogic Network Gatekeeper Tables

The list that follows shows all the tables in that are created on first start-up of the WebLogic Network Gatekeeper database (except deployment-specific tables).

Note: As an optimization, it is possible to remove tables that not are used by communication services that are deployed in a particular installation. If this has happened, those tables will not exist and will not need to be backed up.

Standard 4.0 Tables

WLS_ACTIVE

budget_config_table

budget_service_state_table

mpcc_call_leg_session

mpcc_call_session

pl_cd_sip_addr_cor

pl_cd_sip_cor_sub

pl_cn_px30_osa_associator

pl_cn_px30_osa_info

pl_cn_sip_addr_cor

pl_cn_sip_cor_sub

pl_mms_mo_content_mms

pl_mms_mo_mms

pl_mms_mt_dr_mms

pl_mms_offline_notif

pl_mms_online_notif

pl_mpcc_media_notif

pl_osa_acc_connection

pl_osa_acc_gateway

pl_osa_acc_mapping

pl_osa_keystore

pl_osa_keystore_pwd

pl_push_pap_notification_info

pl_px30_ac_ui_call_info

pl_px30_ui_call_leg_info

pl_sms_mo_sms

pl_sms_offline_notif

pl_sms_online_notif

pl_sms_smpp_mo_sms

pl_sms_smpp_mt_sms

pl_tl_mlp_trigger_info

pl_tpc_inap

pol_app_irl

pol_sp_irl

pres_sip_not

pres_sip_sub

pres_sip_uri_map

prm_users_bc

shortcode_maping_store

slee_alarm

slee_charging

slee_db_locks

slee_instance_ids

slee_license_tps_check_state

slee_license_tps_log

slee_pl_mgr_config

slee_pl_mgr_plugin

slee_pl_mgr_routes

slee_snmp_config

slee_snmp_trap_receivers

slee_statistics_config

slee_statistics_data

slee_supported_statistics

tpc_call_participants_info

wlng_account_groups

wlng_app_accounts

wlng_app_instances

wlng_app_sessions

wlng_blob_configuration

wlng_configuration

wlng_mgmt_usergroups

wlng_mgmt_users

wlng_prm_sessions

wlng_slas

wlng_sp_accounts

wlng_subscr_contracts

Backwards Compatible Tables

If you are using backwards compatible listeners (CORBA-based) for alarms, EDRs, or CDRs, there are additional tables in the database. These tables are not included by default in the out-of-the-box version of Network Goalkeeper.

slee_alarm_config

slee_alarm_params

slee_alarm_severity_mapping

slee_charging_config

slee_charging_listeners

slee_edr_filter_properties

slee_edr_filters

slee_edr_listener_filters

slee_edr_listeners