Preface

Audience....................................................................................................................... vii
Downloading Oracle Communications Documentation............................................... vii
Related Documents........................................................................................................ vii

1 Starting and Stopping ASAP

Starting ASAP ................................................................................................................. 1-1
Starting All ASAP Servers ............................................................................................. 1-2
Starting the Control Server ............................................................................................ 1-4
Starting the ASAP Daemon ............................................................................................ 1-4
Starting ASAP Daemon Using a Start Command ......................................................... 1-5
Starting ASAP Daemon Using a Wrapping Script ....................................................... 1-5
Confirming that ASAP Started Successfully ................................................................ 1-5

Stopping ASAP ............................................................................................................ 1-6

2 Setting Up and Managing ASAP Security

Overview of Setting Up ASAP Security Features ....................................................... 2-1
Configuring WebLogic Server Security ....................................................................... 2-1
About ASAP WebLogic Users, Groups, Roles, and Methods .................................... 2-2
Understanding OCA Client Group Permissions ......................................................... 2-3
Configuring Virtual Network Operator Authorization for OCA Users ....................... 2-5
Configuring Authentication Providers for ASAP ...................................................... 2-6
Configuring a Primary Authentication Provider in WebLogic Server ....................... 2-6
Managing WebLogic Server ASAP User Security .................................................... 2-7
Configuring the WebLogic Server Change Password Utility Page ......................... 2-8
Setting WebLogic Server ASAP Password Policies .................................................. 2-9
Changing WebLogic Server ASAP User Passwords ................................................ 2-11
Disabling the Change Password Feature in the OCA Client .................................... 2-11
Managing Locked-out User Accounts ........................................................................ 2-13
Updating Methods Role Assigned to a Group or User in WebLogic ......................... 2-14

Configuring ASAP Server and Database Credential Security .................................. 2-15
About Credential Store Factory Wallet Secure Data Management ......................... 2-15
Setting up and Maintaining Secure Data Storage ..................................................... 2-15
Data Encryption ........................................................................................................... 2-15
Using the Credential Store Factory Wallet with ASAP Utilities and Scripts ........................................ 2-16
Changing Database Passwords in the Credential Store Factory Wallet ........................................... 2-16

**Configuring Security for Network Elements Communication** .................................................. 2-17
Understanding the Custom Secure Data Structure......................................................................... 2-17
Managing Custom Secure Data......................................................................................................... 2-18
  Setting up and Maintaining Secure Data Storage ......................................................................... 2-19
  Encrypting Data During Network Element Provisioning .............................................................. 2-19
  Understanding Key Distribution for Custom Secure Data............................................................ 2-19
  Caching Secure Data on Local Servers......................................................................................... 2-20
Securing Network Element Credentials with the Security Administration Tool............................ 2-20

**Additional Security Considerations** ............................................................................................ 2-22
Setting Secure Diagnostic Levels .................................................................................................. 2-22
  Setting the Network Element Dialog Diagnostic Configuration Parameter ................................ 2-22
  Setting Work Order Information Diagnostic Levels .................................................................... 2-23
Securing Java or State Table NEP or JNEP to NE Connection Implementations.............................. 2-24
Setting SRP or JSRP to SARM Security Properties ........................................................................ 2-24
Setting Security Between Servers................................................................................................... 2-24
Enabling Schema Validation for the JSRP JMS Interface................................................................. 2-24

3  **Monitoring and Managing ASAP**

**Overview of Monitoring and Managing ASAP** .......................................................................... 3-1

**Configuring System Events and Alarms Using Stored Procedures** ........................................... 3-2
  Configuring Alarm Centers for Alarm Notification and Escalation .............................................. 3-3
  Configuring System Alarms ............................................................................................................ 3-3
  Configuring System Events ............................................................................................................ 3-5
    Defining System Events ............................................................................................................. 3-5
    Setting Database Thresholds ..................................................................................................... 3-6
  Sample Alarm Program - admin.sh.............................................................................................. 3-6
  Sample Alarm Output .................................................................................................................... 3-7
  Sample Alarm Program - POTS Cartridge Sample Code.............................................................. 3-8

**Understanding Default System Events** .................................................................................... 3-8
  API System Events ....................................................................................................................... 3-8
  SARM System Events .................................................................................................................... 3-9
  Control Server Events ................................................................................................................... 3-10
  NEP System Events ..................................................................................................................... 3-10

**Configuring and Reading Log and Diagnostic Files** .................................................................. 3-11
  About Log Files ........................................................................................................................... 3-12
  About Diagnostic Files ................................................................................................................ 3-12
    Using ASAP Diagnostic Tools.................................................................................................... 3-13

**Configuring and Reading WebLogic-based Log and Diagnostic Files** ..................................... 3-13
  Defining Severity Levels ............................................................................................................. 3-13
  Configuring the Severity Levels .................................................................................................. 3-14
  Configuring the log4j.xml File...................................................................................................... 3-15
  Using the log4jAdmin Web Page.................................................................................................. 3-16
  Checking the Current Logging Levels ........................................................................................ 3-17

**Enabling Stored Procedure Error Messages** .......................................................................... 3-18
4 Improving ASAP Performance

About Improving ASAP Performance ................................................................. 4-1
Tuning ASAP Performance with Pre-tuned ASAP Configuration Files ............. 4-1
  About Pre-tuned ASAP Configurations ......................................................... 4-1
  Installing a Pre-tuned Configuration ........................................................... 4-2
  Using a Pre-tuned Configuration with a New ASAP Installation .................. 4-2
  Generating Pre-tuned Configuration Files ................................................... 4-2
  Merging Pre-tuned File Settings into an Existing Installation ...................... 4-2
  Example Pre-tuned Configuration Performance ........................................... 4-2
Troubleshooting and Monitoring ASAP Performance ........................................ 4-4
  Understanding Sysmon Output Files ............................................................ 4-4
  Troubleshooting the Connection Pool .......................................................... 4-4
  Troubleshooting the Message Queue ........................................................... 4-6
Manually Tuning ASAP Performance ............................................................... 4-7
  Tuning Guidelines ......................................................................................... 4-7
  Setting System Limits .................................................................................. 4-8
  Determining the Size of Your System ........................................................... 4-8
  Tuning Message Queue Guidelines ............................................................... 4-9
Tuning ASAP Server Message Queues ............................................................ 4-10
  Tuning JSRP and SRP Message Queues ....................................................... 4-10
  Tips for Tuning the SRP ............................................................................. 4-12
  Tuning SARM Message Queues ................................................................. 4-12
  Tips for Tuning the SARM .......................................................................... 4-16
  Tuning NEP Message Queues ...................................................................... 4-16
  Tips for Tuning NEP .................................................................................. 4-18
Other Performance Issues .............................................................................. 4-19
  Local Versus NFS-mounted File Systems for Diagnostic Files ................... 4-19
  Server Diagnostic Levels ............................................................................ 4-19
  Diagnostic Messages Output ....................................................................... 4-20
  Query Optimization .................................................................................... 4-20

5 Backing Up and Restoring ASAP Files and Data

About Backing Up and Restoring Files and Data .............................................. 5-1
  ASAP System Backup and Restore .............................................................. 5-1
  Database Backup and Recovery ................................................................. 5-1
  Database Backup Strategy .......................................................................... 5-2
  Database Backup in a Distributed Environment ......................................... 5-2
  ASAP WebLogic Server Domain Back Up .................................................. 5-2

6 Managing the Database and File System

Overview of Managing the Database and File System ...................................... 6-1
  Configuring Kernel and Database Initialization Parameters ....................... 6-1
About Monitoring Database Segment and File System Size ......................... 6-2
  Configuring File System Thresholds ......................................................... 6-3
  Configuring Database Thresholds ............................................................... 6-4
Database Management and Tuning Recommendations .................................... 6-5
Enabling Automated ASAP Database Administration Options ..................................................... 6-6
Purging the Database and File System ......................................................................................... 6-6
   Purging the Database .................................................................................................................. 6-6
   Sample Database Purge Script .................................................................................................. 6-8
   Sample Cron Script for Clearing Alarms, Events, and Process Information .................. 6-10
   Purging Test Systems .................................................................................................................... 6-12

7 Troubleshooting ASAP

   Overview of Troubleshooting ASAP ......................................................................................... 7-1
   Troubleshooting Checklist ........................................................................................................... 7-1
   Using Error Logs to Troubleshoot ASAP ................................................................................ 7-2
   Understanding Error-Message Syntax ....................................................................................... 7-2
   Collecting Diagnostic Information ............................................................................................. 7-2
   Common ASAP Problems and Solutions .................................................................................. 7-2
   Problem: ASAP Servers Do Not Start - Database Tablespaces Used Up ........................... 7-3
   Problem: ASAP Servers Do Not Start - Interfaces File Empty or Missing ......................... 7-3
   Problem: ASAP Servers Do Not Start - No Information In Interfaces File ....................... 7-3
   Problem: ASAP Servers Do Not Start - Wrong ASAP User Owner and Permissions .......... 7-4
   Problem: ASAP Servers Processes Do Not Start - Database Server Processes Used .......... 7-4
   Problem: ASAP Servers Processes Do Not Start - Database Server Sessions Used .......... 7-4
   Problem: ASAP Servers Do Not Start - Insufficient Server User Connections Defined ....... 7-4
   Problem: ASAP Servers Do Not Start - Insufficient Number of Threads ......................... 7-5
   Problem: Control Server Crashes - No Free Messages ......................................................... 7-5
   Problem: JNEP Server Does Not Start - Wrong Database Connection Information .......... 7-5
   Problem: JNEP Server Does Not Start - Invalid Server Port Numbers ................................... 7-6
   Problem: NEP Server Does Not Start - Problem with JNEP Java Process Start Script ....... 7-6
   Problem: WebLogic Server Fails to Detect Passive RAC Database During Failover .......... 7-6

Component Failure Scenarios ........................................................................................................ 7-7
   WebLogic Administration and Managed Server Failure and Recovery Scenarios ............. 7-7
   SRP Failure Scenario .................................................................................................................. 7-7
   SARM Failure Scenario ............................................................................................................. 7-7
   NEP Failure Scenario .................................................................................................................. 7-7
   Control Server Failure Scenario ............................................................................................... 7-8
   Database Failure Scenario ......................................................................................................... 7-8
   NE Unavailability Scenario ......................................................................................................... 7-8
   SRP and SARM Failure Scenario ............................................................................................... 7-8
   SARM and NEP Failure Scenario ............................................................................................... 7-8

Getting Help with ASAP Problems .............................................................................................. 7-8
   Before You Contact Global Support ....................................................................................... 7-8
   Reporting Problems ................................................................................................................... 7-9

A ASAP Directory Structure

   ASAP Directory Structure and Contents ................................................................................ A-1
Preface

This document describes how to start and stop Oracle Communications ASAP, configure security for ASAP, and manage and monitor ASAP. It also includes information about improving ASAP performance, managing the database and file system, backing up and restoring ASAP, and troubleshooting ASAP.

Audience

This document is intended for system administrators, system integrators, and other individuals who need to maintain and work with ASAP.

Downloading Oracle Communications Documentation

Oracle Communications ASAP documentation, is available from Oracle Help Center:

http://docs.oracle.com

Related Documents

For more information, see the following documents in the Oracle Communications ASAP documentation set:

- Oracle Communications ASAP Release Notes
- Oracle Communications ASAP Concepts
- Oracle Communications ASAP Service Request Translator User’s Guide
- Oracle Communications ASAP Order Control Application User’s Guide
- Oracle Communications ASAP Server Configuration Guide
- Oracle Communications ASAP Installation Guide
- Oracle Communications ASAP Cartridge Developer’s Guide
- Oracle Communications ASAP Security Guide
- Oracle Communications ASAP Developer’s Guide
Note: To download the Oracle Communications ASAP Developer's Guide from the Oracle software delivery Web site, you must select Oracle Communications Service Activation Developer Documentation Pack. You can visit the Oracle software delivery Web site at:

http://edelivery.oracle.com
Starting and Stopping ASAP

This chapter describes how to start and stop Oracle Communications ASAP.

Starting ASAP

The Control server creates a list of ASAP processes for each ASAP territory and system. The list enables the components of all ASAP systems in a territory to be specified and maintained in a single database.

In a particular ASAP implementation, there can be many territories. A territory is the geographical area served by the ASAP system. A Control server manages all applications and processes within the territory. Territories are generally mutually exclusive because there is no communication between ASAP systems in different territories.

The Control server determines each application’s details from the static database table `tbl_appl_proc`. This table specifies whether each application is a client or a server and specifies whether each application process should start automatically or manually, the sequence in which applications are to be started, and the diagnostic details of the process.

When ASAP is started, the Control servers on each machine in the network are started first. The primary Control server then either starts each ASAP application process individually (or sequentially) on its local machine, or instructs one of the secondary Control servers to start the application process on its machine. Every Control server determines the application details from a static database configuration table.

To shut down an ASAP application, you must submit a request to the Control server that manages the shutdown.

Using the Control server for all startup and shutdown activities provides a consistent approach to starting and stopping the system, and provides the ability to start ASAP processes on remote machines when ASAP is in a distributed environment.

Oracle recommends that you restart ASAP when:

- adding, updating, or deleting configuration information
- adding, updating, or deleting service definitions
- loading database schemas
- rolling back the system clock (for example, to adjust for daylight savings time).

Scripts are provided to start up and shut down an entire ASAP instance or individuals application components.
When the Control server is requested to start an ASAP application, the Control server performs the following steps:

- If the local Control server is not the same as the Control server for the application as defined in the application process table (that is, this application server should be started by a remote Control server), the local Control server opens a network connection to the remote Control server. The remote Control server then starts the application on the remote machine before returning a successful indication back to the requesting client.

- If the application server’s Control server is the local Control server, this local Control server first verifies that the UNIX program executable file for the application server is located in the $PROGRAMS directory and is executable. If it is executable, the Control server then instructs the fork agent process to spawn a child process which in turn overlays itself with the application executable. At this point, the application process starts up and the fork agent returns details of the application back to the Control server.

If the spawned application is a server, the parent Control server process goes into a retry loop and attempts to open a network connection to the newly created application process. If the connection to the application process is not established, a system event is issued and the Control server terminates the application server.

---

**Note:** You must define the server application name within the brackets [ ] in the header for each Server Configuration Parameters section in the ASAP.cfg file. You must define the server application name for each section, for example, the CTRL, SRP, SARM, NEP, and ADMIN. There can be no empty brackets [ ] in any of the sections in the Server Configuration Parameters, otherwise system errors occur.

---

### Starting All ASAP Servers

Use the `start_asap_sys` script located in the `ASAP_home/scripts` folder to start all ASAP servers (where `ASAP_home` is location of the ASAP installation).

After you start the WebLogic Server domain for ASAP and the ASAP database instance, you can call the `start_asap_sys` script to start the entire ASAP system.

**Note:** The `start_asap_sys` script must be started from the host that the primary Control server resides on. However, if you want to start ASAP remotely, R-shell into the host where the primary Control server resides.

This script does the following:

- Starts the ASAP Control server from the command line.
- Verifies that the Control server is running.
- Starts all configured ASAP application components in the ASAP system and territory in the sequence defined in the database.

**Usage**

```
start_asap_sys [-d] [-U] [-P] [-n] [-a] [-m] [SlaveCtrl ... ] [-C]
```

or
start_asap_sys -h

Table 1–1 lists and describes the arguments for the **start_asap_sys** script.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d</td>
<td>Development mode. If used, do not specify -U or -P options.</td>
</tr>
<tr>
<td>- U</td>
<td>The control database user ID.</td>
</tr>
<tr>
<td>- P</td>
<td>The password for the user ID.</td>
</tr>
<tr>
<td>- n</td>
<td>Indicates that no Control server should be started as a result of running this script, however, any other lapsed server should restart.</td>
</tr>
<tr>
<td>- a</td>
<td>An ASAP profile to be sourced before starting up Remote or Slave Control servers.</td>
</tr>
<tr>
<td>- m</td>
<td>The master Control server. This option is specified with -m. The -p option can be used to specify compatibility with a previous version of this script.</td>
</tr>
<tr>
<td>SlaveCtrl</td>
<td>The slave Control server(s) to start.</td>
</tr>
<tr>
<td>- C</td>
<td>Starts ASAP services concurrently, rather than sequentially.</td>
</tr>
<tr>
<td>- h</td>
<td>Displays command help.</td>
</tr>
</tbody>
</table>

When you start the ASAP servers, you must observe the following logic:

- If you do not specify the master Control server in the command, the script obtains the master server specification from the environment variable $MASTER_CONTROL. You specify the master Control server in the command line using the -m option otherwise the script uses the $MASTER_CONTROL environment variable.

- If you do not specify a Control server name, the slave Control servers identified in the environment variable $SLAVE_CONTROL_SERVERS are started. $SLAVE_CONTROL_SERVERS should provide the real names of the slave Control servers as follows:

  ```
  SLAVE_CONTROL_SERVERS="SlaveCtrl1[:Host1] ... SlaveCtrlN[:hostN]" export SLAVE_CONTROL_SERVERS
  ```

  If $SLAVE_CONTROL_SERVERS is not set, only the master Control server is started.

  For example:

  ```
  SLAVE_CONTROL_SERVERS="CTRLSVR1:192.168.10.251
  CTRLSVR1:192.168.10.252"
  ```

- If the host is not specified with a Slave Control server, the interfaces file $SYBASE/interfaces is searched to find the host for that server.

- If the host for a slave server is not given or cannot be determined by looking in the interfaces file, the Control server is not started.

- If an ASAP profile is specified in the command line, it is sourced before starting the Slave Control servers.

- If no ASAP profile is specified, the profile specified in the environment variable Environment_Profile is sourced before starting the Control servers.

- If Environment_Profile is not set, Slave Control servers are not started.
To start a Control server on a host, the ASAP user must have permission to use rsh to the host from the current host.

Starting the Control Server

Use the `start_control_sys` script located in the `ASAP_home/scripts` folder to start the ASAP Control server (where `ASAP_home` is location of the ASAP installation).

This script observes the same logic as `start_asap_sys`.

After you start the WebLogic Server domain for ASAP and the Oracle database, you can call the `start_control_sys` script to start the ASAP Control server(s). Application servers are not started with this script.

The `start_control_sys` script does the following:

- Starts the ASAP Control server. The standard input, output, and error from this startup are sent to the `ASAP.Console` file. The `ASAP.Console` file contains standard input, standard output, and errors that are sent to a console screen. The `ASAP.Console` file records any startup errors.
- Verifies that the Control server is running.

Usage

```
start_control_sys [-d] [ -U ] [ -P ] [ -a ] [ -m ] [ SlaveCtrl ... ]
```

Table 1–2 lists and describes the arguments for the `start_control_sys` script.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-U</td>
<td>The control database user ID.</td>
</tr>
<tr>
<td>-P</td>
<td>The password for the user ID.</td>
</tr>
<tr>
<td>-d</td>
<td>Development mode. If used, do not specify -U or -P options.</td>
</tr>
<tr>
<td>-a</td>
<td>An ASAP profile to be sourced before starting up Remote or Slave Control servers.</td>
</tr>
<tr>
<td>-m</td>
<td>The master Control server. This option is specified with “-m”; “-p” can be used to specify compatibility with a previous version of this script.</td>
</tr>
<tr>
<td>SlaveCtrl</td>
<td>The slave Control server(s) to start.</td>
</tr>
</tbody>
</table>

Starting the ASAP Daemon

When using an ASAP application that interfaces with the WebLogic Server (such as the Java SRP, SACT, or SADT), you must start the WebLogic Server first, and then start the specified ASAP applications.

---

**Note:** The `start_asap_sys` script also starts the ASAP Daemon.

---

When you use an application that uses the WebLogic Server, and that application calls an IO operation against the ASAP instance, the WebLogic Server sends a remote file request or a remote command to the ASAP daemon. All WebLogic Server IO operations against the ASAP instance are handled by the daemon server process.

You can start the ASAP daemon through using a start server command, or through a wrapping script. If starting the ASAP daemon using the start server command, start ASAP first.
Starting ASAP Daemon Using a Start Command
Oracle recommends that you start the ASAP daemon using the following command.

After you have started ASAP, type:

```
startc -d $DAEM
```

The ASAP daemon is started.

Starting ASAP Daemon Using a Wrapping Script
You can optionally use the following script to start the ASAP daemon server:

```
asapd -start -d | -password control_password -port port [-asap_base asap_base_dir] [-sybase sybase_dir]
```

Use the following script to stop the ASAP daemon server:

```
asapd -stop -d | -password control_password -url host:port
```

The following command displays help information (see Table 1–3):

```
asapd -h
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d</td>
<td>Used in development mode only.</td>
</tr>
<tr>
<td>-port=[port number]</td>
<td>Specifies a port number on which the ASAP daemon listens for requests from the WebLogic applications in the current ASAP instance. On the WebLogic Server, the daemon client configuration must contain an identical port number. In other words, when starting an ASAP Daemon server, the port number must be identical to the port that is referenced in the WebLogic Server jmx_ connector descriptor.</td>
</tr>
<tr>
<td>-password=[control password]</td>
<td>This optional argument specifies the control password for authentication purposes. In a development environment, if the password is missing, the script attempts to obtain the password from the environment.</td>
</tr>
<tr>
<td>-asap_base=[ASAP base directory]</td>
<td>This optional argument specifies the ASAP base path that the current ASAP instance works on. In a development environment, if the base path is missing, the script attempts to obtain the path from the environment.</td>
</tr>
<tr>
<td>-sybase=Sybase interfaces directory</td>
<td>This optional argument specifies the parent directory of the Sybase interfaces file used by the current ASAP instance. In a development environment, if this reference is not specified, the script will attempt to obtain the path from the environment.</td>
</tr>
<tr>
<td>help</td>
<td>Displays a help screen.</td>
</tr>
</tbody>
</table>

For more information on configuring the ASAP daemon, refer to *ASAP Developer’s Guide*.

Confirming that ASAP Started Successfully
To verify the status of each server process, use the following procedure:

1. From a UNIX terminal, source the `ASAP_home/Environment_Profile` (where `ASAP_home` is location of the ASAP installation).
2. Enter the following command:

```
status
```

```
**** ASAP Application Status ****

<table>
<thead>
<tr>
<th>#</th>
<th>CPU</th>
<th>PID</th>
<th>Program</th>
<th>Application</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0:01</td>
<td>796</td>
<td>$ASAP_BASE/programs/ctrl_svr</td>
<td>CTRLdc2</td>
<td>LOCAL</td>
</tr>
<tr>
<td>2</td>
<td>0:01</td>
<td>900</td>
<td>$ASAP_BASE/programs/fork_agent</td>
<td>CTRLdc2</td>
<td>LOCAL</td>
</tr>
<tr>
<td>3</td>
<td>0:01</td>
<td>978</td>
<td>$ASAP_BASE/programs/admn_svr</td>
<td>ADM_dc2</td>
<td>LOCAL</td>
</tr>
<tr>
<td>4</td>
<td>0:01</td>
<td>1019</td>
<td>$ASAP_BASE/programs/srp_emul</td>
<td>SRP_dc2</td>
<td>LOCAL</td>
</tr>
<tr>
<td>5</td>
<td>0:05</td>
<td>993</td>
<td>java</td>
<td>DAEMdc2</td>
<td>LOCAL</td>
</tr>
<tr>
<td>6</td>
<td>0:11</td>
<td>945</td>
<td>java</td>
<td>JNEP_dc2</td>
<td>LOCAL</td>
</tr>
<tr>
<td>7</td>
<td>0:02</td>
<td>960</td>
<td>$ASAP_BASE/programs/asc_nep</td>
<td>NEP_dc2</td>
<td>LOCAL</td>
</tr>
<tr>
<td>8</td>
<td>0:02</td>
<td>984</td>
<td>$ASAP_BASE/programs/sarm</td>
<td>SARMdc2</td>
<td>LOCAL</td>
</tr>
</tbody>
</table>

**** End of Application Status ****

3. Verify that all the server processes you wanted to start appear in the output from the `status` command.

**Stopping ASAP**

To shut down ASAP, a script called `stop_asap_sys` is provided in the `ASAP_home/scripts` folder (where `ASAP_home` is location of the ASAP installation). This script terminates the ASAP applications in the inverse order to which they were started.

In the distributed environment, the Control servers must be stopped separately on each machine in the network. This is supported by the `stop_asap_sys` script.
Setting Up and Managing ASAP Security

This chapter describes the security features for Oracle Communications ASAP.

Overview of Setting Up ASAP Security Features

ASAP security is designed to provide maximum confidentiality and data integrity, while ensuring on-demand access to services for authorized users. ASAP security is designed for three essential functions: managing ASAP WebLogic-based users, securing data, and protecting diagnostics files.

ASAP provides these security functions in the following locations:

- **Oracle WebLogic Server security**
  
  The WebLogic Server contains default users, groups, and roles that support the various WebLogic-based ASAP functionality, like the Order Control Application (OCA) clients, the Service Activation Configuration Tool (SACT), the Service Activation Deployment Tool (SADT). Java Service Request Processor (JSRP) clients also need WebLogic Server credentials to connect to the JSRP interfaces.

- **ASAP server and database credential security**
  
  The ASAP environment contains a credential store factory (CSF) wallet that stores the ASAP schema user names and passwords, and the WebLogic Server user name and password. These credentials are called class A secure data. Each ASAP server can use this wallet to obtain these credentials. The CSF wallet provides both secure storage and encryption for these credentials.

- **Network Element (NE) credential security for Network Element Processor (NEP) to NE communication**
  
  The ASAP Control database stores credentials required for NEPs to access NEs. These credentials are called class B secure data. You can use a security tool to add, change, or delete credential information, and also to enable encryption.

- **ASAP system configuration parameters**
  
  Some ASAP system configuration parameters can have a significant impact on security. Parameters settings, such as diagnostic levels and server security attributes should be configured correctly to ensure data security.

Configuring WebLogic Server Security

ASAP uses the embedded Lightweight Directory Access Protocol (LDAP) server included with the WebLogic Server software to manage default ASAP users, groups,
roles, and methods. For more information on this embedded LDAP sever, see the WebLogic Server documentation.

About ASAP WebLogic Users, Groups, Roles, and Methods

User security is managed with the WebLogic Server. For more information, see

http://docs.oracle.com/middleware/1213/wls/wls-secure.htm

The WebLogic Server provides the following features:

■ ASAP OCA client users can change their passwords. For more information, refer to the ASAP Order Control Application User’s Guide.

■ ASAP Administrators can use the WebLogic Administration Console to add, delete, or edit users or user groups.

■ ASAP Administrators can use the WebLogic Administration Console to assign or unassign permissions to users or user groups.

The administrator uses security roles, method names, and principal names to configure permissions using the WebLogic Administration Console.

For information on migrating security from a previous version of ASAP, refer to the ASAP Installation Guide.

You can create, delete, and modify ASAP users and groups in WebLogic Server. For more information, see

http://docs.oracle.com/middleware/1213/wls/wls-secure.htm

Note: ASAP requires that WebLogic user passwords have eight characters or more.

Table 2–1 lists and describes the default ASAP users.

<table>
<thead>
<tr>
<th>Default Users</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WebLogic_Admin</strong></td>
<td>This is the WebLogic Server administrator account, where WebLogic_admin is the user name you selected when you created your WebLogic Server.</td>
</tr>
<tr>
<td><strong>ASAP_admin</strong></td>
<td>This is the default OCA client user. This user can create OCA orders, manage fallout orders, and view audit log reports in the OCA client. For more information about OCA permissions for this user, see &quot;Understanding OCA Client Group Permissions&quot;. This user is a member of the ASAP_VNOS and ASAP_Operators groups.</td>
</tr>
<tr>
<td><strong>ASAP_monitor</strong></td>
<td>This is a default OCA client user that can be used to view OCA audit logs and reports. For more information about OCA permissions for this user, see &quot;Understanding OCA Client Group Permissions&quot;. This user is a member of the ASAP_Guests group.</td>
</tr>
<tr>
<td><strong>ASAP_operator</strong></td>
<td>This is a default OCA client user. This user can create OCA orders, manage fallout orders, and view audit log reports in the OCA client. For more information about OCA permissions for this user, see &quot;Understanding OCA Client Group Permissions&quot;. This user is a member of the ASAP_Operators group.</td>
</tr>
</tbody>
</table>
Table 2–2 lists and describes the default ASAP groups.

### Table 2–2 Default ASAP WebLogic Groups

<table>
<thead>
<tr>
<th>Default Groups</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdminChannelUsers</td>
<td>AdminChannelUsers can access the admin channel.</td>
</tr>
<tr>
<td>Administrators</td>
<td>Administrators can view and modify all resource attributes and start and stop servers.</td>
</tr>
<tr>
<td>AppTesters</td>
<td>AppTesters group.</td>
</tr>
<tr>
<td>ASAP_Guests</td>
<td>Allows an OCA user to login to OCA and access various methods.</td>
</tr>
<tr>
<td>ASAP_Monitors</td>
<td>Allows an OCA user to view OCA audit logs and reports in the OCA client.</td>
</tr>
<tr>
<td>ASAP_Operators</td>
<td>Allows an OCA user to creates orders, manage fallout orders, and view audit logs and reports in the OCA client.</td>
</tr>
<tr>
<td>ASAP_VNOS</td>
<td>The virtual network operator (VNO) group ID must be included as a member of ASAP_VNOS group. The group will be the parent of all VNO groups. This group enables the OCA client to identify and recognize a group ID as VNO group ID and therefore authorize a user as VNO user.</td>
</tr>
<tr>
<td>ASAP_WS_USERS_GROUP</td>
<td>Group to use role ASAP_WS_USERS.</td>
</tr>
<tr>
<td>Cartridge_Management_WebService</td>
<td>Group to use role CARTRIDGE_MANAGEMENT_WEBSERVICE.</td>
</tr>
<tr>
<td>CrossDomainConnectors</td>
<td>CrossDomainConnectors can make inter-domain calls from foreign domains.</td>
</tr>
<tr>
<td>Deployers</td>
<td>Deployers can view all resource attributes and deploy applications. DefaultAuthenticator.</td>
</tr>
<tr>
<td>Monitors</td>
<td>Monitors can view and modify all resource attributes and perform operations not restricted by roles. DefaultAuthenticator.</td>
</tr>
<tr>
<td>Operators</td>
<td>Operators can view and modify all resource attributes and perform server lifecycle operations. DefaultAuthenticator.</td>
</tr>
<tr>
<td>OracleSystemGroup</td>
<td>Oracle application software system group. DefaultAuthenticator.</td>
</tr>
</tbody>
</table>

Understanding OCA Client Group Permissions

Table 2–3 list and describes the ASAP permissions for OCA client groups.
### Table 2–3  WebLogic Permissions for the OCA Client Groups

<table>
<thead>
<tr>
<th>Method</th>
<th>ASAP_Operator</th>
<th>ASAP_Monitor</th>
<th>ASAP_VNOS</th>
<th>Enables Users to…</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA_Order_Control_Application</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Access the OCA client.</td>
</tr>
<tr>
<td>SA_View_Audit_Log</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>Use the audit log query function.</td>
</tr>
<tr>
<td>SA_Order_Entry</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Use the order entry function.</td>
</tr>
<tr>
<td>SA_Order_Management</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>Use the order management features. This includes the Work Order Query window.</td>
</tr>
<tr>
<td>SA_Abort_CSDLs</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Abort Common Service Description Layers (CSDLs) in the Work Order Details window.</td>
</tr>
<tr>
<td>SA_Add_CSDL_Parameters</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Add CSDL parameters in the Work Order Details window and access the CSDL Parameter dialog box.</td>
</tr>
<tr>
<td>SA_Delete_CSDL_Parameters</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Delete an optional CSDL parameter from the Work Order Details window.</td>
</tr>
<tr>
<td>SA_Edit_Global_Macros</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Override a global parameter to make it a local CSDL parameter and change the value in the Global Parameter dialog box.</td>
</tr>
<tr>
<td>SA_Edit_CSDL_Parameters</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Modify CSDL parameter values in the Input dialog box.</td>
</tr>
<tr>
<td>SA_Insert_CSDLs</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Add CSDLs to work orders (WOs) using the Select CSDLs dialog box.</td>
</tr>
<tr>
<td>SA_Release_Work_Order_Change_Due_Date</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Release a WO to the Provisioning queue with a changed due date in the Release dialog box.</td>
</tr>
<tr>
<td>SA_Resequence_CSDLs</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Correct the CSDL sequence using the Resequence – CSDL dialog box.</td>
</tr>
<tr>
<td>SA_Retry_CSDLs</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Retry individual or failed CSDLs in the Work Order Details window.</td>
</tr>
<tr>
<td>SA_Abort_Work_Orders</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Abort WOs in the Work Order Details window.</td>
</tr>
<tr>
<td>SA_Cancel_Work_Orders</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Cancel WOs from the Work Order Query window.</td>
</tr>
<tr>
<td>SA_Delete_Work_Orders</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Delete WOs from the Work Order Query window.</td>
</tr>
<tr>
<td>SA_View_Work_Order_Details</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>Query WOs and view their details in the Work Order Details window.</td>
</tr>
<tr>
<td>SA_Release_Work_Orders</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>Release WOs for provisioning from the Work Order Details window.</td>
</tr>
</tbody>
</table>
Configuring Virtual Network Operator Authorization for OCA Users

The ASAP administrator should create WebLogic Server user accounts for VNO fallout managers and include their user login names under one of VNO groups. The login name of any VNO user must be a member of one or more VNO groups and one ASAP group. Below is an example of "User_A" created as a VNO operator for the "Telco 1" group and "User_B" created as a VNO monitor for "Telco 1" and "Telco 2" groups.

Table 2–4 lists and describes possible VNO user and group settings.

<table>
<thead>
<tr>
<th>Login Name</th>
<th>Is a Member of</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>User_A</td>
<td>is a member of…</td>
<td>ASAP_Operators</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Telco 1</td>
</tr>
<tr>
<td>User_B</td>
<td>is a member of…</td>
<td>ASAP_Monitors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Telco 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Telco 2</td>
</tr>
</tbody>
</table>

Do the following:

- Create VNO groups with unique names and add VNO groups as members to ASAP_VNOS group.
- Add OCA user login names as members to one or more VNO groups. Ordinarily, an OCA user is added to one VNO group.
- Add each OCA user’s login name as a member to one ASAP group (ASAP_Operators or ASAP_Monitors).

VNO functionality is available to divide OCA user groups into geographic regions using the VNO_ID_DEFAULT and VNO_ID_STRIP parameters of the ASAP.cfg file. Refer to the discussion about the Service Request Processor (SRP) Emulator Server Configuration Parameters in the ASAP Server Configuration Guide.
Configuring Authentication Providers for ASAP

During the ASAP installation process, the ASAP installer creates default ASAP users, groups, roles, and methods in the embedded LDAP authentication provider included with WebLogic Server. You can use this authentication provider to configure the default ASAP users, groups, roles, and methods, or add, delete, or modify your own users, groups, roles, and methods. You can also integrate an external LDAP server for ASAP users, groups, roles, and methods to your WebLogic Server.

---

**Note:** You must ensure that the ASAP_VNO group name is defined in the external LDAP server.

---

You must perform the following steps to integrate an external LDAP server with ASAP:

2. Configuring a Primary Authentication Provider in WebLogic Server

**Configuring an External Lightweight Directory Access Protocol Server**

To configure an external LDAP server for use with ASAP, use the following procedure:

1. Create an Administrators group in the LDAP server.
2. Create a user in LDAP that can be used as an administrator login for Oracle WebLogic Server.
3. Add the newly created user to the Administrators group.
4. Create the following user groups for ASAP:
   - **ASAP_Guests**
   - **ASAP_Monitors**
   - **ASAP_Operators**
   - **ASAP_VNOS**
   - **ASAP_WS_USERS_GROUP**
   - **Cartridge_Management_WebService**
   - **everyone**

   **Note:** You may provide custom group names in the LDAP server corresponding to the seven groups mentioned in step 4.

5. Create users in the LDAP server and add them to the groups created in step 4.

For detailed instructions on configuring LDAP, see the LDAP documentation specific to your LDAP Authentication provider.

**Configuring a Primary Authentication Provider in WebLogic Server**

To configure the external LDAP server in Oracle WebLogic Server:

1. Configure the following authentication providers on the WebLogic Administration Console:
   - Default Authentication provider
LDAP Authentication provider
See the WebLogic Server documentation for information on configuring authentication providers.

2. Set the control flag as follows:
   - For Default Authenticator provider, set **SUFFICIENT**.
   - For LDAP Authentication provider, set **REQUIRED**.
See the WebLogic Server documentation for information on configuring authentication providers.

3. Reorder the authentication providers.

---

**Note:** Ensure that the Default Authentication provider is above the LDAP Authentication provider.

See the WebLogic Server documentation for information on reordering authentication providers.

---

### Managing WebLogic Server ASAP User Security

These procedures apply to user security that is maintained using WebLogic Server and ASAP, and assume that `myrealm` is the only active security realm. These procedures do not support other realms supported by WebLogic Server, such as the LDAP realm. When an administrator configures WebLogic Server with security realms other than `myrealm`, all features described in this section are disabled, including the change password menu in the OCA client.

ASAP administrators can configure user password policies through the WebLogic Administration Console and the Password Policy Utility page. Password policies defined in the WebLogic Administration Console include:

- Minimum password length
- Lockout enabled
- Lockout duration
- Reset duration
- Lockout cache

Password policies defined in the Password Policy Utility page include:

- Password aging
- Password expiration warning period
- Enabling/disabling password policies

If your ASAP installation includes the OCA client, observe the restrictions described in Table 2–5.

---

**Table 2–5  ASAP Client Password Restrictions**

<table>
<thead>
<tr>
<th>Feature</th>
<th>OCA Client</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password change</td>
<td>Supported</td>
<td>If the administrator has enabled password policies, users must change passwords in the OCA client.</td>
</tr>
</tbody>
</table>
The procedure for changing end user passwords is contained in the *ASAP Order Control Application User’s Guide*.

### Configuring the WebLogic Server Change Password Utility Page

**Note:** Secure Shell (SSH) must be enabled on the WebLogic Server in order for the **Password Policy** and **Change Password Utility** Java server pages (JSPs) to be reachable.

To enable SSH on the WebLogic Server:

1. Log in to the **WebLogic Administration Console**.
2. Click **Lock & Edit** if not already selected. See the WebLogic Server online Help for more information.
3. In the **Domain Structure**, click on the domain.
4. On the **Control** tab, select **Servers**.
5. In the **Servers** table, click the name of the server to be configured.
6. Select the **SSL Listen Port Enabled** checkbox.
7. Provide an appropriate SSL listen port number in the **SSL Listen Port** field (that is `BEA_SSL_PORT` from the Installation Values in the *ASAP Installation Guide*). Ensure that this is not the same as the server listen port (that is `BEA_PORT`).
8. Click **Release Configuration**.

### Table 2–5 (Cont.) ASAP Client Password Restrictions

<table>
<thead>
<tr>
<th>Feature</th>
<th>OCA Client</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password change at first login</td>
<td>Supported</td>
<td>If the administrator has enabled password policies, users must change passwords in the OCA client.</td>
</tr>
<tr>
<td>Password length</td>
<td>Supported</td>
<td>Passwords must be between 6 and 21 characters in length.</td>
</tr>
<tr>
<td>Password aging</td>
<td>Supported</td>
<td>If the administrator has enabled password policies, users must change passwords in the OCA client.</td>
</tr>
<tr>
<td>Password syntax</td>
<td>Supported</td>
<td>Password syntax.</td>
</tr>
<tr>
<td>Reuse of previously-used passwords</td>
<td>Supported</td>
<td>Enforced when the user specifies a new password in the OCA client.</td>
</tr>
<tr>
<td>Lockout features</td>
<td>Supported</td>
<td>Lockout features.</td>
</tr>
</tbody>
</table>
The Password Policy page and Change Password Utility page are JSPs accessed through a web browser at the following URLs:
- https://hostname:BEA_SSL_PORT/security/PasswordPolicy.jsp
- https://hostname:BEA_SSL_PORT/security/ChangePassword.jsp

Note that the SSL Listen Port (configured above) is used, not the BEA_PORT value.

Note: You must configure the BEA_WLS_HOST and BEA_WLS_PORT variables of these JSPs in ASAP.cfg. The ASAP installer populates these variables automatically during the installation process.

Setting WebLogic Server ASAP Password Policies

Administrators must set password policies using both the WebLogic Administration Console and the Password Policy page.

To define lockout conditions:

1. In the WebLogic Administration Console, click Lock & Edit if not already selected. See the WebLogic Server online Help for more information.

2. In the Domain Structure panel of the Change Center in the WebLogic Administration Console, click Security Realms.

   The Summary of Security Realms screen appears.

3. In the Realms list, click the name of the security realm to be configured.

   The Settings screen for the selected realm appears.

4. Select the Configuration tab, and the User Lockout sub-tab to access user lock-out settings.

5. On the User Lockout tab, complete the following fields:
   - **Lockout Enabled**: Requests the locking of a user account after invalid attempts to log in to that account exceed the specified Lockout Threshold. By default, this attribute is enabled.
   - **Lockout Threshold**: The number of failed user password entries that can be tried before that user account is locked. Any subsequent attempts to access the account (even if the username/password combination is correct) raise a Security exception. The account remains locked until the System Administrator unlocks it or another login attempt is made after the lockout attempt.
duration period ends. Invalid login attempts must be made within a span defined by the Lockout Reset Duration attribute.

- **Lockout Duration**: The number of minutes that a user’s account remains inaccessible after being locked in response to several invalid login attempts within the amount of time specified by the Lockout Reset Duration attribute.

- **Lockout Reset Duration**: The number of minutes within which invalid login attempts must occur for the user’s account to be locked. An account is locked if the number of invalid login attempts defined in the Lockout Threshold attribute occurs within the amount of time defined by this attribute.

- **Lockout Cache Size**: The number of minutes within which invalid login attempts must occur for the user’s account to be locked. An account is locked if the number of invalid login attempts defined in the Lockout Threshold attribute occurs within the amount of time defined by this attribute.

6. Click Save.

7. Click Release Configuration.

To set password policies:

1. In the **WebLogic Administration Console**, click **Lock & Edit** if not already selected. See the WebLogic Server online Help for more information.

2. In the **Domain Structure** panel of the **Change Center** in the **WebLogic Administration Console**, click **Security Realms**.

   The Summary of Security Realms screen appears.

3. In the **Realms** list, click the name of the security realm to be configured.

   The Settings screen for the selected realm appears.

4. Select the **Providers** tab and then the **Authentication** subtab.

5. In the **Authentication Providers** list, select **DefaultAuthenticator**. (The WebLogic Authentication provider is configured in the default security realm with the name **DefaultAuthenticator**. If you have configured a different authentication provider, select it instead.)

   The Settings for DefaultAuthenticator screen appears.

6. Make any required configuration on the **Configuration** tab, including **Minimum Password Length**.

7. Click Save.

8. Click Release Configuration.

9. Access the Password Policy Utility page by entering the following URL in your web browser:

   https://WebLogic_Host:BEA_SSL_PORT/security/PasswordPolicy.jsp to access.

   The Enter Network Password dialog appears.

10. Enter the WebLogic administrator username and password and click OK.

    The **Password Policy Utility** screen appears.
11. Complete the following fields:

- **PasswordPolicyEnabled**: Enables or disables the following password security features:
  - Password change, including password change upon first login, passwords being changed to a previously-used password
  - Password aging

**Note**: The following security features are always enabled:

- Lockout
- Password length of 6 characters or more, but no longer than 20 characters
- Inclusion of at least one special character and one number

If you disable the password policy, the password history is deleted.

- **Expiration Days**: Specifies the period of time, in days, for which the password is valid
- **Warning Days**: The number of days prior to password expiration that the user is notified of impending password expiration. The user is prompted to change the password.
- **Keep Password History**: The number of previously-used passwords that are tracked in history. If the password policy is enabled, users are unable to reuse passwords tracked in history.

12. Click **Submit**.

**Changing WebLogic Server ASAP User Passwords**

ASAP Administrators can change user passwords at any time using the Change Password Utility JSP, provided that the system uses the WebLogic Server as the user security repository.

An administrator can also change passwords through the WebLogic Administration Console. Oracle does not recommend this because the underlying password policies, such as password length and syntax, cannot be enforced and the administrator is not notified of the policy violation.
Use of the Change Password JSP is not applicable if you are using the WebLogic Server LDAP Realm as the user security repository.

To access the Change Password JSP, you will need the following information:

- Protocol for the WebLogic Server – Use of https is required to provide security.
- The Change Password URL is:
  
  `https://hostname:BEA_SSL_PORT/security/ChangePassword.jsp`

- Root context for Security Service – By default, the ASAP installer sets security as the root context of the Security Service.

  For information on the SECURITY_SERVICE configuration variable in the `ASAP.cfg` file, see the `ASAP Server Configuration Guide`.

JPasswords must contain at least one uppercase character and one number as well as one special character (that is ~ ! # $ % & * ( ) _ + | \ [ ] ` : " ; ' < > ? , . /). This is an internal feature and is not subject to configuration.

To change user passwords:

1. Type the URL for the JSP, for example,

   `https://hostname:BEA_SSL_PORT/security/ChangePassword.jsp`

   The Enter Network Password dialog opens.

2. In the User Name field, type your user name.

3. In the Password field, type your password. Click OK.

   The Change Password Utility window opens.

4. In the User ID field, type the user name of the user whose password you want to change.

5. In the Old Password field, type the user’s current password.
6. In the **New Password** field, type the user’s new password.

7. In the **Confirm Password** field, type the user’s new password again.

8. In the **Admin User Id** field, enter the user ID of the Oracle WebLogic Server administrator.

9. In the **Admin Password** field, enter the password of the Oracle WebLogic Server administrator.

10. In the **Confirm Admin Password** field, enter the password of the Oracle WebLogic Server administrator again.

11. Click **Submit**. A confirmation message appears.

### Disabling the Change Password Feature in the OCA Client

Table 2–6 lists the permission that applies only to the **Change Password** menu item in the OCA client. This feature lets ASAP Administrators disable the change password feature in the OCA client.

<table>
<thead>
<tr>
<th>ACL</th>
<th>Permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceActivation.ASAP.ENV_ID. ASAP_Provisioning_Management</td>
<td>ChangePassword</td>
</tr>
</tbody>
</table>

If the new ACL is not defined in WebLogic Server, the **Change Password** menu item in the OCA client is disabled.

### Managing Locked-out User Accounts

WebLogic user accounts have settings which automatically lock out the user for a period of time after a number of login attempts which fail due to an incorrect password. These settings are modifiable by the administrator.

#### To access WebLogic user settings:

In the **Domain Structure** panel of the Change Center in the WebLogic Administration Console, click **Security Realms**. In the Summary of Security Realms window, click the name of the security realm to be configured.

Select the **Configuration** tab, and the **User Lockout** sub-tab to access user lock-out settings.

See “Setting WebLogic Server ASAP Password Policies” for more details.

#### Settings include:

- **Lockout Threshold** – Number of failed password entries before the account is locked. The account remains locked until it is unlocked by the system administrator, or the lockout duration period ends. The default is 5 failed password entries.

- **Lockout Duration** – Number of minutes that a user’s account remains inaccessible after being locked after **Lockout Threshold** invalid login attempts. The default is 30 minutes.

- **Lockout Reset Duration** – Number of minutes within which invalid login attempts must occur in order for the user’s account to be locked. The default is 5 minutes.

To unlock a locked user account:
1. In the Change Center of the WebLogic Administration Console, click Lock & Edit if not already clicked. See WebLogic Server online Help for more information.

2. In the Domain Structure panel of the Change Center in the WebLogic Administration Console, click the name of the domain.

3. Select the Security tab, then the Unlock User sub-tab.

4. Enter the name of the user to unlock and click Save.

5. In the Change Center of the WebLogic Administration Console, click Release Configuration.

---

**Note:** On a Managed Server, you cannot unlock a locked account through the WebLogic Administration Console. The unlock information is propagated through a multicast message which is only configured in a cluster environment. Instead, use the following command:

```
java weblogic.Admin -url url -username adminuser -password adminuserpassword -type weblogic.management.security.authentication.UserLockoutManager -method clearLockout lockedusername
```

Alternatively, wait until the Lockout Duration time has passed.

---

**Updating Methods Role Assigned to a Group or User in WebLogic**

This section provides information on how to update methods in an ASAP role assigned to a group or a user in WebLogic Server. You can use the information defined in deployment descriptors to grant security roles and define security policies.

You can update the deployment descriptors by using WebLogic Workshop (Eclipse component) or by editing the `ejb-jar.xml` and `weblogic-ejb-jar.xml` files manually.

To edit the XML files manually:

1. Navigate to `WebLogic_domain/servers/WebLogic_server/upload/asapENV_ID/app` (where the `WebLogic_domain` is the installation directory for your WebLogic Server domain, `WebLogic_server` is the name of your WebLogic Server domain, and `ENV_ID` is the ASAP environment ID).

2. Do the following:

   ```
   jar xvf asapENV_ID.ear ssam.jar
   jar xvf ssam.jar META-INF/ejb-jar.xml
   jar xvf ssam.jar META-INF/weblogic-ejb-jar.xml
   ```

3. Edit `ejb-jar.xml` and `weblogic-ejb-jar.xml` to add, remove, or modify roles to users or groups.

For example, you may remove the following three methods from the Monitor Role in `ejb-jar.xml` file:

```xml
  <method>
    <ejb-name>ASAPSecurityServices</ejb-name>
    <method-name>SA_Aborted_Work_Order_Report</method-name>
  </method>
  <method>
    <ejb-name>ASAPSecurityServices</ejb-name>
    <method-name>SA_Activity_Report</method-name>
  </method>
```
<method>
<ejb-name>ASAPSecurityServices</ejb-name>
<method-name>SA_ASDL_Report</method-name>
</method>

4. Do the following:
   jar uvf ssam.jar META-INF/ejb-jar.xml
   jar uvf ssam.jar META-INF/weblogic-ejb-jar.xml
   jar uvf asap$ENV_ID.ear ssam.jar

5. Redeploy the asap$ENV_ID.ear file.

Configuring ASAP Server and Database Credential Security

Secure data must be stored in a secure location and distributed to authorized users. The ASAP security system governs how secure data is managed and the ASAP diagnostics files are secured:

- **Secure Data Storage**
  
The ASAP security administrator pre-defines the nature and accessibility of secure data for each ASAP server. Class A secure data is stored in the CSF wallet during the initial ASAP installation procedures (see the ASAP Installation Guide for more information). The ASAP installer distributes this wallet to each server during the ASAP installation.

- **Secure Data Encryption**
  
The CSF wallet encrypts all data contained in it and obtained from it. In addition, the CSF wallet file (cwallet.sso) has restricted access permissions. Many ASAP utilities and scripts use the passwords contained in the CSF wallet. For more information about how these utilities and scripts use the CSF wallet security feature, see "Using the Credential Store Factory Wallet with ASAP Utilities and Scripts."

Your ASAP security administrator creates the class A secure data while installing ASAP. Your administrator can also modify ASAP Server passwords using the ModifyPasswords script (see "Changing Database Passwords in the Credential Store Factory Wallet").

About Credential Store Factory Wallet Secure Data Management

Secure data management in ASAP involves:

- Setting up and maintaining secure data storage
- Encrypting secure data during provisioning

**Setting up and Maintaining Secure Data Storage**

During the ASAP installation procedure, your ASAP security administrator must enter a predefined user name and password for each ASAP server. The administrator must also enter the WebLogic Server, and Oracle database user names and passwords. ASAP stores these user names and passwords in the CSF wallet (class A secure data).

**Data Encryption**

The CSF wallet provides transparent encryption features that protect all credentials it stores and transmits.
Using the Credential Store Factory Wallet with ASAP Utilities and Scripts

When the ASAP security feature is configured (security level of the Master Control server is not 0), every ASAP utility (scripts or programs) that require access to the ASAP server or database are prompted to enter a password based on the target.

The following utilities have optional arguments -P and -d for ASAP security (see Table 2–7 for more information about these arguments).

- asap_recompile
- asap_utils
- start_asap_sys
- start_control_sys
- startc
- starts
- stop_asap_sys
- stopc
- stops

Table 2–7  ASAP Security Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-P</td>
<td>Used to specify the control database password and used in production.</td>
</tr>
<tr>
<td>-d</td>
<td>The utilities retrieve password information from the CSF wallet. The ASAP installer creates the CSF wallet during the installation of ASAP. You can modify the passwords using the ModifyPassword script (see Changing Database Passwords in the Credential Store Factory Wallet).</td>
</tr>
</tbody>
</table>

You can use either the -P and -d option. However, when both options are used, the -d option overrides -P.

For example:

```
start_asap_sys -P control_database_password
start_asap_sys => User needs to enter control_database_password
start_asap_sys -d => Retrieve password the CSF wallet
```

**WARNING:** The functionality to include your password in the command line when using ASAP utilities and scripts is a security risk. This functionality has been deprecated and will be removed in a future release. You can avoid this risk by omitting the password, and entering it only when you are prompted to enter it.

Changing Database Passwords in the Credential Store Factory Wallet

The cwallet.sso file is found in the ASAP_Home/install directory. This file contains security information for installation purposes and includes the ASAP database schema user names and passwords, the Oracle DBA user name and password, and the WebLogic Server domain user name and password. The ASAP installer creates this file during the installation process.

To change the CSF Wallet passwords, use the following procedure:
1. Source the Environment_Profile located in the ASAP_Home/ directory.

2. Set the TNS_ADMIN UNIX variable to the location of your tnsname.ora file.
   
   ```bash
   export TNS_ADMIN=/home/examplelocation/
   ```

3. From ASAP_Home/scripts directory run the ModifyPassword tool.

4. Enter the DBA user name and password.

5. Enter and modify the ASAP database user names and passwords as required.

Configuring Security for Network Elements Communication

NE credentials (also called custom secure class B data) used primarily by NEPs to establish network connections to NEs must be stored in a secure location and distributed to authorized users.

The ASAP security tool supports the following features to protect NE credentials:

- **Secure Data Storage**
  
  An administrator can use the ASAP security tool to create NE credentials and store these credentials in a central repository on the Master Control server. The Master Control server distributes these credentials to SRPs, JSRPs, NEP, or Java-enabled NEPs (JNEPs).

  ASAP stores NE credentials in the Master Control server in the tbl_classB_secu database table.

- **Secure Data Encryption**
  
  The Master Control server uses a symmetric secret key encryption method to achieve data confidentiality for custom secure data.

- **Key Distribution**
  
  The Master Control server acts as a key distribution server, and distributes custom secure data to every ASAP server during provisioning. To acquire the custom secure data, ASAP server has a pre-defined key distribution protocol.

Understanding the Custom Secure Data Structure

Your ASAP security administrator can define class B custom secure data through application programming interfaces (APIs) or action functions used by the customized SRP, JSRPs, NEP, and JNEPs. Your ASAP security administrator can also set up your custom secure data using the ASAP security tool.

The class B custom secure data typically includes the user names and password for the NEs and should only be used by custom NEPs, JNEPs, SRPs, or JSRPs.

The Master Control server contains the secure data that has name, value, creation date, and description fields. The Master Control server distributes the secure data to each ASAP server during provisioning.

Table 2–8 provides a detailed description of a secure data entry in the secure data storage. The Security Level, Alg (sdru), and Audit Level fields apply only to ASAP secure data.
For custom secure data storage, the required fields are Name, Value, Creation Date, and Desc.

### Managing Custom Secure Data

Secure data management in ASAP involves:

- Setting up and maintaining secure data storage
- Encrypting secure data during provisioning
- Key distribution (for custom secure data)
- Local caching of custom secure data, for improved system performance

---

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Length</th>
<th>Encryption</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Char</td>
<td>80</td>
<td>No</td>
<td>The name field of a secure data entry. This is used as a key to retrieve the secure data entry.</td>
</tr>
<tr>
<td>Value</td>
<td>Char</td>
<td>80</td>
<td>Yes</td>
<td>The encrypted value of the secure data entry.</td>
</tr>
</tbody>
</table>
| Security Level | Integer |        | No         | This field is only applicable to ASAP secure data (the class field value is 1). If the name is an ASAP server name (for example NEPab12), then this controls the level of ASAP security; otherwise this is ignored. Possible values:
  - 0 – Turn off security feature (default).
  - 1 – Turn on security feature.
  The security level of the Master Control server controls the level of the entire ASAP security. The security level of each ASAP server is only applicable when the security level of the Master Control server is not 0. |
| Cache      | Integer  | 1      | No         | This field is only applicable to ASAP secure data (the class field value is 1). If the name is an ASAP server name, then this controls caching ASAP secure data; otherwise this is ignored. Possible values:
  - 0 – Turn Off cache feature (default).
  - 1 – Turn On cache feature. |
| Audit Level| Integer  |        | No         | Reserved for future use.                                                                                                                    |
| Creation Date | Date    |        | No         | The creation date.                                                                                                                         |
| Alg        | Integer  |        | No         | The deployed secret key algorithm. The default value is 1 (BLOWFISH_ALGORITHM). The applies to the ASAP secure data only.                   |
| Class      | Integer  |        | No         | The secure data classes are:
  - ASAP secure data
  - Custom secure data
  The custom SRP, JSRP, NEP, and JNEP only manipulates custom secure data.                                                                 |
| Desc       | Char     | 255    | No         | The description of the secure data entry.                                                                                                  |
Setting up and Maintaining Secure Data Storage
Your administrator can set up the initial custom secure data storage repository and can predefine this custom secure data. ASAP stores this data in a central repository: the Master Control server.

Encrypting Data During Network Element Provisioning
To protect custom secure data during provisioning, symmetric or secret key encryption is used. You can use the ASAP APIs and action functions to:
- Retrieve ASAP secure data and custom secure data
- Update or add custom secure data

To control the security-sensitive messages that come from state table action functions, use the `ACT_FUNC_SEC` state table variable.

The following example shows how to use `ACT_FUNC_SEC`.

```
BEGIN TEST_PROG
  110 CONCAT 'ACT_FUNC_SEC = 1'
  110 DMS_LEN '%FORMAT_LEN = %MCLI:%NXX:%LEN'
  120 CONCAT 'ACT_FUNC_SEC = 0'
  130 CONCAT '%MY_LEN = %LEN'
END TEST_PROG
```

The NEP diagnostic file does not show the CSDL parameter values of NXX, LEN, and MCLI from 110 because `ACT_FUNC_SEC` is set to 1. The value of MY_LEN (for example, 6742727 from CSDL parameter LEN) appears in the NEP diagnostic since `ACT_FUNC_SEC` is set to 0 on line 120.

Any State Table action function can be placed between two CONCAT statements, but only the following action functions support the security feature:
- CONCAT
- DMS_FEATS
- DMS_LEN
- EXEC
- EXEC_RPC
- GET
- GET_INCPT
- GET_LTG
- SEND
- SENDECHO
- SUBSTR

Understanding Key Distribution for Custom Secure Data
The Master Control server provides key distribution to every ASAP server to ensure the secure transmission of custom secure data. To acquire the secure data, the ASAP server must follow the pre-defined key distribution protocol. The Master Control server does the following:
- Ensures that the implementation details of the secure data storage, such as the physical media, file location, and content, are transparent to ASAP server. By
following pre-defined key distribution protocol set up by your ASAP security administrator, each ASAP server can acquire the necessary secure data.

- Ensures that the secure data is accessible only to authorized users. Before requesting secure data from the key distribution server, each ASAP server authenticates itself using the session ID given by the Master Control server.

- Provides a central location for secure data storage.

The Master Control server uses a one-time password named as a session ID to authenticate the requester. The session ID is the unique integer value per instance of the Master Control server.

**Caching Secure Data on Local Servers**

By default, each ASAP server retrieves the secure data from the Master Control server. In addition, to provide better performance in retrieving secure data, you can configure a cache that contains recently-used secure data in the local ASAP server. In this case, the ASAP server checks the local cache first, then follows the key distribution protocol. A cache provides better performance in retrieving secure data, but it can reveal secure data to unauthorized users. For example, the attacker may kill the process to generate core file and retrieve secure data from the core file.

**Securing Network Element Credentials with the Security Administration Tool**

You can use the command line ASAP security administration tool to set up and maintain the secure data.

```
asap_security_tool [-u CtrlUserID][-p CtrlPasswd] [-l diag_level] [-c ctrl_svr_name] [-f diag_file_name][-h]```

Table 2–9 lists and describes the ASAP security tool arguments.

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-u</td>
<td>The login ID of the ASAP security administrator (Control Database User).</td>
</tr>
<tr>
<td>-p</td>
<td>The password of the ASAP security administrator (Control Database Password).</td>
</tr>
<tr>
<td>operation_option</td>
<td>One of the operations in the Detailed Operation section that follows.</td>
</tr>
<tr>
<td>-c</td>
<td>The name of the Control server for the application (can be omitted).</td>
</tr>
<tr>
<td>-l</td>
<td>The diagnostic level for the application. The default value is <strong>LOW_LEVEL</strong>. For more information on diagnostic levels, see “Server Diagnostic Levels”.</td>
</tr>
<tr>
<td>-f</td>
<td>The name of the diagnostic file for the application. The default is <strong>ASC_SECU.diag</strong>.</td>
</tr>
<tr>
<td>-h</td>
<td>Displays a help page.</td>
</tr>
</tbody>
</table>

All arguments are optional. When the **operation_option** is omitted from the command line, a menu is provided, displaying all the available operations.

The ASAP Security Utility Script provides the following options:

***** ASAP Security Utility Script *****

1. Initialize the secure data storage
10. Add/Modify secure data entry
11. Delete secure data entry
12. Query secure data entry

20. Import secure data from file to the secure storage
30. Change current security level
40. Flush secure data cache for each ASAP server

100. Display usage message for this Tool

Enter Choice <Q - Quit>:

Table 2–10 lists and describes the ASAP security tool operations.

<table>
<thead>
<tr>
<th>Operations</th>
<th>Description</th>
</tr>
</thead>
</table>
| -i         | Initializes secure data storage. With this option, the ASAP security administrator can set up class B custom secure data.  
            Note: This operation is only available when ASAP is not running.  
            Note: Selecting option 1 to initialize the secure data storage, deletes all the security data in the class A CSF wallet and class B table. |
| -x         | Checks and changes the current security level.  
            Note: this operation is only available when ASAP is not running. |
| -e         | Flushes the secure data cache for each ASAP server. |
| -a data    | Adds or modifies a secure data entry.  
            ■ ASAP data format:  
            name:value:class:secu_level:cache_mode:  
            audit_level: alg:description  
            ■ Custom data format:  
            name:value:description  
            You can add customer secure data while the Control server is running. When the addition of data is complete, the `asap_security_tool` requests the Control server to flush cached secure data. |
| -d data    | Deletes a secure data entry.  
            ■ Data format:  
            name:class  
            You can delete customer secure data while the Control server is running. When the deletion of data is complete, the `asap_security_tool` requests the Control server to flush cached secure data. |
The cache functionality is controlled by the value of the cache_mode defined for the control database login and password. The caching functionality can be changed through menu option 10 in the \textit{asap_security_tool} by modifying the control database login. If the value of the cache mode for the control database login is 0, then the Control server caches the secure data. Otherwise, the Control server retrieves the secure data from the database as required.

### Additional Security Considerations

In addition to the other security features, observe the following guidelines described in this section.

#### Setting Secure Diagnostic Levels

The ASAP diagnostic files contain information logs on provisioning activity and confidential provisioning information such as WO parameters and NE dialog as plain text. The secure ASAP diagnostic feature addresses the following key provisioning data:

- NE dialog to control the content of diagnostic file
- WO parameter

#### Setting the Network Element Dialog Diagnostic Configuration Parameter

The ASAP NEP diagnostic file contains switch-sensitive information sent and received from the NE. In production environments, Oracle recommends that the audit level is set to SANE so that switch-sensitive information is not included in diagnostics files. Oracle also recommends that you delete old archives and/or store archives in a secure manner.

In addition to setting the audit level to SANE, you can also enable or disable the configurable variable, \texttt{NE\_DIALOG\_OFF}, in \texttt{ASAP.cfg}. This variable controls the source code to print out all NE dialog messages from NEP diagnostic file. All the output NE dialog in the NEP application checks against the value of \texttt{NE\_DIALOG\_OFF}, and cuts off the message if the \texttt{NE\_DIALOG\_OFF} is set to 1.

Table 2–11 lists and describes the \texttt{NE\_DIALOG\_OFF} option.
Although the NE dialog from the state table action function can be controlled by the action statement, some NE dialogs are not related to any action function, and in this case the \texttt{NE\_DIALOG\_OFF} is used to hide NE information.

**Setting Work Order Information Diagnostic Levels**

Work orders typically contain business sensitive information. The WO is processed by several components, like Service Request Processor (SRP), Service Activation Request Manager (SARM), OCA; consequently, the WO information is exposed to different diagnostic files. For this reason, Oracle recommends that the diagnostic level be set to SANE in productions environments to avoid unnecessarily exposing sensitive information. As well, any archival diagnostics files should be stored in a secure manner. For more information on diagnostic levels, see “Server Diagnostic Levels”.

In addition to setting the audit level to SANE, you can also enable or disable the configurable variable, \texttt{WO\_SECURITY\_PROP}, in the ASAP WO. This variable controls the source code to print out the information messages from all ASAP diagnostic files for a particular WO.

*Table 2–12* lists and describes the \texttt{WO\_SECURITY\_PROP} option.

**Table 2–11  \texttt{NE\_DIALOG\_OFF} Configuration Variable**

<table>
<thead>
<tr>
<th>ASAP Configuration Variable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{NE_DIALOG_OFF}</td>
<td>0</td>
<td>Controls the NE dialog message in the diagnostic file. Possible values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 0 – NE dialog messages appear in the diagnostic file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 1 – Secure the NE dialog. No NE dialog messages appear in the diagnostic file.</td>
</tr>
</tbody>
</table>

The WO information is output through a generic diagnostic function call, which can also output other information in addition to the WO information. A filter list controls the output. When \texttt{WO\_SECURITY\_PROP} = 1 and the message type is contained in the filter list, there is no output to diagnostic file.

All possible output message function calls that are scattered in several ASAP applications must be examined.

*Note:* This security measure also applies to WO information that is fetched from the SARM database.
Securing Java or State Table NEP or JNEP to NE Connection Implementations

Since the NEP and the JNEP are designed to communicate with a variety of NEs, there are various methods used for NEP or JNEP to NE security. You can implement the Login State Table or equivalent Java method to retrieve a user_id and password from the NEP or JNEP database. The connection to the NE can then be opened with the correct identification.

Advanced State Table or Java programmers can also implement a password timeout and automatic password change functionality as requested by the switch vendor and company policies.

Setting SRP or JSRP to SARM Security Properties

Each WO within ASAP can be authorized prior to its acceptance in the SARM. The user_id and password properties are compared against the SARM database table tbl_uid_pwd.

- **User_id** (optional) – The user_id that the SARM uses for security validation. The user_id is required if the SARM’s security validation feature is enabled in the SECURITY_CHECK configuration parameter. By default, the security validation feature is turned off.

- **User Password** (optional) – Indicates the password which the SARM uses for security validation. The User Password is required if the SARM’s security validation feature is enabled in the SECURITY_CHECK configuration parameter.

Setting Security Between Servers

To increase security between Open Servers, the configuration variable CLIENT_SECURITY_ON determines whether to authorize every connection made to the Open Server. When a client (or another server) establishes a connection to a server, the user_id and password used by that connection is employed to open a connection to the SQL server. If the connection to the SQL server is successful, then the connection is accepted; otherwise it is rejected.

Enabling Schema Validation for the JSRP JMS Interface

You can enable the JSRP to validate incoming Java Message Service (JMS) XML messages at the JSRP JMS interface against the ASAP schemas that enforces xml WO message formation. Enabling schema validation helps to secure the JSRP JMS interface against invalid XML messages, but also incurs a performance impact.

To enable schema validation, set the VALIDATION_ENABLED parameter to True in the ejb-jar.xml file from srp.jar file of Domain_home/servers/server_name/upload/asapenvid.ear deployment (where server_name is the name of your WebLogic server instance, and envid is the environment ID for your ASAP instance.)
This chapter describes monitoring and managing Oracle Communications ASAP.

Overview of Monitoring and Managing ASAP

ASAP provides the following management and monitoring capabilities:

- **ASAP process monitoring** – The ASAP Control server starts, stops, and monitors all processes in ASAP. Upon startup, each ASAP application (client or server) establishes a connection to the Control server. The Control server immediately installs a disconnection handler on that connection. When the connection is lost, a callback function is triggered in the Control server, indicating an abnormal termination of the process. A controlled shutdown of ASAP applications does not trigger this callback. When the process termination is detected, the Control server immediately attempts to restart the process (if configured).

- **ASAP database monitoring** – The ASAP Control server periodically monitors the use of database tables and database devices. The frequency of this database monitoring task is defined using the `DB_MONITOR_TIME` attribute in the `ASAP.cfg` configuration file (see "About Monitoring Database Segment and File System Size" for more information on this attribute). The Control server reads a configuration table that lists the database use threshold values and issues a system event if a particular threshold is exceeded. A custom routine can then be called to perform clean-up tasks and notify the system administrators. For more information on configuring these thresholds and system events, see "Configuring System Events".

- **ASAP file system monitoring** – ASAP periodically monitors the use of UNIX file systems. The frequency of this monitoring task is defined using the `FS_MONITOR_TIME` attribute in the `ASAP.cfg` configuration file (see "About Monitoring Database Segment and File System Size" for more information on this attribute). The Control server compares the file system usage against the configuration table, and issues a system event if any threshold value exceeds. A custom routine can then be called to perform clean-up tasks and notify the system administrators. For more information on configuring system events, see "Configuring System Events". For more information on configuring this thresholds, see the ASAP Developer’s Guide.

- **ASAP process information** – Each ASAP application server logs process performance information (for example, CPU utilization, memory usage) and periodically stores it in the database. This data can then be used to determine periods of low activity suitable for maintenance. For more information, see the chapter about the ASAP utility in the ASAP Server Configuration Guide.
- **ASAP database admin process** – Each ASAP server can perform administration tasks within its application database. At a user-configured time, the server administration thread connects to its primary database to perform database archiving, purging, and integrity checks. This thread is also responsible for executing database administration commands to optimize all query plans for the database procedures. This activity ensures that ASAP database performance is not affected by large changes in data volumes. See "Managing the Database and File System" for more information.

- **ASAP diagnostic and system events** – ASAP is equipped with diagnostic call and system activity events that monitor its internal health. ASAP events can be mapped to various alarm categories to generate alarms viewable through a user-defined management console. For more information, see "Configuring System Events and Alarms Using Stored Procedures".

### Configuring System Events and Alarms Using Stored Procedures

System events range from informational events to critical error events. When an ASAP client or server generates a system event, the event is transmitted to the Control server which then saves the event. The Control server determines if the event requires an alarm to be triggered. The Control server determines the alarm centers that are mapped to the alarm and the alarm program to call. This alarm program can be a shell script, a UNIX executable, or a Linux executable to send the alarm to email or print.

*Figure 3–1  Alarm Process*

![Alarm Process Diagram](image)

The Control server also monitors each application process within ASAP. If an application process terminates, the Control server issues an Abnormal Process Termination event that is mapped to a user-defined alarm program. Abnormal Process Termination events are written to both the application diagnostic file and the database.

Each event can be configured to enable or disable a system alarm. Every alarm has an associated alarm level (such as MAJOR, MINOR, or CRITICAL) and Auto Clear flag. The Auto Clear flag specifies that the alarm should be automatically cleared upon generation. If the alarm is configured as non-auto clearing, it is generated every five minutes until you reset it using the `asap_utils` administrative remote procedure call.
Configuring System Events and Alarms Using Stored Procedures

(RPC) interface (function 60). If configured as auto-clearing, the alarm is generated only once.

For more information on asap_utils, see ASAP Server Configuration Guide.

An alarm can map to up to five alarm centers which trigger alarm programs to alert users of the system alarm. Typically, such alarms can email an administrator, print reports to an alarm center, page a particular individual, or communicate with a network management system.

Configuring Alarm Centers for Alarm Notification and Escalation

Alarm centers can be pagers, email accounts, printers, network management systems, and so forth. Alarm centers are notified of alarms by means of a UNIX program, or shell script.

To create an alarm center, specify the UNIX program, or shell script that ASAP executes when the alarm is generated. For example, assume there are two alarm centers: a general ADMIN center, which receives messages on a printer, and another center called ADMINPGR, which receives pager notifications. To notify these alarm centers, you must develop two shell scripts or programs.

Alarm center information is contained in tbl_alarm_center. Alarm centers are later mapped to system alarms in tbl_system_alarm. If you are defining a system event that does not map to an alarm, proceed to “Configuring System Events”.

Use the following stored procedures to define, list, or delete alarm centers:
- CSP_new_center – Defines an alarm center to which alarm notifications must be sent by the Control server.
- CSP_list_center – Lists an alarm center definition from the control database.
- CSP_del_center – Deletes an alarm center definition from the control database.

For more information on these stored procedures and tbl_alarm_center, refer to the ASAP Developer’s Guide.

Configuring System Alarms

System alarms identify system events, such as component malfunction. System events can be mapped to operations which enable, disable, or log alarms.

Alarms in ASAP can be routed to multiple alarm centers, based on the time of day.

The types of alarms that can be configured in ASAP are:
- Continuous alarm – The continuous alarm is issued periodically until your system administrator manually disables it or ASAP issues a system event to disable the alarm.
- Single initiation alarm – The single initiation alarm is generated once each time the system event is issued.

Table 3–1 lists and describes alarms and what action is required in response to the alarm.

<table>
<thead>
<tr>
<th>Alarm</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abnormal Process Termination</td>
<td>This alarm indicates that a component of ASAP has shut down. You can immediately save the log file and take the appropriate steps to restart the process.</td>
</tr>
</tbody>
</table>
### Table 3–1 (Cont.) System Alarms

<table>
<thead>
<tr>
<th>Alarm</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>General System Error</td>
<td>This is a general purpose system alarm. The description of the error from the ASC_event() call should give sufficient details of the error. If ASAP is otherwise running properly, you can simply note this alarm and deal with it at a convenient time.</td>
</tr>
<tr>
<td>General System Warning</td>
<td>A general warning event, not an error situation. This event can be used to indicate errors or exceptions in external systems, for example, Port Bind Failure, Network Element (NE) in Maintenance Mode, and Stored Procedure Deadlock Victim. You can note this alarm and deal with it at your convenience.</td>
</tr>
<tr>
<td>General System Information</td>
<td>System information such as thread spawning, process startup, and graceful shutdown, work order (WO) time-outs, Host CLLI in Busy State, etc. These alarms typically do not require user intervention.</td>
</tr>
<tr>
<td>Process Termination Error</td>
<td>This error is generated by an application process whenever an internal condition is encountered. This causes the application process to log the system event in the database and then terminate the application process in an orderly manner. This is a critical error. You must immediately determine the problem and restart the process.</td>
</tr>
<tr>
<td>Operating System Disk File Error</td>
<td>This is a critical error. This error is generated when an error is encountered when creating, reading, or writing an operating system file. For example, the Network Element Processor (NEP) is unable to create a file to insert the NE response log, and the Service Request Processor (SRP) is unable to create a file for WO completions. You must deal with this alarm immediately.</td>
</tr>
<tr>
<td>Critical Database Resource Error</td>
<td>This can be a serious error. The event is issued by inv_rpc() and other RPC calls if a database or transaction log is full, the database is corrupt, etc. You must attend to this alarm immediately, although you can continue to use the system.</td>
</tr>
<tr>
<td>Invocation Application Remote/Registered Procedure Failed</td>
<td>This alarm is issued if the RPC or registered procedure fails. The text in the ASC_event() gives the invocation details. If ASAP is otherwise running properly, you can note this alarm and address it at your convenience.</td>
</tr>
<tr>
<td>UNIX System Call Error</td>
<td>This is a critical error that is issued if an application encounters a UNIX system call error. You must deal with this alarm immediately.</td>
</tr>
<tr>
<td>Application Network Error</td>
<td>This is a serious error. (Unable to connect to the SQL/ Application server, connection to server goes down, etc.). You should save the log file immediately and deal with this situation.</td>
</tr>
<tr>
<td>Open Server Application Object Access Error</td>
<td>This error is issued if ASAP is unable to create, remove, lock, unlock, or access Open Server object (Mutex, message queue, etc.). If ASAP is otherwise running properly, you can note this alarm and deal with it at your convenience.</td>
</tr>
<tr>
<td>Work Order In Progress Longer Than Specified Threshold</td>
<td>System information event. Indicates that a WO has been in progress longer than a specified threshold time. The description of the error from the ASC_event() call provides information on the error. If ASAP is otherwise running properly, you can note this alarm and deal with it at your convenience.</td>
</tr>
<tr>
<td>Work Order Routing Error</td>
<td>This error is generated when the Service Action Request Manager (SARM) is unable to correctly determine the Host NE to which a WO should be routed. The description of the error from the ASC_event() call provides the routing data (Directory Number (DN) or MCLI ). Updates to the routing tables are required.</td>
</tr>
<tr>
<td>Network Element Has Gone Into Maintenance Mode</td>
<td>An informational alarm. This alarm is generated when an NE enters maintenance mode. If the alarm persists, investigate the reason.</td>
</tr>
</tbody>
</table>
Each system event can be optionally mapped to a system alarm in 
\texttt{tbl\_system\_alarm}.

You can specify the system alarm level: MINOR, MAJOR, or CRITICAL. Each alarm is either self auto-clearing or non-auto-clearing, and may or may not define one or more alarm centers to which the alarm is to be transmitted.

If the alarm is configured as non auto-clearing, it is generated every five minutes until you reset the alarm. If the alarm is configured as an auto-clearing alarm, it is only generated once.

After you have defined the alarm centers for the system, you can define the alarms. An alarm code is used to identify uniquely each alarm in ASAP. You can configure an alarm level for each alarm and up to five different alarm routes. The alarm routes are used to specify daily time intervals for an alarm to be generated at up to five alarm centers and to specify the time period in minutes between a regeneration of the alarm.

Use the following stored procedures to define, list, or delete system alarms:

- \texttt{CSP\_new\_alarm} – Defines an alarm within ASAP, and optionally, the associated “alarm code” that may be associated with the event.
- \texttt{CSP\_list\_alarm} – Lists system alarm codes and their definitions.
- \texttt{CSP\_del\_alarm} – Deletes a system alarm code.

For more information on these stored procedures and 
\texttt{tbl\_system\_alarm}, refer to the 
\textit{ASAP Developer’s Guide}.

\section*{Configuring System Events}

In ASAP, each application can generate system events from within the application code. In addition, you can configure system events to generate an alarm in certain circumstances [for example, Common Service Description Layer (CSDL) failure, SARM to SRP notifications, database, and file system thresholds being exceeded, and so on] using certain static tables within ASAP.

Each system event is configured in the control database and can be mapped to a particular system alarm. Each alarm can, in turn, be mapped to one or more alarm centers. Each alarm center can then execute a user-supplied alarm program to perform the relevant user-determined alarm notification.

\textbf{Note:} You can define system events in custom code. The system events you define do not have to be mapped to core system events. To generate alarms, ensure that the event you have defined is mapped to an alarm in the Event/Alarm Configuration function.

\section*{Defining System Events}

Core and customer-specific subsystems can generate system events. System events are defined in the control database table 
\texttt{tbl\_event\_type}.

Events can be associated with an alarm code. In addition, the event can enable or disable the alarm.

Use the following stored procedures to define, list, or delete system events:

- \texttt{CSP\_new\_event} – Defines an event type within ASAP, and optionally, the associated “alarm code” that may be associated with the event.
- \texttt{CSP\_list\_event} – Lists database threshold definitions.
- \texttt{CSP\_del\_event} – Deletes a database threshold definition.
For more information on these stored procedures and \texttt{tbl\_event\_type}, refer to the \textit{ASAP Developer’s Guide}.

**Setting Database Thresholds**

The Control server monitors database and transaction log sizes during normal operation. You must configure the database and transaction log thresholds for each database and ASAP environment and the system events that trigger an alarm. When the threshold is exceeded, ASAP can issue the system event to trigger an alarm.

Use the following stored procedures to define, list, or delete database thresholds:

- \texttt{CSP\_new\_db\_thresh} – Defines database and/or transaction log thresholds to be used by the Control server.
- \texttt{CSP\_list\_db\_thresh} – Lists database threshold definitions.
- \texttt{CSP\_del\_db\_thresh} – Deletes a database threshold definition.

Database thresholds are defined in \texttt{tbl\_db\_threshold}.

For more information on these stored procedures and \texttt{tbl\_db\_threshold}, refer to the \textit{ASAP Developer’s Guide}.

**Sample Alarm Program - admin.sh**

The \texttt{ASAP\_Home/scripts/admin.sh} sample script is a UNIX shell script. In this example, the \texttt{admin.sh} and \texttt{adminp.sh} scripts are linked copies of each other. Alarm scripts can also be a UNIX executable program. You must copy this script from the \texttt{ASAP\_Home/scripts} folder to the \texttt{ASAP\_Home/programs} folder in order for a control server alarm center to use it.

System events can be issued for critical, major, and minor system errors, system warnings, and system information.

```
# SCCS Id: @(#) admin.sh 1.1@(#)
# Script to perform alarm notification.
#
# Note that this script should be placed in the $PROGRAMS directory in order # for the Control Server to pick it up.
# TMPFILE=/tmp/alarm.$$
TMPFILE=/tmp/alarm.$$ PERMFILE=$LOGDIR/ASAP_Alarm_log USR=oper1 DEVICE=console
do
case $c in
  i) event_id=$OPTARG;;
n) alarm_name=$OPTARG;;
e) event_code=$OPTARG;;
E) event_desc=$OPTARG;;
d) event_text=$OPTARG;;
f) source_file=$OPTARG;;
l) source_line=$OPTARG;;
a) level=$OPTARG;;
s) application=$OPTARG;;
\?) ;; esac
```

3-6  ASAP System Administrator's Guide
done
shift `expr $OPTIND - 1`
echo "\n*************************************************************************
ASAP Alarm Issued @ `date`  
Alarm Program = $0\n*************************************************************************
\nArguments:\nEvent Id = $event_id\nAlarm Name = $alarm_name\nEvent Code = $event_code\nEvent Desc = $event_desc\nEvent Text = $event_text\nSource File = $source_file\nSource Line = $source_line\nAlarm Level = $level\nApplication = $application\n\nAdditional Parameters = $*\n** End of Alarm **"> $TMPFILE

If the script name is **admin.sh**, the alarm message is appended to the alarm log file. This alarm program is only called for minor errors, and therefore does not have to be sent to the users immediately.

If the script name is **adminp.sh**, the alarm message is appended to the alarm log file, written to **oper1** on the system console and mailed to **oper1** as well. This alarm program sends a beep to the console and is only called for major alarms such as a process crashing.

**Sample Alarm Output**
This section contains sample alarm output.

*************************************************************************
ASAP Alarm Issued @ Wed Aug 7 21:00:29 ADT 2002
Alarm Program = /ASAP/PRODUCTION/programs/adminp.sh
*************************************************************************
Arguments:
Event Id  = 71457
Alarm Name = ABNORMAL
Event Code = ABNORMAL
Event Desc = Abnormal Process Termination - Application
Event Text = Warning: Abnormal Termination of Process LU62SEND
Source File = process.c
Source Line = 701
Alarm Level = CRITICAL
Application = CONTROL

Additional Parameters =
The output specifies the time and date of the alarm, the script called by the alarm, together with the following information:

**Table 3-2 Alarm Output**

<table>
<thead>
<tr>
<th>Alarm Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event ID</td>
<td>Unique ID for the event that generated the alarm.</td>
</tr>
<tr>
<td>Alarm Name</td>
<td>Alarm code associated with the system event.</td>
</tr>
<tr>
<td>Event Code</td>
<td>ASAP event generated by the application.</td>
</tr>
<tr>
<td>Event Desc</td>
<td>Brief description of the event.</td>
</tr>
<tr>
<td>Event Text</td>
<td>Brief description of the reason for the system event within the source code.</td>
</tr>
<tr>
<td>Source File</td>
<td>Line in the source file where the event was generated.</td>
</tr>
<tr>
<td>Source Line</td>
<td>Source file name where the event was generated.</td>
</tr>
<tr>
<td>Alarm Level</td>
<td>Possible values: MINOR, MAJOR, or CRITICAL.</td>
</tr>
<tr>
<td>Application</td>
<td>Logical name of the ASAP application server that generated the system event</td>
</tr>
</tbody>
</table>

**Sample Alarm Program - POTS Cartridge Sample Code**

ASAP provides a working sample cartridge that you can install if you select the POTS Service Activation Model installation option. This cartridge contains the `ASAP_Home/samples/sadl/DemoInstall/Nortel/DMS/POTS/control/PLSQL/demo_control.sql` script that populates alarm definitions and creates an alarm center mapped to the `ASAP_Home/programs/control_prog` script. The ASAP installer automatically runs the `demo_control.sql` script if you select the POTS Service Activation Model installation option.

The alarm center created by the `demo_control.sql` script runs the `control_prog` script when it received an alarm or event notification. The `control_prog` script creates an alarm log file located in `ASAP_Home/DATA/logs/ControlProgramOutput` that records the alarm output.

**Understanding Default System Events**

The static table `tbl_event_type` contains the system events that the ASAP application can generate and, if required, the system alarm code associated with that event. You can modify existing system events in this table or add custom events.

For information on adding alarms and events, see "Configuring System Alarms" and "Configuring System Events".

Each system event must have a record in `tbl_event_type`.

The following tables contain the system events that are contained in `tbl_sys_event` and are generated by the application. You can update the static text, alarm level, and description of these system events and add custom events.

**API System Events**

Table 3-3 lists and defines the system events included in the core application programming interface (API).
### Table 3–3  Core API System Events

<table>
<thead>
<tr>
<th>Event</th>
<th>Static Text</th>
<th>Alarm Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABNORMAL</td>
<td>Abnormal process termination as the application process terminated unexpectedly.</td>
<td>Critical. Non auto-clearing.</td>
<td>Event issued by the Control server if any process (server or client) it monitors has terminated unexpectedly.</td>
</tr>
<tr>
<td>DISK_ERR</td>
<td>Operating system disk file error.</td>
<td>Critical. Auto-clearing.</td>
<td>Event issued when an error is encountered creating, reading, or writing an operating system file.</td>
</tr>
<tr>
<td>NTWK_ERR</td>
<td>Application network connection error.</td>
<td>Minor. Auto-clearing.</td>
<td>Unable to connect to SQL/Application server, connection to server gone down, etc.</td>
</tr>
<tr>
<td>RPCSPACE</td>
<td>Critical database resource error.</td>
<td>Critical. Non auto-clearing.</td>
<td>Event issued by inv_rpc() and RPC calls if the database full, transaction logs full, database corrupt, etc.</td>
</tr>
<tr>
<td>SRVOBJER</td>
<td>Open Server Application object (Msg Queue, Mutex, etc.) access error.</td>
<td>Minor. Auto-clearing.</td>
<td>Unable to create, remove, lock, unlock, or access Open Server Object (Mutex, message queue, etc.).</td>
</tr>
<tr>
<td>SYS_INFO</td>
<td>General system information notification.</td>
<td>None.</td>
<td>System information such as thread spawning, process startup, and graceful shutdown, WO Timeouts, Host CLLI in Busy State, and administrative flushing of data from memory.</td>
</tr>
<tr>
<td>SYS_TERM</td>
<td>Process termination event.</td>
<td>Critical. Auto-clearing.</td>
<td>This event is called by an application process when an internal condition is encountered. The event causes the application process to log the system event in the database and then terminate the application process in an orderly manner.</td>
</tr>
<tr>
<td>SYS_WARN</td>
<td>General system warning.</td>
<td>Minor. Auto-clearing.</td>
<td>Warning event, not an error situation. This event can be used to indicate errors or exceptions in external systems, for example, Port Bind Failure, Host LU6.1/LU6.2 Bridge Down, NE in Maintenance Mode, and Stored Procedure Deadlock Victim.</td>
</tr>
<tr>
<td>UNIX_ERR</td>
<td>UNIX system call error.</td>
<td>Minor. Auto-clearing.</td>
<td>Event issued if application encounters UNIX system call error. For example, the NEP is unable to create a file to insert the NE response log or the SRP is unable to create a file for WO completions.</td>
</tr>
</tbody>
</table>

### SARM System Events

Table 3–4 lists and describes the events that the SARM can issue.
### Table 3–4  **SARM System Events**

<table>
<thead>
<tr>
<th>Event</th>
<th>Static Text</th>
<th>Alarm Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUT_ERR</td>
<td>WO routing error.</td>
<td>Major. Auto-clearing.</td>
<td>Notifies when the SARM process is unable to determine the correct NE to which a particular WO should be routed. The routing data (MCLI or DN) is included in the event text.</td>
</tr>
<tr>
<td>WOINPROC</td>
<td>WOs in progress longer than specified threshold.</td>
<td>Informational. Determined by individual customer.</td>
<td>Informs when one or more WOs are in progress for more than the specified threshold time.</td>
</tr>
</tbody>
</table>

### Control Server Events

*Table 3–5* lists and describes the events that the Control server can issue.

### Table 3–5  **Control Server Events**

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APP_ERR</td>
<td>ASAP application start-up error.</td>
</tr>
<tr>
<td>APP_STRT</td>
<td>ASAP application local or remote start-up.</td>
</tr>
<tr>
<td>APP_STOP</td>
<td>ASAP application local or remote shutdown.</td>
</tr>
</tbody>
</table>

### NEP System Events

*Table 3–6* lists and describes the events that the NEP can issue.

### Table 3–6  **NEP System Events**

<table>
<thead>
<tr>
<th>Event</th>
<th>Static Text</th>
<th>Alarm Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIND_ERR</td>
<td>Port binding error event.</td>
<td>Minor. Auto-clearing.</td>
<td>Unable to allocate device to connect to NE if, for instance, the maximum number of connections to the NE is exceeded. Each device in ASAP has a command processor thread that is always running. When there is a connection request for an NE, the session manager tries to obtain an enabled and unbinded (available) device. If the session manager cannot obtain such a device, the session manager will throw a BIND_ERR event.</td>
</tr>
<tr>
<td>CONN_ERR</td>
<td>NE connection error event.</td>
<td>Minor. Auto-clearing.</td>
<td>NE connection attempt failed.</td>
</tr>
<tr>
<td>DIAL_ERR</td>
<td>Dialup error event.</td>
<td>Minor. Auto-clearing.</td>
<td>The dial-up program to connect to NE has failed. After a connection is established to a dialup type device, but before the NE_LOGIN (or LOGIN) is performed, a state table named DIALUP is executed. If the DIALUP state table fails, then DIAL_ERR event is thrown.</td>
</tr>
</tbody>
</table>
Configuring and Reading Log and Diagnostic Files

There are two sets of files maintained by ASAP applications for monitoring and troubleshooting purposes: log files and diagnostic files. These files are created by ASAP daily or whenever an application starts up.

**Note:** If you write to diagnostic or log files while ASAP is running, you will cause the associated ASAP application to malfunction.

Figure 3–2 shows the sources of log and diagnostic information.
About Log Files

Log files contain high-level error messages. These files are created by ASAP daily or whenever a server starts up. Log files are located in the directory $LOGDIR (where $LOGDIR is $ASAP_BASE/DATA/logs) under appl_name.log, where appl_name is the appropriate client or server application.

Every diagnostic call made in ASAP has an associated diagnostic level denoting the importance of the diagnostic message. A specific diagnostic level is also applied to every application process so that only messages with a level greater than, or equal to, the configured level are written to the diagnostic file. This arrangement facilitates the configuration of the diagnostic logging without recompiling the application. For more information on diagnostic levels, see "Server Diagnostic Levels".

Every application executes with the diagnostic level specified in the Application Process Table. The diagnostic level of an application server can be modified dynamically while the server is running by means of an administrative RPC sent using the asap_utils RPC interface.

About Diagnostic Files

Diagnostic files include all low-level activities for each application. These files are created by ASAP daily or whenever a server starts up. Diagnostic files are located in the directory $LOGDIR/yyyyymmdd under the name appl_name.diag, where yyyyymmdd is the date for which you want to view the diagnostics and appl_name is the appropriate client or server application.
The diagnostic file rolls over to a new file whenever the maximum diagnostic file size is reached. This file size is configurable using the `MAX_DIAG_FILE` configuration parameter. You can also configure the number of diagnostic files to be created. This field, together with the maximum file size, limits the amount of disk space that may be used by diagnostic files. When the file limit is reached, the API closes the existing file and opens a new one.

The value of the `MAX_DIAG_FILE` parameter may depend on the level of diagnostic messages written by the application and the available disk space.

### Using ASAP Diagnostic Tools

The `diag_filt` and `diag_filt1` scripts are diagnostic tools contained in the `$SCRIPTS` directory allow you to view the diagnostic files to filter out the diagnostic messages from a particular thread. You can use this feature to follow the operation of just one thread. Usually, however, it is the interaction of multiple threads with each other that is of most interest to programmers.

```
diag_filt diagnostic_file thread_number [thread_number ...] > output_file
```

where:

- `diagnostic_file` is the name of the diagnostic file.
- `thread_number` is the thread number
- `output_file` is the name of an output file

For example, to filter out the diagnostic messages of thread number 7 and 21 from the `SARM.diag` file and insert the result into the `7_21.out` file, enter the following:

```
diag_filt SARM.diag 7 21 > 7_21.out
```

### Configuring and Reading WebLogic-based Log and Diagnostic Files

ASAP uses log4j to generate and manage the log messages from ASAP's WebLogic components. It does not affect any logs generated by ASAP's C-based (SARM, NEP, Control (CTRL) server) components. The generated log messages are stored in the `asap.log` file located in `WebLogic_domain`.

Through the use of log4j, you can develop and maintain a logging strategy that minimizes the overall impact of logging operations on the application's resources. log4j does this by letting you control the volume of log messages generated.

You also have the ability, when necessary, to dynamically change the level and detail of the logged messages. This feature helps you to, for example, increase the level and detail of logged messages to help analyze performance problems within a production environment.

To read more about log4j, refer to Apache Web site:

http://logging.apache.org/log4j

### Defining Severity Levels

To control the volume of log messages generated and written to the output destinations (the console and the WebLogic log file, which are referred to as Appenders by log4j), you assign severity levels to the various areas of the application that generate their own, discreet messages (these areas are referred to as Categories by log4j). If an event occurs inside a given category that triggers a message below the
severity level assigned to the category (that is, less severe than the assigned level), log4j does not generate the message.

Example

If you assigned the severity level Warning to a given category, log4j does not generate any messages for that category that are flagged in the ASAP code as Debug or Info level messages. log4j will, however, generate all messages that are flagged as Warning, Error, and Fatal.

You can further control the number of messages written to the output destinations, or Appenders, by also assigning a severity level to them. When you assign a severity level to an Appender, it rejects messages below that severity level, even if log4j passes the message to it.

Example

If you configure the console to the Warning severity level, and one of the Categories is configured with the Info level, the console will not display the Info message, even though log4j generates the message and passes it on to the console, because the message’s severity level falls below the threshold for which the Appender is configured. If, however, that same Category later generates an Error level message, the console will accept and display the message, because it carries a severity level equal to or higher than the console’s threshold.

By default, the console and the WebLogic log file accept all error messages, from the least severe to the most severe.

The Levels

The following is an ascending list of the severity levels, starting with the least severe:

- Debug – (Least severe) Designates fine-grained informational events that are most useful to debug an application
- Info – Designates informational messages that highlight the progress of the application at coarse-grained level
- Warning – Designates potentially harmful situations
- Error – Designates error events that might still allow the application to continue running
- Fatal – (Most severe) Designates very severe error events that will presumably lead the application to abort

Configuring the Severity Levels

There are two methods by which you can configure the severity levels:

- log4j.xml – The log4j.xml file is located in asap$ENV_/LD.ear/APP-INF/lib/asaplibcommon.jar. This method allows you to modify the log4j configuration (like log message format, name of the log file, volume of the log files, how many are saved, and so on) and requires you to stop, then restart the servers before the changes will take effect. For more information, see “Configuring the log4j.xml File”.
- log4jAdmin – The log4jAdmin Web page lets you to dynamically change the severity levels while the servers are running. The changes take effect immediately. As the changes are persisted to the database the severity levels remain at the new level when you restart the server. For more information, see “Using the log4jAdmin Web Page”.

3-14  ASAP System Administrator’s Guide
The logging levels configured in the database take precedence; all other configuration is controlled by the log4j.xml configuration file embedded inside asap$ENV_ID.ear:APP-INF/lib/asaplibcommon.jar.

Configuring the log4j.xml File

Use this method to modify the log4j configuration as described in "Configuring the Severity Levels". To change the log4j configurations later, you must stop the server, modify the log4j.xml file, then restart the server. In most cases, therefore, if you need to modify the logging levels, you should use the log4jAdmin Web page, as described in "Using the log4jAdmin Web Page".

There are two sections of the log4j.xml file that you need to look at when configuring this file:

- **Appenders** – This section defines the output destination for the messages. At installation, the Appenders section contains two entries:
  - **Console** – From this entry, you control the level of messages that the WebLogic console accepts.
  - **WebLogic** – From this entry, you control the level of messages that the WebLogic log file accepts.

- **Categories** – This section contains references to all of the ASAP categories that generate messages and gives you the ability to control the level of messages they generate.

To configure the log4j.xml file:

1. Unpack the asap.ear or the srt.ear file, depending on the application for which you are configuring logging parameters.
   - The asap.ear file is found in the $ASAP_BASE/lib directory. After the .ear file is unpacked, unpack APP-INF/lib/asaplibcommon.jar. The log4j.xml file is located in the root directory.
   - The srt.ear files is found in the $ASAP_BASE/SRT/lib directory. After the .ear file is unpacked, unpack APP-INF/lib/asaplibcommon.jar. The log4j.xml file is located in the root directory.

2. In the Appenders section of the log4.xml file, search for the following string, which appears at the top of the Appenders section:

   ```
   <!-- Append messages to the console -->
   ```

   The Console entry governs what level of messages are written to the console.

3. If necessary, change the threshold level. By default, it is configured to DEBUG, allowing the console to display all messages sent to it. If you want to restrict the number of messages displayed, change the threshold entry to the severity level appropriate for your installation (see "Severity levels" above, for a description of the severity levels).

   In the example below, the level is changed from DEBUG to INFO.

   Before
   ```
   <param name="Threshold" value="DEBUG"/>
   ```

   After
   ```
   <param name="Threshold" value="INFO"/>
   ```

4. In the Appenders section, search for the following string:
5. If necessary, change the threshold level as described in Step 3.

6. Go to top of the file and search for the following string:

   category name

   This takes you to the first category entry in the file.

7. Review each of the categories in this section, changing the severity level where necessary. In the example below, the level is changed from INFO to WARNING.

   Before

   <category name="com.mslv.system"> <priority value="info"/> </category>

   After

   <category name="com.mslv.system"> <priority value="warning"/> </category>

8. When you finish updating the categories, save and close the log4j.xml file.

9. Repack the .ear file.

10. Redeploy the .ear file:

    ■ For Java Service Request Processor (JSRP):

        In $ASAP_BASE/lib, run the following commands:

        ModDeployDescriptor -u <WebLogic Admin Id>
        java weblogic.Deployer -adminurl http://$WLS_HOST:$WLS_PORT
        -user $WLS_USER -password $WLS_PASSWORD -name asap$ENV_ID -remove
        java weblogic.Deployer -adminurl http://$WLS_HOST:$WLS_PORT
        -user $WLS_USER -password $WLS_PASSWORD -upload
        -source $ASAP_BASE/lib/asap$ENV_ID.ear -name asap$ENV_ID
        -targets $TARGET_WLS_SERVER -activate

    ■ For SRT:

        In $ASAP_BASE/SRT run the following commands:

        ant -f install.xml undeploy
        ant -f install.xml deploy

Using the log4jAdmin Web Page

Use log4jAdmin Web page to check the current logging levels or to change the logging levels dynamically.

**Note:** You can only use this method to change the severity levels of the Categories. To change the Appender’s levels, you must reconfigure the log4j.xml file. See “Configuring the log4j.xml File” for an explanation of the Categories, the Appenders, and how to configure the XML file.

The changes you make to the logging severity levels using this method are persisted to the database in the table tbl_code_list on the Control server.
Checking the Current Logging Levels

You can use the Filter Loggers feature at the top of the page to check the logging level of specific categories or subcomponents. If you know the name of the category or subcomponent that you want to check, you can use the filter to display only that category, or related, categories.

To check logging levels:

1. Open the log4jAdmin Web page by entering the following path in the browser's address line (the URLs are case sensitive):
   - JSRP:
     http://BEA_HOST:BEA_PORT/ASAP_ENVID/log4jAdmin.jsp
     or
     https://BEA_HOST:BEA_SSL_PORT/ASAP_ENVID/log4jAdmin.jsp
   - SRT:
     http://BEA_HOST:BEA_PORT/ASAP_ENVID/SRT/log4jAdmin.jsp
     or
     https://BEA_HOST:BEA_SSL_PORT/ASAP_ENVID/SRT/log4jAdmin.jsp

2. In the Filter Loggers field, enter the beginning of the name or a part of the name.

3. Do one of the following:
   - Click Begins With to filter on the beginning of the name.
   - Click Contains to filter on part of the name.
     The list displays the categories and subcategories that match the entry in the Filter Loggers field.
   - (Optional) To change the logging level do the following:
     a. Scan the entries in the left-hand column and find the category or sub-component which you want to change.
     b. Scan across the row to the severity levels. The level that currently is selected is highlighted in a different color from the other levels and appears in the Effective Level column.
     c. Click the logging level which you want to change:
        To change an entire category, click the category name.
        To change the subcomponent, click the sub-component name.
        The change takes place immediately.
     d. When you finish making the changes, close the page.
Enabling Stored Procedure Error Messages

To enable stored procedure error messages for a sqlplus session, use the following procedure:

1. From a UNIX terminal, source your ASAP `ASAP_home/Environment_Profile`.
   ```bash
   ./Environment_Profile
   ```

2. Log into your sqlplus session for the database you want to run stored procedures on. For example:
   ```sql
   sqlplus $CTRL/password
   ```
   Where `password` is the password for your ASAP server database schema.

3. Run the following command to enable stored procedure error messages.
   ```sql
   set serveroutput on;
   ```
   For an error message example, see the following error message in bold:
   ```sql
   var retval number;
   exec :retval := SSP_new_comm_param('T', 'TEL_HOST', 'COMMON_DEVICE_CFG', 'HOST_USERID', 'asapXXX', 'userid');
   Host TEL_HOST For Device Type T And Parameter HOST_USERID Does Not Exist, No Comm Param Inserted.
   
   PL/SQL procedure successfully completed.
   
   SQL> print retval;
   
   RETVAL
          ---------
              0
   ```
This chapter describes ways to improve Oracle Communications ASAP performance.

**About Improving ASAP Performance**

This chapter is intended to aid those who have prior knowledge of the ASAP configuration and UNIX operating systems. Before starting the tuning exercises described in this chapter, you should be familiar with the following items:

- Location of ASAP diagnostic files and the UNIX utilities that are used to view and manipulate them such as grep, tail, pg, top, vmstat, sar, prstat, glance (on HP), and so on.
- Location of the ASAP configuration files (ASAP.cfg, ASAP.properties, Environment_Profile, NEP.jinterpreter, config.xml, startWebLogic.sh), how to use an editor such as vi, how to modify the configuration files, the layout of configuration files, for example, server specific versus global variables.
- How to use UNIX utilities, such as top and sar to monitor the resources being used by ASAP.

For more information, consult your system’s online documentation about UNIX utilities or the ASAP documentation.

**Tuning ASAP Performance with Pre-tuned ASAP Configuration Files**

Stock configuration files are shipped with the product for varying sizes of ASAP installations. You can choose to use one of these versions of the ASAP.cfg file as-is, or as a starting point for additional changes.

This section provides you with tools and guidelines to properly select a default configuration.

**About Pre-tuned ASAP Configurations**

ASAP ships with pre-tuned configurations for small, medium and large installations. (Definitions of small, medium and large configurations in this chapter are consistent with those in the planning chapter of the ASAP Installation Guide.)

Details are given below for these stock configurations, with sample performance figures. Your results will vary depending on your hardware and other configuration choices you may have implemented.
Installing a Pre-tuned Configuration
The pre-tuned configuration files are generated by a script which is run as part of the installation process for new installs only. The files are placed in `ASAP_HOME/samples/sample_configs`, where `ASAP_HOME` is the directory in which ASAP is installed.

Table 4–1 lists and described the pre-tuned configuration files.

<table>
<thead>
<tr>
<th>Default Configuration File</th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASAP.cfg file</td>
<td>ASAP.cfg.small</td>
<td>ASAP.cfg.medium</td>
<td>ASAP.cfg.large</td>
</tr>
<tr>
<td>ASAP.properties file</td>
<td>ASAP.properties.small</td>
<td>ASAP.properties.medium</td>
<td>ASAP.properties.large</td>
</tr>
<tr>
<td>Environment_Profile file</td>
<td>Environment_Profile.small</td>
<td>Environment_Profile.medium</td>
<td>Environment_Profile.large</td>
</tr>
<tr>
<td>Performance</td>
<td>50,000 orders/day</td>
<td>500,000 orders/day</td>
<td>20.95 orders/sec</td>
</tr>
<tr>
<td></td>
<td>1 order/sec</td>
<td>11.5 orders/sec</td>
<td>146.65 ASDL/sec</td>
</tr>
<tr>
<td></td>
<td>7 ASDL/sec</td>
<td>80.5 ASDL/sec</td>
<td></td>
</tr>
<tr>
<td>DB connections</td>
<td>52</td>
<td>89</td>
<td>145</td>
</tr>
<tr>
<td>Log level</td>
<td>SANE</td>
<td>SANE</td>
<td>SANE</td>
</tr>
</tbody>
</table>

Using a Pre-tuned Configuration with a New ASAP Installation
After the installation of ASAP is complete, back up the following files:
- `ASAP_HOME/config/ASAP.cfg`
- `ASAP_HOME/ASAP.properties`
- `ASAP_HOME/Environment_Profile`

Replace these files with the appropriate three files corresponding to the desired pre-tuned configuration from `ASAP_HOME/samples/sample_configs`.

Generating Pre-tuned Configuration Files
For upgrade installations, the pre-tuned configuration files are not generated. However, the generation script can be run manually:

```bash
$ASAP_BASE/scripts/generate_sample_configs.ksh $ASAP_BASE
```

This will generate the pre-tuned configuration files to `ASAP_HOME/samples/sample_configs`.

Merging Pre-tuned File Settings into an Existing Installation
If you have already made changes to the `ASAP.cfg`, `ASAP.properties`, or `Environment_Profile` files, you will have to manage the differences between your altered files and the pre-tuned files. For example, simply copying over the pre-tuned files will overwrite any changes you have made. To merge the pre-tuned file settings with your existing settings, compare the differences between your existing files and the pre-tuned files and manually add in the changes from the pre-tuned files.

Example Pre-tuned Configuration Performance
Table 4–2, Table 4–3 and Table 4–4 provide example performance results on hardware. Your actual results will vary.
### Table 4-2 Small Installation Pre-tuned Configuration Performance

<table>
<thead>
<tr>
<th></th>
<th>CPU, % of server</th>
<th>CPU, % of 1 proc</th>
<th>RAM, MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>V880 SARM</td>
<td>0.79</td>
<td>3.16</td>
<td>43</td>
</tr>
<tr>
<td>V880 NEP</td>
<td>1.39</td>
<td>5.56</td>
<td>37</td>
</tr>
<tr>
<td>V880 JENEP</td>
<td>0.64</td>
<td>2.56</td>
<td>181</td>
</tr>
<tr>
<td>sclust01</td>
<td>4.73</td>
<td>9.46</td>
<td>261</td>
</tr>
<tr>
<td>Compaq</td>
<td>1.65</td>
<td>13.2</td>
<td>N/A</td>
</tr>
<tr>
<td>N/A</td>
<td></td>
<td></td>
<td>350</td>
</tr>
<tr>
<td>N/A</td>
<td></td>
<td></td>
<td>261</td>
</tr>
<tr>
<td>N/A</td>
<td></td>
<td></td>
<td>611</td>
</tr>
</tbody>
</table>

### Table 4-3 Medium Installation Pre-tuned Configuration Performance

<table>
<thead>
<tr>
<th></th>
<th>CPU, % of server</th>
<th>CPU, % of 1 proc</th>
<th>RAM, MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>V880 SARM</td>
<td>12.86</td>
<td>51.44</td>
<td>52</td>
</tr>
<tr>
<td>V880 NEP</td>
<td>15.63</td>
<td>62.52</td>
<td>44</td>
</tr>
<tr>
<td>V880 JENEP</td>
<td>7.49</td>
<td>29.96</td>
<td>376</td>
</tr>
<tr>
<td>sclust01</td>
<td>24.23</td>
<td>48.46</td>
<td>750</td>
</tr>
<tr>
<td>Compaq</td>
<td>10.14</td>
<td>81.12</td>
<td>N/A</td>
</tr>
<tr>
<td>N/A</td>
<td></td>
<td></td>
<td>550</td>
</tr>
<tr>
<td>N/A</td>
<td></td>
<td></td>
<td>750</td>
</tr>
<tr>
<td>N/A</td>
<td></td>
<td></td>
<td>1300</td>
</tr>
</tbody>
</table>

### Table 4-4 Large Installation Pre-tuned Configuration Performance

<table>
<thead>
<tr>
<th></th>
<th>CPU, % of server</th>
<th>CPU, % of 1 proc</th>
<th>RAM, MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>V880 SARM</td>
<td>28.71</td>
<td>114.84</td>
<td>64</td>
</tr>
<tr>
<td>V880 NEP</td>
<td>13.04</td>
<td>52.16</td>
<td>41</td>
</tr>
<tr>
<td>V880 JENEP</td>
<td>9.22</td>
<td>36.88</td>
<td>374</td>
</tr>
<tr>
<td>N/A</td>
<td>18.03</td>
<td>72.12</td>
<td>43</td>
</tr>
<tr>
<td>N/A</td>
<td>13.73</td>
<td>54.92</td>
<td>375</td>
</tr>
<tr>
<td>sclust01</td>
<td>51.54</td>
<td>103.08</td>
<td>750</td>
</tr>
<tr>
<td>Compaq</td>
<td>18.74</td>
<td>147.76</td>
<td>N/A</td>
</tr>
<tr>
<td>N/A</td>
<td></td>
<td></td>
<td>1050</td>
</tr>
<tr>
<td>N/A</td>
<td></td>
<td></td>
<td>750</td>
</tr>
<tr>
<td>N/A</td>
<td></td>
<td></td>
<td>1800</td>
</tr>
</tbody>
</table>
Troubleshooting and Monitoring ASAP Performance

The ASAP utility script (asap_utils) is a menu driven utility that provides access to a set of monitoring and troubleshooting tools for ASAP. These tools provide information that is relevant to the performance tuning of an ASAP system.

This section describes the ASAP utility script option 109 for Real-Time System Monitoring and its uses with respect to troubleshooting and monitoring ASAP performance. This option calls the Sysmon tool that gathers performance data and stores this data in the ASAP_home/DATA/logs directory (where ASAP_home is the location of the ASAP installation).

---

**Caution:** Running the Sysmon tool against an ASAP system in production causes a 20 to 25% degradation in system performance.

---

Additional information about the Sysmon tool is contained in the section describing the ASAP utility script in ASAP Server Configuration Guide.

The WebLogic Administration Console can be used to monitor the Java Service Request Processor (JSRP).

Understanding Sysmon Output Files

Although Sysmon output files can be very large, only a small part of the information contained in the file is used to tune the system. During the tuning process, the main areas of interest in the Sysmon output file are:

- Connection Pool
- Message Queue

Troubleshooting the Connection Pool

The following Sysmon sample output illustrates the cause of a slowdown in production. When the number in the “all connections in use (count)” row grows continually, the system is waiting for a free connection and is not being used to its maximum capacity.

```
<table>
<thead>
<tr>
<th>Description</th>
<th>Count</th>
<th>Total</th>
<th>Min</th>
<th>Max</th>
<th>Average</th>
<th>Mean</th>
<th>Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Pool</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>add connections</td>
<td>2</td>
<td>4.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>0.0</td>
</tr>
<tr>
<td>all connections in use (count)</td>
<td>250</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>allocate wait-time</td>
<td>34454</td>
<td>1276</td>
<td>0.1</td>
<td>26.2</td>
<td>37.1</td>
<td>4386.3</td>
<td>4355.4</td>
</tr>
<tr>
<td>connections in use</td>
<td>34454</td>
<td>1817.0</td>
<td>1.0</td>
<td>5.3</td>
<td>5.9</td>
<td>5.9</td>
<td>1.3</td>
</tr>
<tr>
<td>pool size</td>
<td>6</td>
<td>42.0</td>
<td>5.0</td>
<td>9.0</td>
<td>7.0</td>
<td>7.3</td>
<td>0.7</td>
</tr>
<tr>
<td>Remove connection (count)</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Returned (count)</td>
<td>34453</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

To correct this situation, do one of the following:

Increase the value of the configuration variable, APPL_POOL_SIZE to make more connections available.
Enable the auto-tuning option of ASAP using the configuration variable, CPM_OPTIMIZE_POOLS in the ASAP configuration file (ASAP.cfg), so that connections can be added automatically when necessary.

If a stored procedure fails, the thread running the procedure goes to sleep for the time determined by the RPC_RETRY_SLEEP configuration parameter. ASAP then tries to execute the procedure with the number of times determined by the RPC_RETRY_COUNT configuration parameter. All of these attempts may fail. Since the thread cannot be released for the entire duration of this error retry process, poor performance is reported, increasing the number of connections in the use and long waiting times.

Table 4–5 lists and describes terms used in the Sysmon connection pool output.

<table>
<thead>
<tr>
<th>Column/Row Heading</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row</td>
<td>Add Connections</td>
<td>Only appears if auto-tuning is enabled. Provides information regarding the addition of connections to the pool by the auto-tuning process during the time period that the Sysmon data was collected.</td>
</tr>
<tr>
<td>Row</td>
<td>All Connections In Use</td>
<td>Counts the number of times the system had to wait for connections to become available in the pool. It only appears if this condition occurred during the time period that the Sysmon data was collected.</td>
</tr>
<tr>
<td>Row</td>
<td>Allocate Wait-time</td>
<td>Measures the amount of time it took ASAP to acquire a connection from the pool during the time period that the Sysmon data was collected. This measurement is highly correlated with the “all connections in use” parameter, as ASAP must wait until a connection becomes free if none are available.</td>
</tr>
<tr>
<td>Row</td>
<td>Connections In Use</td>
<td>Measures the number of connections in use by ASAP during the time period that the Sysmon data was collected, and calculated as connections are retrieved from the pool.</td>
</tr>
<tr>
<td>Row</td>
<td>Pool Size</td>
<td>Only appears if auto-tuning is enabled. Provides information regarding the size of the connection pool during the time period that the Sysmon data was collected.</td>
</tr>
<tr>
<td>Row</td>
<td>Remove Connection</td>
<td>Only appears if auto-tuning is enabled. Counts the number of times connections were removed from the pool by the auto-tuning process during the time period that the Sysmon data was collected.</td>
</tr>
<tr>
<td>Row</td>
<td>Returned</td>
<td>Counts the number of times a connection was returned to the pool for re-use during the time period that the Sysmon data was collected.</td>
</tr>
<tr>
<td>Column</td>
<td>Count</td>
<td>For rows where data is presented for Min, Max, Average, Mean and Deviation, this number represents the number of data points that were used to calculate the listed statistics. For all other rows, count presents the data collected for the row. In other words, count represents the sample size.</td>
</tr>
<tr>
<td>Column</td>
<td>Total</td>
<td>The total of the collected values for the corresponding row heading.</td>
</tr>
<tr>
<td>Column</td>
<td>Min</td>
<td>The lowest collected value for the corresponding row heading.</td>
</tr>
<tr>
<td>Column</td>
<td>Max</td>
<td>The highest collected value for the corresponding row heading.</td>
</tr>
<tr>
<td>Column</td>
<td>Average</td>
<td>Calculated as the Total / Count for the corresponding row heading.</td>
</tr>
<tr>
<td>Column</td>
<td>Mean</td>
<td>An estimate of the mean for the corresponding row heading. Calculated as the (Min + Max + 4 * Average)/6.</td>
</tr>
<tr>
<td>Column</td>
<td>Deviation</td>
<td>An estimate of the standard deviation for the corresponding row heading. Calculated as the Min + Max / 6.</td>
</tr>
</tbody>
</table>
Troubleshooting the Message Queue

The following Sysmon output section demonstrates the use of the Sysmon tool, and illustrates a condition which can degrade the responsiveness of ASAP.

The Group Manager Msg Q size has maximum, average, and mean sizes that are greater than required. For tunable message queues in ASAP, a non-zero size less than 5 is optimal, but not always achievable. In this case, messages are building up as they wait for processing by the group manager threads.

<table>
<thead>
<tr>
<th>Tuning - Message Queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>ASDL Provision Queue</td>
</tr>
<tr>
<td>message read wait time</td>
</tr>
<tr>
<td>messages sent (count)</td>
</tr>
<tr>
<td>queue idle-time (ms)</td>
</tr>
<tr>
<td>queue size (count)</td>
</tr>
</tbody>
</table>
...

| Group Manager Msg Q   |       | ------ | ---- | ---- | ------- | ----- | --------- |
| message read wait time| 2469  | 38896.8| 2.5  | 290.5| 15.75   | 59.33 | 48.0      |
| messages sent (count) | 2469  |        |      |      |         |       |           |
| queue idle-time (ms)  | 2469  | 1870.6 | 0.0  | 718.3| 0.758   | 120.22| 119.72    |
| queue size (count)    | 2469  | 27184.0| 0.0  | 27.1 | 11.01   | 11.86 | 4.5       |
...

To decrease the length of the queue and enhance the responsiveness of the system, increase the number of group manager threads in the system.

This is a simplified example of how to use the Sysmon tool to tune a message queue. Refer to the sections on the SRP, NEP, and SARM tuning for further details.

Table 4–6 lists and describes terms used in Sysmon output.

<table>
<thead>
<tr>
<th>Column/Row Heading</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row</td>
<td>Message Read Wait-time</td>
<td>The time, in milliseconds, that a message was sitting in the message queue waiting for processing, during the time period that the Sysmon data was collected.</td>
</tr>
<tr>
<td>Row</td>
<td>Messages Sent</td>
<td>The number of messages put into the queue during the time period that the Sysmon data was collected.</td>
</tr>
<tr>
<td>Row</td>
<td>Queue Idle Time</td>
<td>The time, in milliseconds, required for a message to enter the queue for processing during the period when the Sysmon data was collected.</td>
</tr>
<tr>
<td>Row</td>
<td>Queue Size</td>
<td>The number of messages in the queue during the period when the Sysmon data was collected. The reported number is decreased when the queue length is recalculated after a message leaves the queue.</td>
</tr>
<tr>
<td>Column</td>
<td>Count</td>
<td>The number of messages added to the queue.</td>
</tr>
<tr>
<td>Column</td>
<td>Total</td>
<td>The total collected values for the corresponding row heading.</td>
</tr>
<tr>
<td>Column</td>
<td>Min</td>
<td>The minimum collected value for the corresponding row heading.</td>
</tr>
<tr>
<td>Column</td>
<td>Max</td>
<td>The maximum collected value for the corresponding row heading.</td>
</tr>
<tr>
<td>Column</td>
<td>Average</td>
<td>The total of the collected values for the corresponding row heading.</td>
</tr>
<tr>
<td>Column</td>
<td>Min</td>
<td>The lowest collected value for the corresponding row heading.</td>
</tr>
</tbody>
</table>
Manually Tuning ASAP Performance

The performance of an ASAP system is governed by the available hardware, installation, and configuration decisions made during the initial installation phase. Due to the multi-threaded nature of ASAP, fine-tuning the system to will help you to obtain the maximum benefits from the allocated resources.

This section provides you with tools and guidelines to tune your ASAP system in a short period of time. It covers the following topics:

- A recommended approach to tuning.
- A list of system limits that must be monitored to ensure that they are not exceeded during tuning.
- Guidelines to tune the JSRP/ SRP, SARM, and NEP processes.

Tuning Guidelines

If you wish to go beyond the provided pre-tuned configurations, there are many ways to tune an ASAP system. However, the following technique has been verified by the internal Oracle Communications testing team. You can use it to optimize a simple ASAP configuration in less than half a day. A simple configuration consists of all ASAP components residing in the same system with small numbers of individual components, for example, fewer than five NEPs and one or two SRPs.

The following steps are the order in which the tuning process is carried out:

1. Setting a Target

Select a performance target that is based on realistic throughput (work orders (WOs) per second) or resource consumption early in the process. Without a goal, an iterative process, such as tuning, could continue indefinitely.

2. Using Simple Work Orders

To achieve consistent results, use simple familiar WOs during the tuning process. Use a repeatable test and pick a scenario such as batches, or workflows. Once tuning is complete, verify the performance with realistic data.

3. Starting with Minimum Configuration Values

Start with minimum configuration values because it is easier to detect and correct bottlenecks than it is to determine where excess resources are being consumed.

4. Following Work Order Flow

The tuning process follows the same flow that WOs take through the system. Tuning starts at the SRP (that is CSRP or JSRP depending on your implementation) proceeds

Table 4–6 (Cont.) Sysmon Output Terms

<table>
<thead>
<tr>
<th>Column/Row Heading</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column</td>
<td>Max</td>
<td>The highest collected value for the corresponding row heading.</td>
</tr>
<tr>
<td>Column</td>
<td>Average</td>
<td>Calculated as the Total divided by the Count for the corresponding row heading.</td>
</tr>
<tr>
<td>Column</td>
<td>Mean</td>
<td>An estimate of the mean for the corresponding row heading. Calculated as the (Min + Max + 4 * Average)/6.</td>
</tr>
<tr>
<td>Column</td>
<td>Deviation</td>
<td>An estimate of the standard deviation for the corresponding row heading. Calculated as the Min + Max / 6.</td>
</tr>
</tbody>
</table>
through to the SARM and then to the NEP (that is CNEP or Java-enabled NEP (JNEP) depending on your implementation) and on to the NE, then it returns back through the same steps.

5. Checking for Bottlenecks

Bottlenecks that can develop as resources are shifted among the components which make up an ASAP system. Bottlenecks may occur in areas that were previously optimized. When you move to a new area of the system, you should review the servers that have already been tuned to ensure that their configurations have remained optimal. For example, if you tune the NEP after the SRP and SARM have been optimized, review the SRP and SARM after you have finished tuning the NEP to ensure that they have remained optimized.

Setting System Limits

During the tuning process, you must change configuration variable settings to levels that are higher than their defaults. These increases have two direct effects which you must monitor during the tuning process:

- Increased demands are placed on the hardware allocated to the ASAP system. You must use a utility, such as `top` to continually monitor the ASAP system in order to ensure that it is not consuming more resources than planned.
- If system limits are exceeded, increased ASAP resource consumption may cause errors to be reported in the systems diagnostic files. Monitor the diagnostic files closely during the tuning process so that these limits can be altered to higher values when required.

This section details errors which may appear in the diagnostic files and the configuration variables used to control system limits. Configuration variables are located in the `ASAP.cfg` file. The following are the configuration variables used for tuning.

- `APPL_POOL_SIZE`
- `CONTROL_POOL_SIZE`
- `MAX_CMD_DBPROCS`
- `MAX_CONNECTIONS`
- `MAX_CORE_DBPROCS`
- `MAX_MSGPOOL`
- `MAX_MSGQUEUES`
- `MAX_SERVER_PROCS`
- `MAX_THREADS`
- `MAX_ORDERS_IN_PROGRESS`
- `WO_AUDIT_LEVEL`

For more information on configuration variables, refer to the chapter describing configuration parameters in the `ASAP Server Configuration Guide`.

Determining the Size of Your System

Refer to the `ASAP Installation Guide` for detailed information on sizing small, medium and large system configuration.
Tuning Message Queue Guidelines

The optimum balance between throughput and response time for ASAP operation is achieved when all tunable message queues in the system are stable, short, and non-zero. When message queues are short and stable, threads operate more efficiently and are able to keep up with the flow of incoming messages.

Queue lengths depend on the quantity of threads that are either added or removed from the queue. To decrease the length of a queue, either decrease the rate at which messages are added to the queue or increase the rate at which they are removed. To increase the length of the queue, reverse the process.

Workload balancing is an end-to-end process. Bottlenecks can occur anywhere along the message processing path, reducing the message flow over the remainder of the path. For example, to avoid bottlenecks, the SRP must have a sufficient number of translation threads to handle the WO volume rate and enough SARM drivers to send WOs to the SARM as fast as they are generated. The SARM must also have enough Work Order Handler threads to handle the incoming Common Service Description Layer (CSDL) commands. By this point enough NEPs should be configured to efficiently secure the network elements (NEs). The following section guides you in tuning an ASAP system to achieve the ideal operation outlined above.

To tune an ASAP system:

1. Submit a representative batch of WOs into the system – Enough for about ten minutes of work.
2. Monitor the OS with `top` and `sar`. Make sure the system has enough idle CPU cycles to prevent being choked (approximately 15% idle).
3. Execute the `Sysmon` Tool on the appropriate server(s).
4. Examine the thread message queue section of the `Sysmon` output file for queues that are growing or that are consistently zero in length.
5. Modify the ASAP configuration variables which control the addition and/or removal rates for the queue.
6. Repeat steps 1 to 4 until the tunable message queues in the server have the desired queue lengths and then move onto the next server, for example, from the SRP to the SARM or from the SARM to the NEP.

To help you execute the tuning process, a standard set of information is presented for each message queue that you can tune in each ASAP component. This information includes:

- Name of the queue.
- Tool/Process used to monitor the queue. The tool used is: `asap_utils`, `Real-time System Monitoring` (option 109), `Target Server`.
- Addition rate configuration variable(s), which is the parameter that controls the rate at which messages are added to the queue (description, ASAP configuration variable, small/medium/large initial values).
- Removal rate configuration variable, which is the parameter that controls the rate at which messages are removed from the queue (description, ASAP configuration variable, small/medium/large initial values).

Table 4–7 is an example of this information in a table format.
Table 4–7 Queue Name

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool</td>
<td>asap_utils option server name</td>
</tr>
<tr>
<td>Parameter Controlling Message Addition Rate to Queue</td>
<td>description configuration variable</td>
</tr>
<tr>
<td></td>
<td>Variable size:</td>
</tr>
<tr>
<td></td>
<td>■ small</td>
</tr>
<tr>
<td></td>
<td>■ medium</td>
</tr>
<tr>
<td></td>
<td>■ large</td>
</tr>
<tr>
<td>Parameter Controlling Message Removal Rate from Queue</td>
<td>description configuration variable</td>
</tr>
<tr>
<td></td>
<td>Variable size:</td>
</tr>
<tr>
<td></td>
<td>■ small</td>
</tr>
<tr>
<td></td>
<td>■ medium</td>
</tr>
<tr>
<td></td>
<td>■ large</td>
</tr>
</tbody>
</table>

Tuning ASAP Server Message Queues

The following sections contains the ASAP servers that can be tuned:

- JSRP/SRT
- SRP
- SARM
- NEP
- JNEP
- WebLogic domain

Tuning JSRP and SRP Message Queues

The purpose of tuning the SRP is to provide WOs at a rate that creates an even flow to the downstream SARM process.

Figure 4–1 illustrates the schematic flow of the SRP.
Manually Tuning ASAP Performance

**Table 4–8**  Work Order Manager Queue

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool</td>
<td>asap_utils, Real-time System Monitoring option 109, SRP Server</td>
</tr>
<tr>
<td>Parameter Controlling Message Addition Rate to</td>
<td>Number of SRP Driver Threads in the SARM.</td>
</tr>
<tr>
<td>Queue</td>
<td><strong>MAX_SRP_DRIVERS</strong></td>
</tr>
<tr>
<td></td>
<td>Variable size:</td>
</tr>
<tr>
<td></td>
<td>■ small: 5</td>
</tr>
<tr>
<td></td>
<td>■ medium: 10</td>
</tr>
<tr>
<td></td>
<td>■ large: 25</td>
</tr>
<tr>
<td>Parameter Controlling Message Removal Rate</td>
<td>Number of WO Manager Threads</td>
</tr>
<tr>
<td>from Queue</td>
<td><strong>MAX_WO_MGRS</strong></td>
</tr>
<tr>
<td></td>
<td>Variable size:</td>
</tr>
<tr>
<td></td>
<td>■ small: 5</td>
</tr>
<tr>
<td></td>
<td>■ medium: 10</td>
</tr>
<tr>
<td></td>
<td>■ large: 25</td>
</tr>
</tbody>
</table>

Table 4–9 lists the SARM driver message queues.
Tips for Tuning the SRP

To tune the SRP, use the following:

- As an initial estimate, set the number of WO Manager Threads (MAX_WO_MGRS) in the SRPs equal to the number of configured MAX_SRP_DRIVERS threads in the SARM. The initial setting can be modified depending upon the complexity of the WO (number of CSDLs and parameters), and the number of events defined to return to the SRP. Monitor the WO Manager queue(s) in the SRP(s) and the SRP driver queue in the SARM to verify these configuration variables.

- The sum of all MAX_SARM_DRIVER threads in the operating SRPs must be equal to the number of configured MAX_WO_HANDLERS threads in the SARM.

- The number or existence of translation threads is entirely dependent upon the customized implementation of the SRP at your site. If multiple translation threads are available and the SARM Driver queue is consistently empty, consider increasing the number of translation threads. However, increasing these threads may have a negative affect on the downstream process. You must examine the system for downstream bottlenecks.

- You must disable all unnecessary notifications being sent to the SRP by updating tbl_asap_srp in the SARM database.

- During production, you can use sanity level diagnostics.

- If the save and dump WO features, which are enabled by using configuration variables SAVE_SARM_DATA and DUMP_WO_FLAG, are not needed, disable them.

Tuning SARM Message Queues

The purpose of tuning the SARM is to:

- Provide the Atomic Service Description Layer (ASDL) commands to the NEPs.

- Send event notices back to the SRPs at an even rate to both the upstream and downstream processes.

Since only one SARM process exists in an ASAP system, the performance of the SARM cannot be enhanced by spreading the load across multiple Central Processing Units (CPUs) or systems. Therefore, the SARM must be well tuned to get high performance from ASAP.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool</td>
<td>asap_utils, Real-time System Monitoring option 109, SRP Server</td>
</tr>
<tr>
<td>Parameter Controlling Message Addition Rate to Queue</td>
<td>Number of Translation Threads (implementation dependent).</td>
</tr>
</tbody>
</table>
| Parameter Controlling Message Removal Rate from Queue | Number of SARM Driver Threads MAX_SARM_DRIVER Variable size:  
  - small: 5  
  - medium: 10  
  - large: 25 |
Figure 4–2 illustrates the flow of messages through the SARM.

Table 4–10 lists and describes the group Mgr message queue.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool</td>
<td>asap_utils Real-time System Monitoring option (109), SARM Server</td>
</tr>
</tbody>
</table>
Table 4–10 (Cont.) Group Mgr Message Queue

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Controlling Message Addition Rate to Queue</td>
<td>The number of WO Handler threads and number of NEPs in the system are fixed for the purpose of Group Manager message queue tuning. Do not configure them at this time.</td>
</tr>
</tbody>
</table>
| Parameter Controlling Message Removal Rate from Queue | Number of Group Manager Threads.  
MAX_GROUP_MGRS  
Variable size:  
- small: 1  
- medium: 5  
- large: 10 |

Table 4–11 lists and describes the WO Mgr message queues.

Table 4–11 Work Order Mgr Message Queues

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool</td>
<td>asap_utils, Real-time System Monitoring option (109), SARM Server</td>
</tr>
<tr>
<td>Parameter Controlling Message Addition Rate to Queue</td>
<td>The number of WO Handler Threads and Number of NEPs in the system are fixed for the purpose of WO Manager message queue tuning. Do not configure them.</td>
</tr>
</tbody>
</table>
| Parameter Controlling Message Removal Rate from Queue | Number of WO Manager Threads  
MAX_WO_MGRS  
Variable size:  
- small: 5  
- medium: 10  
- large: 25 |

Table 4–12 lists and describes the WO provision queue.

Table 4–12 Work Order Provision Queue

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool</td>
<td>asap_utils, Real-time System Monitoring option (109), SARM Server</td>
</tr>
</tbody>
</table>
| Parameter Controlling Message Addition Rate to Queue | Number of Group Manager threads  
MAX_GROUP_MGRS  
Variable size:  
- small: 1  
- medium: 5  
- large: 10 |
| Parameter Controlling Message Removal Rate from Queue | Number of WO Provision threads  
MAX_WO_HANDLELY  
Variable size:  
- small: 5  
- medium: 10  
- large: 25 |
Table 4–13 lists and describes the ASDL Provision Message Queues.

**Table 4–13  ASDL Provision Message Queues**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool</td>
<td><strong>asap_utils, Real-time System Monitoring</strong> option (109), SARM Server</td>
</tr>
</tbody>
</table>
| Parameter Controlling Message Addition Rate to Queue | Number of WO Provision Threads **MAX_WO_HANDLERS**  
Variable size:  
■ small: 5  
■ medium: 10  
■ large: 25 |
| Parameter Controlling Message Removal Rate from Queue | Number of ASDL Provision Threads **MAX_PROVISION_HANDLERS**–(Less) **MAX_WO_HANDLERS**  
Variable size:  
■ small: 10  
■ medium: 20  
■ large: 50 |
| Example | If you have five (5) WO Handlers and you want ten (10) ASDL Provision Threads, set the **MAX_PROVISION_HANDLERS** to fifteen (15). The difference is the ten (10) that you wanted. |

Table 4–14 lists and describes the NEP driver message queues.

**Table 4–14  NEP Driver Message Queues**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool</td>
<td><strong>asap_utils Real-time System Monitoring</strong> option (109), SARM Server</td>
</tr>
</tbody>
</table>
| Parameter Controlling Message Addition Rate to Queue | Number of ASDL Provision Threads **MAX_PROVISION_HANDLERS** (less) **MAX_WO_HANDLERS**  
Variable size:  
■ small: 10  
■ medium: 20  
■ large: 50 |
| Parameter Controlling Message Removal Rate from Queue | Number of NEPs in the system (dependent on throughput requirements and machine resources). |

There is one NEP Driver Queue for each NEP in the system.

Table 4–15 lists and describes the SRP driver message queues.

**Table 4–15  SRP Driver Message Queues**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool</td>
<td><strong>asap_utils, Real-time System Monitoring</strong> option (109), SARM Server</td>
</tr>
</tbody>
</table>
Manually Tuning ASAP Performance

There is one SRP Drive Queue for each SRP in the system.

**Tips for Tuning the SARM**

To tune the SARM, use the following:

- The number of WO threads (`MAX_WO_HANDLERS`) in the SARM must be equal to the sum of all SARM Driver Threads (`MAX_SARM_DRIVER`) in all of the SRPs.
- The total number of configured handler threads (`MAX_WO_MGRS`) in all SRPs must be equal to the driver threads (`MAX_SRP_DRIVERS`) in the SARM.
- All event notifications not used by any customized SRP implementation must be turned off.
- Sanity level diagnostics during production should be used.
- The configured number of provision handlers (ASDL and WO) must be no less than the sum of the number of handler threads (`MAX_WO_HANDLERS`) and the number of operating NEP servers. Start tuning with a ratio of 1:3 of WO Manager Threads to provision handlers (ASDL and WO).

**Tuning NEP Message Queues**

The purpose of tuning a NEP is to provide:

- ASDLs to the NEs
- Remote Procedure Call (RPC) responses back to the SARM at a rate that does not cause excessive buildup in any of the SARM Notify, Session Manager, or NE Command queues.

*Figure 4–3* illustrates the schematic flow of the NEP.

---

**Table 4–15 (Cont.) SRP Driver Message Queues**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Controlling Message Addition Rate to Queue</td>
<td>Nearly every thread in the SARM can add messages to this queue. Therefore, it is not possible to control the number of messages that are added.</td>
</tr>
</tbody>
</table>
| Parameter Controlling Message Removal Rate from Queue | Number of SRP Driver Threads. **MAX_SRP_DRIVERS** Variable size:  
  - small: 5  
  - medium: 10  
  - large: 25 |
Figure 4–3  NEP Message Queues

Table 4–16 lists and describes the SARM notify message queue.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool</td>
<td>asap_utils Real-time System Monitoring option (109), NEP Server</td>
</tr>
<tr>
<td>Parameter Controlling</td>
<td>Session Manager Thread for the NE. Non-configurable</td>
</tr>
<tr>
<td>Message Addition Rate to Queue</td>
<td>Session Manager Thread for the NE. Non-configurable</td>
</tr>
</tbody>
</table>
Manually Tuning ASAP Performance

You can manage this queue indirectly by decreasing the number of NEs supported by the NEP (usually by increasing the absolute number of NEPs, if machine resources permit) or by balancing the load between NEPs (by moving a busy NE from a busy NEP to a less busy NEP).

Table 4–17 lists and describes the NE command queue.

**Table 4–17 NE Command Message Queues**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool</td>
<td><strong>asap_utils, Real-time System Monitoring option (109), NEP Server</strong></td>
</tr>
<tr>
<td>Parameter Controlling Message Addition Rate to Queue</td>
<td>Session Manager Thread for the NE. Non-configurable</td>
</tr>
<tr>
<td>Parameter Controlling Message Removal Rate from Queue</td>
<td>Command Processor Thread for the NE. Non-configurable</td>
</tr>
</tbody>
</table>

You can manage this queue indirectly by increasing or decreasing NE response times (faster response times decrease the length of the queue).

Table 4–18 lists and describes the session manager queue for the NE.

**Table 4–18 Session Manager Message Queues for the NE**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool</td>
<td><strong>asap_utils, Real-time System Monitoring option (109), NEP Server</strong></td>
</tr>
<tr>
<td>Parameter Controlling Message Addition Rate to Queue</td>
<td>Command Processor Thread of the NE - NEP Driver thread in the SARM for the NEP. Non-configurable</td>
</tr>
<tr>
<td>Parameter Controlling Message Removal Rate from Queue</td>
<td>Session Manager Thread for the NE. Non-configurable</td>
</tr>
</tbody>
</table>

You can manage this queue indirectly using the NE response times (fast responses increase the length of the queue). However, depending on communication and switch technology, this may not be configurable.

**Tips for Tuning NEP**

To tune the NEP, use the following:

- Unless machine resources are limited, target for a ratio of 50 NEs per NEP.
- If possible, group similar NE technologies and switch software loads on a single NEP. This reduces the number of ASDL programs held in memory because each NEP caches its own copy of every ASDL command that it has been asked to perform.
Manually Tuning ASAP Performance

- Balance the load on each NEP by distributing your busy NEs across several NEPs. You must determine the expected load based on both the number and complexity of the ASDL commands being serviced.

Other Performance Issues

You must take into consideration other factors that can affect performance. This section covers the diagnostic levels and query optimization that you need in the tuning process.

The topics in this section include:

- Local versus NFS-Mounted File Systems for Diagnostic Files
- Server Diagnostic Levels
- Diagnostic Messages Output
- Query Optimization
- Table Partitioning

Local Versus NFS-mounted File Systems for Diagnostic Files

ASAP diagnostic files on NFS-mounted file systems increase network traffic and slow down disk I/O. For production systems, the log directories should be local, not NFS mounted.

Server Diagnostic Levels

Low-level server diagnostic levels increase disk I/O. Oracle recommends that you set the diagnostic level for the production system at either PROGRAM_LEVEL or SANITY_LEVEL.

In the Control server database table tbl_appl_proc, set the diagnostics level (column diag_level) to SANE.

Diagnostic levels set in the tbl_appl_proc are persistent through reboots. For more information on tbl_appl_proc, see ASAP Developer’s Guide.

To set a diagnostic level temporarily, use asap_utils parameter 107. See the ASAP Server Configuration Guide for more information.

Diagnostic levels are defined as follows in the common.h file. Both diagnostics and C/C++ code values are provided.

<table>
<thead>
<tr>
<th>Diagnostic Level</th>
<th>C/C++ Code Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KERNEL_LEVEL</td>
<td>KERN</td>
<td>Used by the kernel to generate diagnostic messages. It is only to be used by the core libraries for very low-level debugging of core code.</td>
</tr>
<tr>
<td>LOW_LEVEL</td>
<td>LOW</td>
<td>Used by the application to generate low-level diagnostic messages from any of its functions. Such messages should enable the programmer to debug an application. Once debugged, the diagnostic level of the application should be elevated above LOW_LEVEL.</td>
</tr>
<tr>
<td>FUNCTION_LEVEL</td>
<td>FUNC</td>
<td>Used by the application at the beginning and end of each function to track the operation of the application. This is generally not used in the core application.</td>
</tr>
<tr>
<td>RPC_LEVEL</td>
<td>RPC</td>
<td>Used by the application to produce remote procedure call (RPC) diagnostic messages.</td>
</tr>
</tbody>
</table>
The most commonly used diagnostic levels are **LOW_LEVEL**, **SANITY_LEVEL**, and **PROGRAM_LEVEL**.

### Diagnostic Messages Output

Output of diagnostic messages can be written to disk line-by-line or buffered by the UNIX I/O subsystem. Buffered output results in diagnostics being written to disk in pages, which results in optimal performance.

> **Note:** Oracle highly recommends that you do not use diagnostic line flushing for production systems. Flushing diagnostic messages to disk in lines results in high disk I/O frequency.

To disable diagnostic line flushing, set the configuration parameter `DIAG_LINE_FLUSH` (in the common application programming interface (API) section of the ASAP.cfg file) to “0.” The default value is 1.

### Query Optimization

Oracle RDBMSs use a cost-based query optimizer to determine query access paths. The optimizer is primarily influenced by table and index statistics (number or rows, data distribution, etc.) available to it in the system catalogue tables. These statistics can be updated manually (usually when the amount and distribution of data in a table changes significantly) using utilities provided by the RDBMS. Keeping these statistics current is extremely important since the optimizer will make default assumptions in the absence of these statistics. The quality of the database optimizer decisions depends on the accuracy of these statistics, and hence, directly affects the performance of the ASAP application.

Updating the statistics on large tables (over 1 million rows) can take a long time. This can affect your online response time or induce rollover to small error messages.

> **Note:** When the Oracle statistics are collected on the SARM schema, you should also collect the histograms on the `TBL_WKR_ORD.WO_STAT` column.
This chapter describes Oracle Communications ASAP backup and restore strategies for the file system and the database.

About Backing Up and Restoring Files and Data

As an ASAP system administrator, you should observer the following guidelines for backing up and restoring your ASAP system and databases:

- ASAP System Backup and Restore
- Database Backup and Recovery

ASAP System Backup and Restore

Backup the ASAP environment, Oracle WebLogic Server, and Oracle database schemas in the following situations:

- After you successfully install ASAP
- Before you upgrade ASAP in case you need to rollback your upgrade
- After you successfully upgrade ASAP

See the ASAP Installation Guide for procedures to backup your ASAP system.

In addition to these ASAP system backup scenarios, you should also follow the database backup schedule and procedures described in “Database Backup and Recovery”.

See the section about rolling back your ASAP system in the ASAP Installation Guide to restore your ASAP system. These roll back procedures described in the ASAP Installation Guide will return your ASAP system to the initial installation state and post or pre upgrade state. To recover lost data, restore your database backups described in "Database Backup and Recovery".

Database Backup and Recovery

ASAP application databases must be protected in the event of corruption or failure.

The Service Activation Request Manager (SARM) and Service Request Processor (SRP) databases should be backed up daily. The Control and Network Element Processor (NEP) databases can be backed up less often since they contain only static and monitoring data. The schedule of transaction dumps is dictated by the amount of space available for the transaction logs.
The Administrative database does not need to be backed up because it only contains performance statistics.

**Note:** Backup in a distributed environment is no different than in a non-distributed environment, except that tapes have to be placed in numerous machines.

### Database Backup Strategy

Backup procedures must be performed periodically and should be fully automated by the System Administrator and Database Administrator. Cron scripts can be written which will dump the ASAP application databases, and associated transaction logs to disk and then to tape. In the case of Oracle, these cron scripts should backup to tape all data files, the control file, and archive logs to tape. For Oracle databases the Database Administrator should first decide whether to employ hot backups of the tablespaces or a full cold backup. It is recommended that a cold backup be done at least once a week in addition to a full database export. The backup strategy you choose will be determined primarily by the amount of data loss that can be tolerated in the event of failure.

The ASAP databases are as follows:

- **Control** – Can be recreated from the data on the switches, but may be backed up if required.
- **SRP** – Should be backed up daily.
- **SARM** – Should be backed up daily.
- **NEP** – Can be recreated from the data on the switches, but may be backed up if required.
- **Admin** – Not necessary to back up because it contains only performance statistics.

### Database Backup in a Distributed Environment

Backup in a distributed environment should be no different than a non-distributed environment, except that tapes may have to be placed in numerous machines.

### ASAP WebLogic Server Domain Back Up

You can archive domain configurations:

- Automatically to other media (such as tapes or disks)
- Manually by copying files to another directory, and preferably, other machines

You should regularly back up the following:

- Configuration data
- Security data
- Deployed applications

#### Configuration Data

Back up configuration data, including the `config.xml` file and the `config.xml.booted` file from the `Domain_Home/config/` directory. The `config.xml.booted` file is created when you successfully start the administration server and can serve as backup in the event that the `config.xml` file becomes unusable. You can overwrite the `config.xml` file
with the `config.xml.booted` file to restore the domain to the last successful startup configuration.

**Security Data**

Back up the Administration Server security data for a domain. The security data that you should archive includes:

- Security Configuration Data
- WebLogic LDAP Repository
- `SerializedSystemIni.dat` and Security Certificates

For more information on administering domains, refer to

http://docs.oracle.com/middleware/1213/wls/wls-administer.htm
Managing the Database and File System

This chapter provides information about managing your Oracle Communications ASAP database and file system.

Overview of Managing the Database and File System

The Oracle database and file system management tasks include:

- Configuring the operating system kernel and Oracle database initialization parameters for optimal system performance.
- Monitoring database segment and file system size by defining maximum thresholds for both.
- Using Oracle database management tools to maintain and tune your database.
- Enabling automated ASAP database administration features.
- Using scripts to purge the Oracle database tables, clear system events, alarms, and process information, and, in a test environment, to periodically purge all data.

Configuring Kernel and Database Initialization Parameters

This sections outline some important issues for implementing ASAP. These issues are listed in point form for easy scanning. For a complete reference on the subjects, refer to the appropriate Oracle documentation.

- Before installing the Oracle instance ensure that the default values for kernel and resource settings for the server are set to provide enough shared memory and semaphores. Specifically, check the `project.max-shm-memory` resource. See the *Oracle Installation Guide* for more details.

- You can change the Oracle Server configuration parameters from their default values to suit the requirements of ASAP. To view the Oracle Server parameters, select from the `V$PARAMETER` table or look in the `INITsid.ora` file for the server. Most of the changes to the parameters must be accompanied by shutting down the server and restarting it, in order for the change to take effect.

- Oracle works best with ASAP when given sufficient resources. There are four parameters that are extremely important to configure correctly: `DB_BLOCK_SIZE`, `DB_BLOCK_BUFFERS`, `SHARED_POOL_SIZE`, and `SESSIONS`.

1. **DB_BLOCK_SIZE**

   Set the `DB_BLOCK_SIZE` – 8K is the default. ASAP benefits from a larger block size. This must be done during database creation and cannot be changed.
2. **DB_BLOCK_BUFFERS**

*DB_BLOCK_BUFFERS* is the area of the SGA that is used for the storage and processing of data in memory. As users request information, the information is put into memory. If the *DB_BLOCK_BUFFERS* parameter is set too low, the least recently used data will be flushed from memory. Set this value between 3000 to 5000 depending on the memory limitations of your machine. A value of 5000 with a block size of 8K uses approximately 40MB of memory.

3. **SHARED_POOL_SIZE**

*SHARED_POOL_SIZE* is used to process the procedures, packages, and triggers, as well as the library and data dictionary cache. If the *SHARED_POOL_SIZE* is set too low, you will not get the full advantage of your *DB_BLOCK_BUFFERS*. Set this value between 90MB to 150MB. For most installations this should be sufficient. Again this value should be adjusted depending on the memory limitations of your machine. Setting the *SHARED_POOL_SIZE* too high can cause your system to swap excessively.

4. **SESSIONS**

*SESSIONS* determines the number of user connections that ASAP can establish. Configure 150-200 sessions for each ASAP environment, more if you are running multiple Service Request Processor (SRP) and Network Element Processors (NEPs). Be sure to allocate enough semaphores at the UNIX level to accommodate the amount of sessions.

In all the cases above, the values given as examples only. The appropriate value for your installation will depend on the machine characteristics and limitations.

- Oracle, like all databases, benefits significantly by spreading the I/O among it’s data files across multiple controllers and drives. You must place your *SYSTEM*, *TEMP*, and *ROLLBACK* tablespaces on separate drives and controllers from your *DATA* and *INDEX*. *SARM_DATA* and *SARM_INDEX* will benefit the most from I/O optimization since a lot of activity occurs on these files. In addition, employing optimal striping and data placement of hot files will greatly enhance your database performance. Since minimizing I/O is the primary goal of most database tuning, this point cannot be emphasized enough.

---

**About Monitoring Database Segment and File System Size**

The Control server has a background thread that monitors database table and log segment sizes. This thread reads a static configuration database table to determine database size thresholds. Should a particular threshold be exceeded, the thread issues the system event specified in the static table *tbl_db_threshold* in the Control database. This feature provides a warning to the operations center should the threshold be exceeded.

In addition, the Control server has a background thread that monitors operating system file sizes. This thread reads a static configuration database table to determine file system size thresholds. Should a particular threshold be exceeded, the thread issues the system event specified in the static table *tbl_fs_threshold* in the Control database. This feature allows ASAP to warn the operations center whenever the free disk space drops below the minimum threshold set by you.

The monitoring thresholds function is used to specify thresholds for ASAP databases and transactions affecting it. When these thresholds are defined, an ASAP Control server can monitor the specified database data and transaction log sizes at intervals specified in the ASAP environment variable *DB_MONITOR_TIME*. If any of the
thresholds are exceeded, the Control server issues the system event defined by the user. The event can trigger an alarm or be logged.

See "Configuring System Events and Alarms Using Stored Procedures" for more information on configuring alarms.

An ASAP environment can be partitioned on various machines, and each machine can be monitored by its respective Control server. Typically, one ASAP installation has multiple databases and the ASAP environment can be distributed on one machine or multiple machines. The database and transaction log threshold need to be specified for each database to be monitored.

You can use the DB_MONITOR_TIME and FS_MONITOR_TIME parameters to configure the Control server database and file system monitoring capabilities. For more information about these parameters, see the chapter about the ASAP parameter configuration file (ASAP.cfg) in the ASAP Server Configuration Guide.

---

**Configuring File System Thresholds**

To configure file system thresholds:

1. Open a UNIX terminal to your ASAP environment.

2. Using a text editor, set the FS_MONITOR_TIME parameter in ASAP_home/config/ASAP.cfg to a time interval in seconds. For example 120 seconds.

3. Using sqlplus, connect to the Control server.

   ```
   sqlplus CTRL$/password
   ```

   Where `password` is the password for your Control server.

4. Populate the Control server tbl_fs_threshold database table. The syntax is:

   ```
   INSERT INTO "TEST"."TBL_FS_THRESHOLD" (ASAP_SYS, FILE_SYSTEM, FULL_THRESHOLD, FULL_EVENT) VALUES ('envID', 'fs_path', 'fs_threshold', 'event_type');
   COMMIT;
   ```

   Where the values for the tbl_fs_threshold columns are:
   - `envid`: The ASAP environment ID. The Control server only reads records from the table with values of asap_sys equal to the currently defined ASAP_SYS environment variable.
   - `fs_path`: The name of the file system.
   - `fs_threshold`: The threshold above which the system event is issued (in percentage 0-100).
   - `event_type`: Specifies the system event you want to trigger. Enter an event listed in the the Control server tbl_event_type table (see the ASAP Developer Guide).

5. Populate the Control server tbl_component database table. The syntax is:

   ```
   INSERT INTO "TEST"."TBL_COMPONENT" (TERRITORY, SYSTEM, COMPONENT) VALUES ('envID', 'territory', 'path');
   COMMIT;
   ```

   Where the values for the tbl_component columns are:
   - `envid`: The ASAP environment ID.
   - `territory`: Usually the same name as the ASAP environment ID.
   - `fs_path`: The path to the file system you want to monitor.
6. Restart ASAP for the changes to take effect.

---

**Note:** When a file system size goes beyond the threshold specified in `tbl_fs_threshold`, ASAP triggers an event and logs a diagnostic message in the Control server diagnostic file.

---

**Configuring Database Thresholds**

To configure database thresholds:

1. Open a UNIX terminal to your ASAP environment.
2. Using a text editor, set the `DB_MONITOR_TIME` parameter in `ASAP_home/config/ASAP.cfg` to a time interval in seconds. For example 120 seconds.
3. Using sqlplus, connect to the Control server.
   
   sqlplus CTRL$/password

   Where `password` is the password for your Control server.
4. Populate the Control server `tbl_db_threshold` database table. The syntax is:
   
   INSERT INTO "TEST"."TBL_DB_THRESHOLD" (ASAP_SYS, DB_NAME, DATA_THRESHOLD, DATA_EVENT) VALUES ('envID', 'DB_server', 'data_threshold', 'data_event_type')
   
   COMMIT;

   Where the values for the `tbl_db_threshold` columns are:
   
   - `envid`: The ASAP environment ID. The Control server only reads records from the table with values of asap_sys equal to the currently defined ASAP_SYS environment variable.
   - `DB_server`: The name of the database server tablespace.
   - `data_threshold`: The data threshold above which the system event is issued (in percentage 0-100).
   - `data_event_type`: Specifies the system event you want to trigger as a result of crossing the data threshold. Enter an event listed in the Control server `tbl_event_type` table (see the ASAP Developer Guide).
5. Populate the Control server `tbl_component` database table. The syntax is:
   
   INSERT INTO "TEST"."TBL_COMPONENT" (TERRITORY, SYSTEM, COMPONENT) VALUES ('envID', 'territory', 'tablespace');
   
   COMMIT;

   Where the values for the `tbl_component` columns are:
   
   - `envid`: The ASAP environment ID.
   - `territory`: Usually the same name as the ASAP environment ID.
   - `tablespace`: The tablespace want to monitor.
6. Restart ASAP for the changes to take effect.

---

**Note:** When a tablespace size goes beyond the threshold specified in `tbl_db_threshold`, ASAP triggers an event and logs a diagnostic message in the Control server diagnostic file.
Database Management and Tuning Recommendations

- Oracle can run in two modes ARCHIVELOG or NOARCHIVELOG mode. It is very important to run your production database in the ARCHIVELOG mode. This ensures that you are able to recover up to the minute in case of failure. Oracle uses logs to record transaction information in order to recover from instance and data file failures. Be sure to multi-plex these redo log data files and place them on separate disks and controllers.

- Oracle offers both online and offline database backup choices. Which one you choose will depend on how much downtime your ASAP installation can tolerate. Online backups involve backing up at the tablespace level and require more planning and consideration. Offline backup involves shutting down the database and backing up the data files, control file, and archive log files to tape. While this is occurring, ASAP is down.

- Monitor your tablespaces for index and block fragmentation on a regular basis. Index fragmentation occurs due to the deletion and updating of table rows. Rebuilding the index using the ALTER INDEX name REBUILD (where name is the name of the index) can resolve this problem. Block fragmentation can occur due to PCTFREE and PCTUSED parameters set inappropriately. This results in migrated and chained rows which affect database I/O performance.

- Update your table and index statistics on a regular basis for those tables whose data distributions change dynamically. As mentioned in a previous section, the query optimizer is primarily influenced by these statistics. If a query or transaction that was running fine appears to suddenly slowdown or hang, chances are the table and index statistics need to be updated.

- Tune any custom SQL statements you introduce to your ASAP installation, since they involve both your application and database. To analyze your SQL statements you will need to:
  1. Add the line to your init$SID.ora file: timed_statistics=true
  2. Restart your database.
  3. Start the sql tracing for your session using the ALTER SESSION SET SQL_TRACE=TRUE.
  4. Run your application
  5. Run tkprof against the tracefile created by your application: tkprof tracefile outputfile EXPLAIN=username/passwd
  6. Look at the formatted output of the trace command and make sure that your SQL statements are using indexes correctly. You must run explains on the SQL statements to determine the access path they are using.

Refer to the Oracle Tuning Guide for more detailed information on tuning SQL statements.

- Use the UTLBSTAT and UTLESTAT scripts provided by Oracle to monitor your Oracle instance, and diagnose performance problems. These scripts are located in the Oracle_Home/rdbms/admin directory in a UNIX environment. These scripts create a report file report.txt which shows differences in the statistics from the time UTLBSTAT was submitted, until the time that UTLESTAT was submitted. To obtain time-based statistics from UTLBSTAT and UTLESTAT, the TIMED_STATISTICS parameter must be set to TRUE.
Statspack

Statspack is an alternative to the UTLBSTAT/UTLESTAT tuning scripts. Statspack collects more information, storing performance statistics data permanently in the database for later use in analysis and reporting. The data collected can be analyzed using the report provided, which includes an instance health and load summary page, high resource SQL statements, as well as the traditional wait events and initialization parameters.

For more information, see Oracle 11g Database Performance Tuning Guide and Reference Guide.

Enabling Automated ASAP Database Administration Options

The Control application programming interface (API) provides a background thread that performs database administration tasks within ASAP. At a user-configured time of day, every application server process connects to its primary database and calls a user-defined function that performs the following tasks:

- Gathers customer-defined statistics related to ASAP processing.
- Archives data about to be purged from the database, if required.
- Purges data older than a date and time specified by the user.
- Performs an “orphan” purge of orphaned records to confirm the database’s integrity.

The ASAP configuration variable, DB_ADMIN_ON, enables and disables the database administration procedures. This feature is of value to users who may have many application servers defaulting to the same database and do not want them performing database administration procedures independently on it.

For more information on the Control API, refer to the ASAP Developer’s Guide.

Purging the Database and File System

To keep disk usage at an acceptable level the ASAP administrator should regularly purge the diagnostic and log files generated by ASAP, as well as the work orders (WOs) received from the service order system. Database purging is controlled by database administrator stored procedures that are automatically executed daily.

The diagnostic files are located in the directory $ASAP_home/DATA/logs (where $ASAP_home is the ASAP installation directory). The flat files of service orders are usually located in the $ASAP_home/DATA directory in a dated file.

Purging the Database

Database purging can be performed for the SRP, Service Activation Request Manager (SARM), and control databases, but is most commonly performed in the SARM as this is where the majority of the WO information is stored. The SRP and SARM databases maintain a history of all WOs received, while the ASAP control database maintains a history of alarms, events, performance, and process information.

The purging of the SRP and SARM is based on WO age. The purge age is usually determined by the amount of available disk space. Usually, only orders that have been completed for a certain amount of time are purged.
Database purging is controlled by database administration functions that automatically execute according to the following parameters that you can configure in ASAP.cfg:

- **DB_ADMIN_ON** – Boolean flag. If set, it enables the database administration thread operation in the application server. This can be disabled in particular servers in situations where multiple servers share the same application database (for example, multiple NEPs) and then only one server is required to perform this database administration. Default = 0.

- **DB_ADMIN_TIME** – The number of minutes after midnight when the database administration tasks are to be performed. This is usually performed at a time of low system activity. Default = 300.

- **DB_ADMIN_PROC** – The function the database administration thread calls at a specified time in the day. This function can be configured to perform multiple tasks, including archiving and purging dynamic data. Default = SSP_db_admin.

- **DB_ADMIN_PROC_PARAM** – The integer parameter passed to the database administration function. For example, this can specify a purge interval for a particular database. Default = 100.

- **GATHER_STATS** – Enables the gathering of statistics for tables and indexes. Default = 0.

- **GATHER_STATS_PROC** – Indicates the procedure to use to gather statistics on the SARM database. Default = SSP_gather_asap_stat.

- **DB_PCT_ANALYZE** – Percentage of table to analyze when gathering stats. Default = 20.

- **DB_PCT_ANALYZE_IDX** – Percentage of index to analyze when gathering stats. Default = 40.

- **GATHER_DEGREE** – Degree of parallelism to use when gathering stats. Default = 1.

**Note:** You can configure every ASAP server to run similar administration functions, but the SARM is the only server where a default function (SSP_db_admin) has been provided. To perform administration on databases other than the SARM database, you must copy the above configuration parameters to the appropriate section of the ASAP.cfg configuration file, and load the SSP_db_admin to those servers.

Table 6–1 provides recommended data purge frequency and methods.

<table>
<thead>
<tr>
<th>Data Object Being Purged</th>
<th>Selection Criteria</th>
<th>Frequency of Purge</th>
<th>Method of Purge</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASAP log files</td>
<td>All log files older than X days</td>
<td>Daily</td>
<td>Cron script. A sample cron script is located on &quot;Sample Cron Script for Clearing Alarms, Events, and Process Information&quot;.</td>
</tr>
</tbody>
</table>
## Sample Database Purge Script

The following sample script can be used to delete successfully provisioned WOs from `tbl_work_ord` that are older than a specified number of days. In addition, this script calls the `SSP_orphan_purge` function, which deletes all orphaned records. When you delete WOs, information related to these WO may remain in other tables. The `SSP_orphan_purge` function purges information that is related to WOs that have been removed from the database. The affected tables include:

- `tbl_asap_stats`
- `tbl_info_parm`
- `tbl_srq`
- `tbl_srq_csd1`
- `tbl_srq_log`
- `tbl_asdl_log`
- `tbl_srq_parm`
- `tbl_srq_asdl_parm`
- `tbl_wo_event_queue`
- `tbl_wo_audit`
- `tbl_usr_wo_prop`
- `tbl_aux_wo_prop`

### Note:
The `SSP_orphan_purge` function is time-consuming and requires considerable system resources. Therefore, it should not run during peak hours.

```sql
create proc SSP_db_admin @days int
as
begin
    declare @cutoff_dts datetime
```

### Table 6–1 (Cont.) Database Purge Frequency and Methods

<table>
<thead>
<tr>
<th>Data Object Being Purged</th>
<th>Selection Criteria</th>
<th>Frequency of Purge</th>
<th>Method of Purge</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASAP control database</td>
<td>All dynamic tables older than X days</td>
<td>Daily</td>
<td>Function</td>
</tr>
<tr>
<td>SRP database</td>
<td>All WOs older than X days</td>
<td>Daily</td>
<td>Function</td>
</tr>
<tr>
<td>SARM database</td>
<td>All WOs older than X days</td>
<td>Daily</td>
<td>Function</td>
</tr>
</tbody>
</table>

Sample functions are described in "Sample Database Purge Script".
if (@days is not null and @days > 0)
begin
  select @cutoff_dts = dateadd(day, -@days, getdate())
  delete tbl_wrk_ord
  where comp_dts < @cutoff_dts and wo_stat = 104
end
exec SSP_orphan_purge
end

You can customize this script in a variety of ways (for example, you can delete WOs that have successfully completed (104) or orders that have completed and failed due to timeout). When ASAP triggers the DB_admin procedure, ASAP also performs other optimization, such as recompiling stored procedures for optimal database access, and so forth.

In the following example, after the function has deleted each set of 1000 orders, the function performs a commit and the rollback segment is flushed. This prevents the Oracle rollback segment from being exceeded. This example also employs the SSP_orphan_purge function described for the previous example.

```sql
CREATE OR REPLACE FUNCTION SSP_db_admin(
  days INTEGER )
RETURN INTEGER
AS
StoO_selcnt INTEGER;
StoO_error INTEGER;
StoO_rowcnt INTEGER;
StoO_errmsg VARCHAR2(255);
cutoff_dts DATE;
retval integer;
BEGIN
IF (SSP_db_admin.days IS NOT NULL AND SSP_db_admin.days > 0) THEN
BEGIN
  SSP_db_admin.cutoff_dts := SYSDATE-SSP_db_admin.days;
BEGIN
StoO_error   := 0;
StoO_rowcnt  := 0;
-- Created a loop to split the deletion of orders in portions of 1000
-- to remove the risk of reaching the rollback segment limit
-- NOTE! The orphans are deleted by a trigger defined on the_tbl_wrk_ord table and not by the SSP_orphan_purge function call
-- below
LOOP
  DELETE  FROM tbl_wrk_ord
  WHERE comp_dts < SSP_db_admin.cutoff_dts
  AND wo_stat = 104
  AND rownum <= 1000;
  EXIT WHEN SQL%ROWCOUNT = 0;
  COMMIT;
END LOOP;
StoO_rowcnt := SQL%ROWCOUNT;
COMMIT WORK;
-- EXCEPTION
-- WHEN OTHERS THEN
-- StoO_error := SQLCODE;
END;
END IF;
BEGIN
Purging the Database and File System

Sample Cron Script for Clearing Alarms, Events, and Process Information
The following is a sample cron script that clears alarm entries, event logs and process information.

```
# ASAP database and log housekeeping script
#
# Call this script with one parameter, specifying the number of days.
# Alarm entries, event logs and process info logs are cleared. Log files and directories are also cleared.
#
# The script will exit if the number of days is less than 5
#
$.HOME/.profile > /dev/null

PROG_NAME=`basename "$0"`

if [ "$1" == "" ]
then
  echo "Usage $PROG_NAME <admin_days>"
  exit
fi

let ADMIN_DAYS=$1+0
if [ "$?" != "0" ]
then
  echo "Number of days must be numeric"
  exit
fi

if [ $ADMIN_DAYS -lt 5 ]
then
  echo "Cannot run with less than 5 days lead-time"
  exit
fi

# Clear CTRL entries

echo "Deleting alarm logs, event logs and process info from CTRL more than $ADMIN_DAYS days old"

sqlplus -s $CTRL_USER/$CTRL_USER << HERE

var admin_days number;
```
exec :admin_days := $ADMIN_DAYS;

delete from TBL_ALARM_LOG where start_dts < sysdate - :admin_days;
commit;
delete from TBL_EVENT_LOG where event_dts < sysdate - :admin_days;
commit;
delete from TBL_PROCESS_INFO where info_dts < sysdate - :admin_days;
commit;

HERE

############################################################################
# Truncate ASAP.Console and ControlProgramOutput so that old entries can be
# cleared with the diagnostic files
############################################################################

cd $LOGDIR
FILE1=ASAP.Console
FILE2=ControlProgramOutput
CUR_DTS=`date +'%Y_%b_%d-%T'`

if [ -f $FILE1 ]; then
    NEWFILE=$FILE1.diag.$CUR_DTS
    echo Copying current $FILE1 to $NEWFILE
    cp $FILE1 $NEWFILE
    cp /dev/null $FILE1
else
    echo $LOGDIR/$FILE1 not found
fi

if [ -f $FILE2 ]; then
    NEWFILE=$FILE2.diag.$CUR_DTS
    echo Copying current $FILE2 to $NEWFILE
    cp $FILE2 $NEWFILE
    cp /dev/null $FILE2
else
    echo $LOGDIR/$FILE2 not found
fi

############################################################################
# Clear log files
############################################################################

echo "Clearing old ASAP log files..."

find . -type f -name "*diag*" -atime +$ADMIN_DAYS -exec echo Removing file {} \;
-exec rm -f {} \;

############################################################################
# Clear log directories not accessed within admin days
############################################################################

echo "Clearing old ASAP log directories..."

find . -type d -name '2*' -atime +$ADMIN_DAYS -exec echo Removing directory {} \;
-exec rm -rf {} \;

cd -

echo "$PROG_NAME finished"
Purging Test Systems

The cleandata script is designed to clean data from the test system during system testing. It is intended for use in development environments.

---

**Note:** You must stop the Oracle WebLogic Server before running the cleandata script. Otherwise, a JDBCStoreException is thrown, and a WebLogic Server error is logged.

---

**Usage**

cleandata [-d]

The `[-d]` option instructs the script to retrieve password information from the credential store factory (CSF) wallet, located in `ASAP_Home/install/cwallet.sso` file, where `ASAP_Home` is the directory in which ASAP is installed. The `cwallet.sso` file is typically used only in development environments.

This script does the following:

- Deletes events, process information, and alarms from the Control (CTRL) database (specifically, truncates `tbl_event_log`, `tbl_process_info`, `tbl_alarm_log`, `tbl_unid`)
- Deletes WOs from the ASAP database and executes an orphan purge (specifically, truncates `tbl_wrk_ord`, `tbl_ne_monitor`, `tbl_ne_event`, `tbl_unid`, `tbl_label_value`, `tbl_srt_correlation`, `tbl_srt_ctx`, `temp_wrk_ord`)
  
  **Note:** The `tbl_ne_monitor` table has been deprecated from ASAP 4.6.x onwards.

- Deletes WOs from the SRP database (`tbl_wrk_ord`)

- Deletes performance data from Admin database (truncates `tbl_perf_order`, `tbl_perf_csd1`, `tbl_perf_asdl`, `tbl_perf_ne`, `tbl_perf_ne_asdl`, `tbl_aims_rpc`, `tbl_aims_rpc_param`, `tbl_aims_rpc_dest`, `tbl_aims_audit_log`)

- Deletes diagnostic and log files

---

**Note:** Restart the WebLogic Server for the domain after running the cleandata script.
This chapter provides troubleshooting information for Oracle Communications ASAP.

Overview of Troubleshooting ASAP
As an ASAP system administrator, you can perform the following tasks:

Troubleshooting Checklist
When any problems occur, it is best to do some troubleshooting before you contact Oracle Global Support:

- You know your installation better than Oracle Global Support does. You know if anything in the system has been changed, so you are more likely to know where to look first.
- Troubleshooting skills are important. Relying on Global Support to research and solve all of your problems prevents you from being in full control of your system.

If you have a problem with your Product system, ask yourself these questions first, because Oracle Global Support will ask them of you:

- What exactly is the problem? Can you isolate it? For example, if it is a problem with an application, does it occur on one instance of the application, or all instances?
- Oracle Global Support needs a clear and concise description of the problem, including when it began to occur.
- What do the log files say?
- This is the first thing that Oracle Global Support asks for. Check the error log for the Product component you’re having problems with.
- Have you read the documentation?
- Look through the list of common problems and their solutions in Diagnosing some common problems with Product.
- Has anything changed in the system? Did you install any new hardware or new software? Did the network change in any way?
- Have you read the Release Notes?
- The Release Notes include information about known problems and runarounds.
- Does the problem resemble another one you had previously?
- Has your system usage recently jumped significantly?
Is the system otherwise operating normally?
Has response time or the level of system resources changed?
Are users complaining about additional or different problems?
Can you run clients successfully?
Are any other processes on the system hardware functioning normally?

If you still can’t resolve the problem, contact Oracle Global Support as described in "Getting Help with ASAP Problems".

Using Error Logs to Troubleshoot ASAP

ASAP error log files provide detailed information about system problems. If you’re having a problem with ASAP, look in the log files.

Log files include errors that need to be managed, as well as errors that do not need immediate attention (for example, invalid login attempts). To manage log files, you should make a list of the important errors for your system, as opposed to errors that do not need immediate attention.

Understanding Error-Message Syntax

For more information about error message syntax, see "About Log Files".

Collecting Diagnostic Information

For more information about collecting diagnostic information, see "About Diagnostic Files".

Common ASAP Problems and Solutions

This section describes the following problems, and how to resolve them:

- Problem: ASAP Servers Do Not Start - Database Tablespaces Used Up
- Problem: ASAP Servers Do Not Start - Interfaces File Empty or Missing
- Problem: ASAP Servers Do Not Start - No Information In Interfaces File
- Problem: ASAP Servers Do Not Start - Wrong ASAP User Owner and Permissions
- Problem: ASAP Servers Processes Do Not Start - Database Server Processes Used
- Problem: ASAP Servers Processes Do Not Start - Database Server Sessions Used
- Problem: ASAP Servers Do Not Start - Insufficient Server User Connections Defined
- Problem: ASAP Servers Do Not Start - Insufficient Number of Threads
- Problem: Control Server Crashes - No Free Messages
- Problem: JNEP Server Does Not Start - Wrong Database Connection Information
- Problem: JNEP Server Does Not Start - Invalid Server Port Numbers
- Problem: NEP Server Does Not Start - Problem with JNEP Java Process Start Script
- Problem: WebLogic Server Fails to Detect Passive RAC Database During Failover
Problem: ASAP Servers Do Not Start - Database Tablespaces Used Up

None or some of the ASAP servers can start as the database tablespace used by the server is used up and there is no free space. Error messages like the following are generated to indicate that writing to the database failed:

Error: RPC 'CSP_system_event' Failed

To resolve this issue increase the tablespace sizes. You may also set autoextend for the tablespaces.

Problem: ASAP Servers Do Not Start - Interfaces File Empty or Missing

ASAP servers are configured properly in database tables but information for all servers is missing in the file $SYBASE/interfaces file. If Control server information is missing, then no process will start when ASAP is restarted.

The following messages are generated:

- at the UNIX console
  "Failed to start Master Control Server CTRLH730"

- in UTILITY.diag file:
  'ct_connect(): directory service layer: internal directory control layer error: Requested server name not found.'

- in ASAP.Console file:
  CTRLH730 Server: Fatal Server Error:
  CTRLH730: srv_run() Failed

Make sure that $SYBASE/interfaces file exists and contains correct information for all ASAP servers configured.

Problem: ASAP Servers Do Not Start - No Information In Interfaces File

$SYBASE/interfaces file has only Control server information, but none of the other servers. However all servers are defined in the server configuration tables in Control server database (tables tbl_appl_proc, tbl_component, etc.).

The Control server will start successfully. For example:

Starting Master Control Server CTRLH730...
Successful Startup of Master Control CTRLH730

The following messages are generated:

- at the UNIX console
  Error : ASAP server is already running, or can not start one of ASAP server

- in UTILITY.diag file, messages are logged for each server that had no info in interfaces file For example:
  *Local Startup Error: Can't Startup Server NEP_H730 ...'

- in ASAP.Console file:
  Open Server Error # [16013] Severity [20] State [0] Error Text [Could not find...
Make sure that $SYBASE/interfaces file exists and contains correct information.

Problem: ASAP Servers Do Not Start - Wrong ASAP User Owner and Permissions

Owner and file permissions might have been changed preventing the ASAP servers record data to the file system. For example, file or folder owner changes or file permission changes for DATA and its subfolders may cause this issue.

The following error message is generated:
/oracle/user1/A720_I/scripts/start_control_sys: line 197: /oracle/user1/A720_I/DATA/logs/ASAP.Console: cannot create [Permission denied]

Make sure that all the folders and files under ASAP installation folder are owned by the ASAP user and the user has all the required file permissions. For example, it should have read/write for everything under DATA folder otherwise it cannot create the server diagnostic files.

Problem: ASAP Servers Processes Do Not Start - Database Server Processes Used

The following error message will be seen in the control or other server diagnostic files:
ORA-00020: maximum number of processes (%s) exceeded

Increase the value of the PROCESSES initialization parameter.

Problem: ASAP Servers Processes Do Not Start - Database Server Sessions Used

The following error message will be seen in the control or other server diagnostic files:
Error - ORA-00018: maximum number of sessions exceeded

Increase the value of the SESSIONS initialization parameter.

Problem: ASAP Servers Do Not Start - Insufficient Server User Connections Defined

Some ASAP servers do not start and a message like the following is observed in the Control server diagnostic file:
"Configuration of 70 connections has been exceeded, connection rejected."

Each ASAP server started establishes a defined number of connections to the Control server. There is an open server configuration parameter named MAX_CONNECTIONS which can be used by any open server application to limit the maximum number of connections from client applications. This problem is usually encountered when additional Network Element Processor (NEP) servers are configured as each additional NEP server means more connections to the Control server.

The solution is to increase the value of the parameter MAX_CONNECTIONS for the Control server in the ASAP_home/config/ASAP.cfg file. You can also set a value in the
global section for all ASAP servers. Note that when adjusting this parameter, you may need to adjust additional parameters (see "Improving ASAP Performance").

**Problem: ASAP Servers Do Not Start - Insufficient Number of Threads**

ASAP servers cannot start with the reason being indicated as some remote procedure calls (RPCs) are not defined. For example:

Error: RPC install_exit_handler Not Defined
Error: Unable to Spawn Service Thread TORONTO - Insufficient Resources
SARMH730 Server: Information: Error 16115 Severity 10 State 0 'Could not start thread
Error: Unable to Spawn Service Thread WO Mgr 2 - Insufficient Resources

ASAP Servers are configured to have a maximum number of threads in the `ASAP_home/config/ASAP.cfg` file. If this number is not big enough, when a thread is tried to be started, the server process will generate an error message and terminate.

Adjust the value of the parameter `MAX_THREADS` in the `ASAP.cfg`, globally or for a specific server.

Note that when adjusting this parameter, you may need to adjust additional parameters (see "Improving ASAP Performance").

**Problem: Control Server Crashes - No Free Messages**

Control server crashes with the following messages in the Control server diagnostic file:

Error: [CTRLPRDI] Fatal Thread Error - Terminating

Message resources were used up during high volume WO provisioning because `MAX_MSGPOOL` is not big enough for Control server.

Increase the value of `MAX_MSGPOOL` for the Control server in the `ASAP_home/config/ASAP.cfg` file. This should be done in the CTRL section as otherwise it will affect all the servers.

The minimum value of this configuration value is `MAX_MSGQUEUES*256`. You may need to adjust additional parameters (see "Improving ASAP Performance")

**Problem: JNEP Server Does Not Start - Wrong Database Connection Information**

The database connection information (`DB_CONNECT`) in `ASAP.properties` was wrong and as a result JNEP was unable to connect to database while starting.

While NEP server java process (JNEP) was creating a pool of database connections, it failed with the exception message:

"The Network Adapter could not establish the connection"
java.sql.SQLException: Io exception: The Network Adapter could not establish the connection
at oracle.jdbc.driver.DatabaseError.throwSqlException(DatabaseError.java:125)
at oracle.jdbc.driver.DatabaseError.throwSqlException(DatabaseError.java:162)
at oracle.jdbc.driver.DatabaseError.throwSqlException(DatabaseError.java:274)
at oracle.jdbc.driver.T4CConnection.logon(T4CConnection.java:319)
at oracle.jdbc.driver.PhysicalConnection.(PhysicalConnection.java:344)
Database connection information in ASAP.properties file is wrong. Correct it as below and restart the NEP server.

```
DB_CONNECT=(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = xx.xx.xx.xx)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME = orcl)))
```

HOST, PORT or SERVICE_NAME could be wrong in this property.

**Problem: JNEP Server Does Not Start - Invalid Server Port Numbers**

NEP server does not start with an error message in the Control server diagnostic file. A port number outside the range might have been assigned to the NEP server in $SYBASE/interfaces. If the port number is within the range, it might have been bound by another process. Or the issue could be with the listener port that is assigned to the NEP java process. It could be out of range or already bound by another process.

Correct the port number for the NEP server in $SYBASE/interfaces file to be inside the range. If the port number is bound by another process already, it can not be used by the NEP server. A free, unused port number should be assigned for the NEP server.

Make sure that the port number assigned to the NEP java process (table tbl_listeners in Control server database) is also within the range and not bound by any other process.

**Problem: NEP Server Does Not Start - Problem with JNEP Java Process Start Script**

The NEP server java process is started by a script named $NEP_jinterpret which is under $ASAP_BASE/programs folder. This script may not have proper execute permissions.

Give the proper execution permission for the script.

**Problem: WebLogic Server Fails to Detect Passive RAC Database During Failover**

In a development environment, when testing Oracle RAC failover capabilities, it is possible to experience the following WebLogic error:

```
weblogic.jms.common.JMSException: weblogic.messaging.kernel.KernelException: Error persisting message
```

The starting point for this scenario involves the following configuration:

- A running ASAP server environment and active WebLogic server instance (ASAP Environment).
- A running active Oracle RAC database (RAC1).
- A passive Oracle RAC database that is shutdown (RAC2).

To achieve the error message listed above, you must complete the following steps:

1. Manually startup RAC2.
2. Manually shutdown RAC1 within four minutes of starting RAC2.

The ASAP Weblogic server instance requires up to four minute to detect RAC2. Shutting down RAC1 within this time period will prevent ASAP from failing over
to RAC2 because the ASAP WebLogic server has not had enough time to detect the presence of RAC2.

This error message is possible in a specific scenario that is unlikely to happen in a production environment.

Component Failure Scenarios

This section describes the behavior of ASAP in the event of component failure.

WebLogic Administration and Managed Server Failure and Recovery Scenarios

If the administration server shuts down while the managed server is running, the administration server regains control of the domain upon restart without having to restart the managed server.

If the administration server cannot be restarted following a failure, you must restore the administration server on another machine.

Restarting the WebLogic Managed Server

If the administration server is running, the managed server retrieves configuration data from the administration server. If the administration server is unavailable, the managed server retrieves its configuration and security information from the filesystem.

SRP Failure Scenario

If the Service Request Processor (SRP) fails, the Service Activation Request Manager (SARM) and NEP provision all orders that are saved in the database and are awaiting provisioning. The SARM saves the order notification RPCs in the SARM database. It then sends periodic “heartbeat” messages to the failed SRP to check for availability. When the SRP starts again, the SARM retrieves and processes all order notification RPCs saved on disk before processing new orders.

SARM Failure Scenario

If the SARM fails, the SRP can neither send orders to the SARM nor receive notifications from the SARM, although the SRP is running and receiving orders from the upstream. Therefore, the SRP cannot process orders and remains idle until the SARM is restarted.

NEPs complete the processing of Atomic Service Description Layer (ASDL) requests currently in progress, and then it becomes idle.

The SRP and NEPs send periodic “heartbeat” messages to determine when the failed SARM becomes available.

NEP Failure Scenario

In the event of NEP failure, the SRP continues to translate and send orders to the SARM, which continues to provision ASDLs scheduled to be provisioned by any operational NEP. All ASDLs to be provisioned by NEs managed by the failed NEP are added to SARM Provisioning Pending Queue. The SARM sends periodic “heartbeat” messages to determine when the failed NEP becomes available.
Getting Help with ASAP Problems

Control Server Failure Scenario
If the Control server is shut down, all other ASAP applications controlled by it are also shut down.

Database Failure Scenario
If the SRP database fails, the SRP is shut down if it relies on the SRP database. For example, the C SRP server. Similarly, the SARM and NEP servers shut down whenever their respective databases fail. If the Control database fails, the Control server is shut down, taking down all other ASAP applications controlled by it. In a distributed environment, if the primary Control server goes down, the secondary Control servers detect this and shut down.

If the SQL Server is down, all databases fail and the entire system is shut down.

NE Unavailability Scenario
If an NE becomes unavailable, all ASDLs to the unavailable NE queue up in the SARM. The SARM will not take new work orders (WOs).

If the asdl_timeout parameter is set for an ASDL, and the timeout parameter exceeds asdl_retry_number parameter, the WO to which the ASDL belongs fails and rolls back.

If the request_timeout parameter (an NE timeout parameter) is set for the WO to which an ASDL belongs, and the request_retry_number (an NE timeout parameter) exceeds, the WO fails and rolls back.

If neither ASDL nor NE timeout parameters are set, and the WO timeout parameters exceed, all orders with ASDLs going to the unavailable NE fail due to timeout.

SRP and SARM Failure Scenario
If both the SRP and SARM fail, each NEP completes its processing of ASDL requests currently in progress. The NEP then closes all connections to NEs and then remains idle.

SARM and NEP Failure Scenario
If the SARM and NEP fail, the SRP cannot send orders or receive notifications from the SARM. No order provisioning is possible.

Getting Help with ASAP Problems
If you can’t resolve your ASAP problem, contact Oracle Global Support.

Before You Contact Global Support
Problems can often be fixed by shutting down ASAP and restarting the computer that the ASAP system runs on. To shut down and restart ASAP, see "Starting and Stopping ASAP".

If that doesn’t solve the problem, the first troubleshooting step is to look at the error log for the application or process that reported the problem. See "Using Error Logs to Troubleshoot ASAP". Be sure to observe the checklist for resolving problems with ASAP before reporting the problem to Oracle Global Support. See "Troubleshooting Checklist".
Reporting Problems

If the checklist for resolving problems with ASAP does not help you to resolve the problem, write down the pertinent information:

- A clear and concise description of the problem, including when it began to occur.
- Relevant portions of the relevant log files.
- Relevant configuration files.
- Recent changes in your system, even if you don’t think they are relevant.
- List of all ASAP components and patches installed on your system.

When you are ready, report the problem to Oracle Global Support.
This appendix describes the Oracle Communications ASAP directory structure.

### ASAP Directory Structure and Contents

Table A–1 lists and describes the ASAP directories.

**Table A–1  ASAP Directories**

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ASAP_home/activationModels</code></td>
<td>Contains installed SAR files. When you first install ASAP, the ASAP installer automatically installs the <strong>Nortel_DMS_POTS.sar</strong>. All additional SAR files that you upload using Service Activation Deployment Tool (SADT) or the <code>installCartridge.sh</code> script are added to this folder.</td>
</tr>
<tr>
<td><code>ASAP_home/config</code></td>
<td>This folder contains important configuration files such as <code>ASAP.cfg</code> and <code>OCA.cfg</code>.</td>
</tr>
<tr>
<td><code>ASAP_home/DATA</code></td>
<td>Contains the <code>logs</code> directory.</td>
</tr>
<tr>
<td><code>ASAP_home/DATA/logs</code></td>
<td>Location of all server logs and the ASAP console log files. The ASAP Console file contains standard input, standard output, and errors that are sent to a console screen. The ASAP Console file records any startup errors.</td>
</tr>
<tr>
<td><code>ASAP_home/DATA/logs/yyyymmdd</code></td>
<td>These folders are created by ASAP daily or whenever a server starts up. Diagnostic files are located here and contain <code>appl_name.diag</code> files (where <code>yyyymmdd</code> is the date for which you want to view the diagnostics and <code>appl_name</code> is the client or server application.)</td>
</tr>
<tr>
<td><code>ASAP_home/islq</code></td>
<td>Contains sql scripts for ASAP database upgrades. The sql scripts with the words procs contain stored procedures, functions, and triggers and should be used when upgrading ASAP. The ASAP installer uses the other scripts when you first install ASAP and should not be run manually.</td>
</tr>
<tr>
<td><code>ASAP_home/db_migration</code></td>
<td>Contains database sql and korn shell scripts for upgrading the ASAP database.</td>
</tr>
<tr>
<td><code>ASAP_home/include</code></td>
<td>Contains ASAP C header files.</td>
</tr>
<tr>
<td><code>ASAP_home/include/csolsrp</code></td>
<td>Contains the Service Request Processor (SRP) emulator C header files.</td>
</tr>
<tr>
<td><code>ASAP_home/include/asc_oo_fw</code></td>
<td>Contains object oriented ASAP C++ header files.</td>
</tr>
<tr>
<td><code>ASAP_home/install</code></td>
<td>Contains files related to the ASAP installation.</td>
</tr>
</tbody>
</table>
## Table A–1 (Cont.) ASAP Directories

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASAP_home/JLIB</td>
<td>Contains java archive (jar) files used by the ASAP Java components.</td>
</tr>
<tr>
<td>ASAP_home/ant</td>
<td>Contains the ant tool and related subdirectories used in the Java software build process.</td>
</tr>
<tr>
<td>ASAP_home/JRE</td>
<td>Contains the Java runtime environment and all related sub folders.</td>
</tr>
<tr>
<td>ASAP_home/lib</td>
<td>Contains jar file and Enterprise Archive (.ear) files for WebLogic based ASAP tools and servers.</td>
</tr>
<tr>
<td>ASAP_home/nawk_progs</td>
<td>This folder contains nawk programs.</td>
</tr>
<tr>
<td>ASAP_home/objects</td>
<td>Contains ASAP Server application programming interface (API) libraries.</td>
</tr>
<tr>
<td>ASAP_home/oca_sys_if</td>
<td>This folder contains various Order Control Application (OCA) client and OCA client related sub folders.</td>
</tr>
<tr>
<td>ASAP_home/oca_sys_if/client</td>
<td>This folder contains the OCA client sub folders.</td>
</tr>
<tr>
<td>ASAP_home/oca_sys_if/client/java</td>
<td>This folder contains the OCA client sub folders.</td>
</tr>
<tr>
<td>ASAP_home/oca_sys_if/client/java/config</td>
<td>OCA client configuration files.</td>
</tr>
<tr>
<td>ASAP_home/oca_sys_if/client/java/scripts</td>
<td>OCA client execution scripts.</td>
</tr>
<tr>
<td>ASAP_home/oca_sys_if/client/java/src</td>
<td>OCA client source files and directories.</td>
</tr>
<tr>
<td>ASAP_home/oca_sys_if/client/java/utils</td>
<td>OCA applet splash page.</td>
</tr>
<tr>
<td>ASAP_home/oca_sys_if/idls</td>
<td>This interface description languages (IDLs) folder contains a CORBA ASC interface file used by the OCA client.</td>
</tr>
<tr>
<td>ASAP_home/oca_sys_if/sample</td>
<td>This directory contains a sample external validation server for the OCA SRP.</td>
</tr>
<tr>
<td>ASAP_home/oca_sys_if/sample/validationServer</td>
<td>This sample program illustrates how to create a user defined external validation server in Java. You can configure the OCA SRP server to validate work orders (WOs) with this external validation server. For more information about the sample external validation server, read the README contained in this directory.</td>
</tr>
<tr>
<td>ASAP_home/oracle_communications</td>
<td>Folder reserved for future use.</td>
</tr>
<tr>
<td>ASAP_home/oracle_communications/asap</td>
<td>Folder reserved for future use.</td>
</tr>
<tr>
<td>ASAP_home/patch</td>
<td>This folder contains various patch files installed automatically when you install or upgrade ASAP.</td>
</tr>
<tr>
<td>ASAP_home/profile</td>
<td>This folder contains various profiles that ASAP uses for the different ASAP installation and configuration options.</td>
</tr>
<tr>
<td>ASAP_home/programs</td>
<td>This file contains various programs used to run ASAP utilities, scripts, and servers.</td>
</tr>
<tr>
<td>ASAP_home/samples</td>
<td>This folder contains various sample servers, service models, sar files, and server configurations.</td>
</tr>
<tr>
<td>ASAP_home/samples/ASDL_ROUTE</td>
<td>This sample folder contains the files and instructions for running sample user defined stored procedures to route WOs.</td>
</tr>
<tr>
<td>ASAP_home/samples/CsolSrp</td>
<td>This folder contains a sample C-based SRP.</td>
</tr>
<tr>
<td>ASAP_home/samples/CsolSrp/csslsrp_sample</td>
<td>This folder contains the files and instructions for implementing the most basic functions of an SRP.</td>
</tr>
</tbody>
</table>
Table A–1 (Cont.) ASAP Directories

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASAP_home/samples/DIT</td>
<td>This folder contains a sample cartridge, and related sub folders, configured to run with the Java Service Request Processor (JSRP) sample client and SRT. This sample is provided to test the functionality of ASAP, the JSRP client, and the optional SRT component.</td>
</tr>
<tr>
<td>ASAP_home/samples/EDD</td>
<td>This directory contains two sample programs and related sub folders that demonstrate how to build ASAP external device driver (EDD) applications. Note the EDD functionality has been deprecated.</td>
</tr>
<tr>
<td>ASAP_home/samples/JeNEP</td>
<td>This directory contains Java Network Element Processor (NEP) samples and instructions that demonstrate support for CORBA, Lightweight Directory Access Protocol (LDAP), Socket, and Telnet connection protocols to NEs.</td>
</tr>
<tr>
<td>ASAP_home/samples/JeNEP/async_ne</td>
<td>This directory contains a sample that illustrates how asynchronous messaging to and from an NE can be configured with ASAP.</td>
</tr>
<tr>
<td>ASAP_home/samples/JeNEP/jenep_demo</td>
<td>This directory contains the sample CORBA, LDAP, Socket, and Telnet Java-enabled NEP (JNEP) directories. It also contains a IDLs folder for the CORBA sample.</td>
</tr>
<tr>
<td>ASAP_home/samples/JeNEP/jenep_demo/idls</td>
<td>This folder contains an IDLs file that supports the CORBA JNEP sample.</td>
</tr>
<tr>
<td>ASAP_home/samples/JeNEP/jenep_demo/LDAP</td>
<td>This folder contains the files required to configure the LDAP protocol for a JNEP.</td>
</tr>
<tr>
<td>ASAP_home/samples/JeNEP/jenep_demo/Socket</td>
<td>This folder contains the files required to configure a Socket based protocol for a JNEP.</td>
</tr>
<tr>
<td>ASAP_home/samples/JeNEP/jenep_demo/Telnet</td>
<td>This folder contains the files required to configure the Telnet protocol for a JNEP.</td>
</tr>
<tr>
<td>ASAP_home/samples/JeNEP/jenep_demo/PLSQL</td>
<td>This folder contains plsql scripts and other files that support the JNEP samples.</td>
</tr>
<tr>
<td>ASAP_home/samples/jsrp</td>
<td>This folder contains instructions for configuring and using the sample JSRP, including the JSRP Java Value Types (JVT) and Java Message Service (JMS) interfaces and sample JVT and JMS clients. When you compile the clients, and additional folder appears that contains the java classes for the clients.</td>
</tr>
<tr>
<td>ASAP_home/samples/jsrp/src</td>
<td>This folder contains the source files for the JSRP JVT and JMS clients.</td>
</tr>
<tr>
<td>ASAP_home/samples/jsrp/txt</td>
<td>This folder contains sample JVT WOs that you can use with the JVT client.</td>
</tr>
<tr>
<td>ASAP_home/samples/jsrp/xml</td>
<td>This folder contains sample JMS WOs that you can use with the JMS client.</td>
</tr>
<tr>
<td>ASAP_home/samples/nep</td>
<td>This folder contains sample files and instructions for creating a custom NEP.</td>
</tr>
<tr>
<td>ASAP_home/samples/sadt</td>
<td>This folder contains sample cartridges, sar files, and sample configuration files.</td>
</tr>
<tr>
<td>ASAP_home/samples/sadt/3GWireless</td>
<td>This folder contains a sample cartridge and subdirectories.</td>
</tr>
<tr>
<td>ASAP_home/samples/sadt/ATM_FR</td>
<td>This folder contains a sample cartridge and subdirectories.</td>
</tr>
<tr>
<td>ASAP_home/samples/sadt/DemoInstall</td>
<td>This folder contains a sample cartridge and subdirectories. This cartridge is installed by default when you install ASAP.</td>
</tr>
<tr>
<td>ASAP_home/samples/sadt/IPVPN</td>
<td>This folder contains a sample cartridge and subdirectories.</td>
</tr>
<tr>
<td>Directory</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>ASAP_home/samples/sadt/NetworkManager</td>
<td>This folder contains a sample cartridge and subdirectories.</td>
</tr>
<tr>
<td>ASAP_home/samples/sadt/SampleCommonConfig</td>
<td>This folder contains sample server configuration xml files. You can use these sample files to modify, delete, or add ASAP servers with Service Activation Configuration Tool (SACT).</td>
</tr>
<tr>
<td>ASAP_home/samples/sadt/sar</td>
<td>This file contains the SAR files for the sample cartridges.</td>
</tr>
<tr>
<td>ASAP_home/samples/sadt/sample_configs</td>
<td>This file contains sample ASAPcfg, ASAP.properties files for small, medium, and large ASAP deployments.</td>
</tr>
<tr>
<td>ASAP_home/scripts</td>
<td>This folder contains various scripts. Some scripts are available for ASAP users (for example, asap_utils, start_asap_sys, and so on), and some scripts are only used by the ASAP installer.</td>
</tr>
<tr>
<td>ASAP_home/SYBASE</td>
<td>This folder contains Sybase open server and client subfolder. This folder also contains the interface file that contains IP and port configuration information for ASAP server.</td>
</tr>
<tr>
<td>ASAP_home/SYBASE/charset</td>
<td>This folder contains subfolders for supported Sybase character sets.</td>
</tr>
<tr>
<td>ASAP_home/SYBASE/config</td>
<td>This folder contains Sybase configuration files.</td>
</tr>
<tr>
<td>ASAP_home/SYBASE/locales</td>
<td>This folder contains Sybase language and localization information.</td>
</tr>
<tr>
<td>ASAP_home/SYBASE/OCS-16_0</td>
<td>This folder contains the Sybase software development kit (SDK) and subfolders.</td>
</tr>
<tr>
<td>ASAP_home/sys_test</td>
<td>Legacy test tools for C-based cartridges.</td>
</tr>
<tr>
<td>ASAP_home/uninstall</td>
<td>This folder contains files used to uninstall the ASAP UNIX servers. Note that the file to uninstall the WebLogic ASAP server components (asap_uninstall_WLS) is located in the ASAP_home/scripts directory.</td>
</tr>
<tr>
<td>ASAP_home/XERCES</td>
<td>This folder contains the Xerces libraries. ASAP uses Xerces to parse, validate, and manipulate XML.</td>
</tr>
<tr>
<td>ASAP_home/xml</td>
<td>This folder contains XML based files.</td>
</tr>
<tr>
<td>ASAP_home/xml/xsd</td>
<td>This folder contains the XML schemas that ASAP uses to validate data structures and formats for ASAP server components, SAR files, service models, and so on.</td>
</tr>
<tr>
<td>ASAP_home/xml/xslt</td>
<td>This folder contains xslt style sheets used by various scripts to transform xml documents into new xml documents. The export_tool.sh script located in the ASAP_home/scripts directory uses these files to convert database data into xml content that can be used by SACT.</td>
</tr>
</tbody>
</table>