Oracle® Communications ASAP

Developer's Guide

Release 7.4.1
G13675-01
March 2025

Oracle Communications ASAP Developer's Guide, Release 7.4.1

G13675-01

Contents

3 Shared Libraries

4 Provisioning Interfaces

5 Web Services

A Oracle Examples

Preface

This guide provides an overview of Oracle Communications ASAP (ASAP) application and library architecture, and describes ASAP database tables, shared libraries, provisioning and downstream interfaces, and web services.

Audience

This document is intended for developers, system integrators, and other individuals who implement ASAP.

Documentation Accessibility

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

Access to Oracle Support

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

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

Development Overview

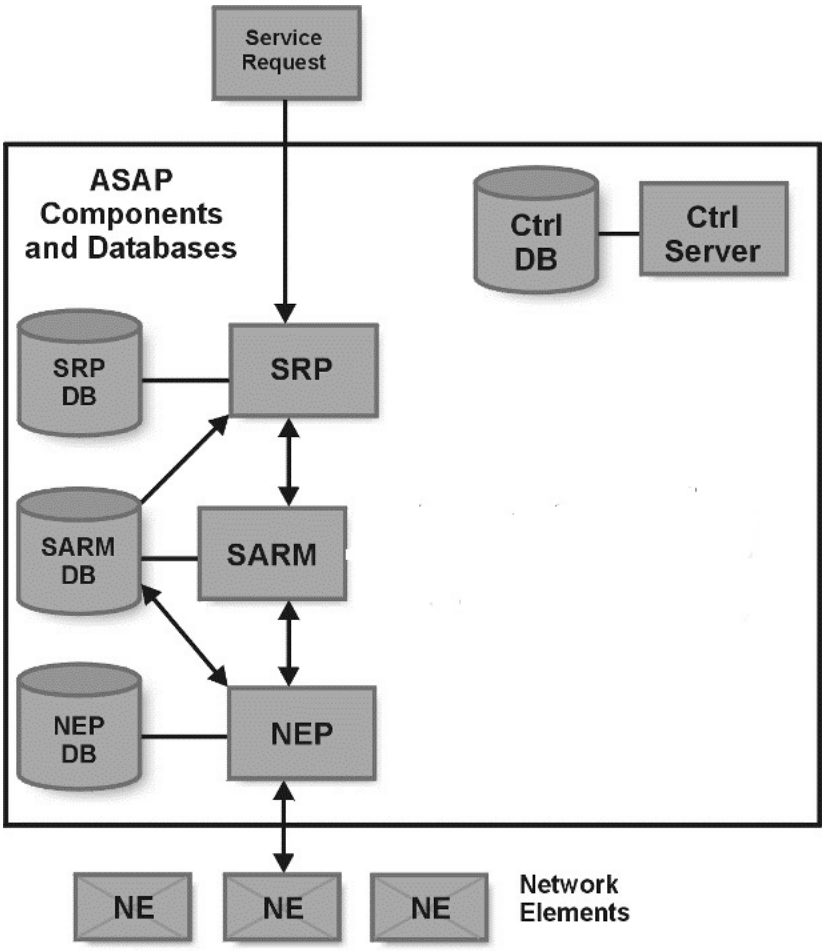
This chapter consists of the following sections:

Application Architecture

Figure 1-1 outlines how ASAP components interrelate. This figure displays all ASAP application processes (for example, SRP and SARM) and their appropriate databases.

ASAP contains several processes and each process has its own database. Because of this close coupling of a process to its database, ASAP is distributed efficiently in a network environment.

Figure 1-1 ASAP Processes and Databases



The database engine:

ASAP uses the SQL Server as the database engine. A single SQL Server can contain one or more ASAP databases. In a completely distributed environment, each ASAP database may reside on a separate SQL Server, on a separate machine. This ability to distribute and scale the ASAP application transparently, is a fundamental feature of ASAP's design.

The ASAP control server:

The ASAP Control server manages ASAP's overall operation. The Control server:

- Starts and stops ASAP applications.
- Distributes ASAP across many machines.
- Maintains process and performance statistics about each application.
- Provides event notification, logging, alarming, and paging facilities that the applications use.
- Monitors the behavior of application clients and application servers.
- Issues system alarms to the proper alarm centers if an ASAP application terminates unexpectedly.

Client/Server Architecture

ASAP consists of a set of multithreaded UNIX Client/Server processes that communicate with each other and the associated database servers.

The Client/Server architecture has several advantages over traditional program architectures:

- Client/Server applications, such as ASAP, are easily distributed across several, possibly heterogeneous, platforms. The applications have a scalable architecture which you can expand upon to meet your future requirements.
- Application size and complexity is reduced significantly because common services are handled in a single location, by the server. This simplifies client applications, reduces duplicate code, and makes application maintenance easier.
- Client/Server architecture enables applications to be developed with distinct components. These components can be changed or replaced without affecting other parts of the application. Such components may be supplied as part of the base product or developed by individual customers to suit their own requirements.
- Client/Server architecture facilitates communication between varied applications. Client applications that use different communication protocols from the server cannot communicate directly with it; instead they must communicate through a “gateway" server that understands both protocols.

Library Architecture

This chapter outlines ASAP's library architecture. The following sections are included in this chapter:

- ASAP API Development Structure
- API Library Structures
- ASAP API Application Development
- SRP Server Application Structure

- Generic NEP Application Structure
- Multi-Protocol NEP Structure

Development of Cartridges Supporting Asynchronous NEs

ASAP processes CSDLs and ASDLs in work orders synchronously. CSDLs and ASDLs are configured sequentially, and an ASDL must complete before the next ASDL can be started.

Some network elements respond to network actions asynchronously. After a request is sent, the *receipt* of the request may be acknowledged immediately but a response indicating *completion* of a request may be received from the network element some time later.

An ASDL in which the completion response arrives later is an asynchronous ASDL. A work order may have a mix of synchronous and asynchronous ASDLs mapped to the CSDLs.

Synchronous ASDLs require any previous ASDLs to be completed, so all asynchronous ASDLs run before a synchronous ASDL must complete before the synchronous ASDL starts.

> **Note:**
>
> Asynchronous Dynamic NEs are not supported.
>
> Asynchronous NE interfaces are supported through Java Enabled NEPs.

Asynchronous NE Response Handler

Instead of ASAP managing the processing of asynchronous ASDLs in the core, facilities are provided to the cartridge developer to handle them. These facilities include a response handler to manage asynchronous ASDL responses implemented in the java classes **ResponseHandler** and **ResponseHandlerManager**. Refer to the *ASAP Online Reference* for details on these classes.

When configuring asynchronous ASDLs, a 'stop work order' ASDL must follow the asynchronous ASDL(s) to halt work order processing. The NE response handler handles asynchronous responses from the NE, and resumes the work order when all outstanding asynchronous ASDL completion responses have been received.

Asynchronous connections:

Asynchronous NE interfaces have an entry in table **tbl_comm_param** with a parameter label **ASYNC_CONN** and parameter value of either **TRUE** or **FALSE**. See "tbl_comm_param."

When the NEP starts, for NEs that have an entry in table **tbl_comm_param** with a parameter label **ASYNC_CONN** and parameter value of TRUE, a connection is automatically established. If the parameter value is FALSE, the NE connection is not automatically established.

Each distinct asynchronous NE connection may have a distinct response handler. In this case, when an NE connection is released, then the associated response handler also should be stopped and removed from the system.

If multiple asynchronous NE connections use the same response handler, then the response handler may be spawned during the first NE connection and either left to keep running or stopped and removed from the JNEP by the cartridge developer when conditions as determined by the developer are met.

Response handler manager

Each Java-enabled NEP server (JNEP) uses a single instance of a response handler manager to manage all asynchronous response handlers within the JNEP. The response handler manager is accessed through static methods in its class. Typically, response handler creation is requested within the connect() method of the NE connection classes.

The response handler manager is implemented in the **ResponseHandlerManager** class. For details, refer to the *ASAP Online Reference.*

A sample implementation is provided with the ASAP product in

$ASAP_BASE/samples/JeNEP/async_ne

2

ASAP Database Tables

This chapter contains table information for the following databases:

- Control Database
- SARM Database
- NEP database

Control Database

This section describes the user-created database tables contained in the Control Database Data Dictionary.

User-created database tables

Following is the list of user-created database tables.

tbl_alarm_center

This static table defines the alarm centers where system alarm notifications are sent. You are responsible for maintaining this table. The initial data for this table is provided as part of the core system.

Table 2-1 tbl_alarm_center

Column_name	Type	Length	Nulls	Description
alarm_center	varchar2	8	0	The unique code representing the alarm center.
control_prog	varchar2	14	0	The program to be run to communicate to the alarm to this alarm center. For example, this could be a shell script written to send a message over the network, to a printer or pager, and so on., although any UNIX program can be provided. The alarm program passes alarm data as command line arguments for interpretation by the alarm program.
description	varchar2	80	1	A description of the alarm center.
opt#_type	varchar2	2	1	The option name passed to the control program, where # is a number between 1 and 5. This is a UNIX executable option to the control program. Can be null or any single wildcard character ("-_"), for example, -d -w, and so on.
opt#_value	varchar2	20	1	The option value that is passed to the control program corresponding to the previous option type, where # is a number between 1 and 5.

Table indexes:

```
alarm_center
```

The following is an example of alarm centers **tbl_alarm_center**:

```
alarm_Center control_prog  description              opt1_type
 opt1_value      opt2_type opt2_value      opt3_type
 opt3_value      opt4_type opt4_value      opt5_type
 opt5_value
 ------------ -------------- ------------------------------------- ---------

 ADMIN     admin.sh    Administration Alarm           NULL
           NULL          NULL    NULL                 NULL
           NULL          NULL    NULL                 NULL
           NULL
 ADMINPGR   adminp.sh   Administration Pager            NULL
            NULL          NULL    NULL                 NULL
            NULL          NULL    NULL                 NULL
            NULL
```

Each alarm center can invoke a program. This program is a UNIX executable or shell script. If set to **NULL**, no alarm program is invoked. In this example, two alarm programs, **admin.sh** and **adminp.sh**, are defined.

tbl_alarm_log

This dynamic table contains all of the system generated alarms.

Table 2-2 tbl_alarm_log

Column_name	Type	Length	Nulls	Description
event_unid	number	38	0	The unique ID of the event generating the alarm. If an alarm is generated by this event, an alarm log entry is created in tbl_alarm_log with this event_unid value.
alarm_code	varchar2	8	0	The alarm code of the generated alarm.
alarm_unid	number	38	0	The unique ID that identifies the alarm.
start_dts	date	-	0	The start date and time of the system alarm.
escalation_dts	date	-	1	If set, the date and time of the last alarm escalation.
clear_dts	date	-	1	If set, the date and time at which the alarm was cleared.
ack_user	varchar2	30	1	The user ID of the user who acknowledged the alarm.

Table indexes:

```
Indexes

alarm_unid
event_unid
alarm_code
```

tbl_appl_proc

This static table contains ASAP application configuration information. The ASAP startup procedure uses this table to determine the applications to start, and their start sequence.

You are responsible for maintaining this table. The initial data for this table is provided as part of the core system.

Table 2-3 tbl_appl_proc

Column_name	Type	Length	Nulls	Description
start_seq	number	38	0	Controls the sequence in which the applications are started. For example, certain client applications may be required to start before server applications, and other client applications after the server applications.
appl_type	char	1	0	Specifies the ASAP application type. • S – server • C – client • M – primary control server • R – remote control server
appl_cd	varchar2	8	0	The logical ASAP application code, for example, SARM, NEP01, NEP02, etc. The ASAP configuration file contains entries for the real name of each application. The real name appears as a parameter value for the application's logical name.
control_svr	varchar2	8	0	The name of the logical ASAP application Control server that spawns the application and monitors its behavior. In a distributed ASAP configuration, there must be a Control server defined on each machine. In each ASAP configuration there is only one primary Control server. This server is defined as having itself as a Control server. Remote Control servers are defined as having the primary Control server as their Control server.
description	varchar2	80	1	A description of the ASAP application.
diag_file	varchar2	14	0	The name of the diagnostics log file where diagnostic messages are written. This file is created in the $LOGDIR directory under a dated directory, for example, $LOGDIR/yymmdd.
auto_start	char	1	0	An auto start flag for the application. Values: • Y – yes • N – no When starting ASAP with the command start_asap_sys, all applications with the autostart value of Y are started automatically. To configure the SRP in tbl_appl_proc, you must set auto_start to 'N'.

Table 2-3 (Cont.) tbl_appl_proc

Column_name	Type	Length	Nulls	Description
program	varchar2	40	0	The name of the UNIX executable to run. This UNIX program must reside in the $PROGRAMS directory and must be executable.
diag_level	varchar2	4	0	The diagnostic level of the ASAP application. The diagnostic level determines whether to log diagnostic information when the ASC_diag() API function call is used. Possible values are: • PROG • SANE • LOW • KERN
isactive	char	1	0	Flag denoting whether the ASAP application is currently active. This stops attempts at starting more than one instance of the same application. Set by CSP_start_server/CSP_start_client and reset by CSP_halt_server/CSP_halt_client functions. Possible values are: • N – ASAP application is not running. • Y – ASAP application is currently running.
last_start	date	-	1	If set, identifies the last start date and time of the ASAP application. It is set by the administrative Control function, CSP_start_server.
last_halt	date	-	1	If set, identifies the last halt or termination date and time of the ASAP application. It is set by the administrative Control function CSP_halt_server/CSP_halt_client.
last_abnormal	date	-	1	If set, identifies the last abnormal termination of the ASAP application process. This is determined by the ASAP application Control server.
svr_type	varchar2	8	1	Possible values include: • CTRL – Control server • MASTER – primary Control server (must be only one/system) • SARM – SARM server • SRP – SRP server • NEP – NEP server

Table indexes:

```
Indexes

appl_cd
start_seq
```

tbl_classA_secu

tbl_classA_secu is an internal ASAP table that contains login ID and password information for each ASAP server.

ASAP class A secure data includes the database login/password for each ASAP server. This information is initially configured during ASAP installation.

The ASAP security administrator can maintain secure data by using the ASAP security tool. The ASAP security tool enables users to initialize and maintain security for ASAP Class A components (each core ASAP server). Refer to the *ASAP System Administrator's Guide* for more information.

Table 2-4 tbl_classA_secu

Column_name	Type	Length	Nulls	Description
name	varchar2	80	0	Name of key entry.
value	varchar2	255	0	Encrypted value.
class	integer	-	0	Possible values: • 0 – ASAP class A secure data.
secu_level	integer	-	0	One of the following: • 0 – Security feature is disabled. Password information appears in plain text • 1 – Security feature is enabled and passwords are encrypted.
s_cache	integer	-	0	Reserved for future use.
audit_level	integer	-	Reserved	Reserved for future use.
c_date	date	-	0	Creation date.
alg	integer	-	0	Type of cypher algorithm, currently supports only BLOWFISH alg (1).
desc1	varchar2	255	1	Description.

Table indexes:

```
name, value
```

tbl_classB_secu

tbl_classB_secu contains user ID, password, and other information used for custom components, particularly the Java-enabled NEP, and NEs.

User-defined secure data can be set up in the following ways: using the ASAP security administration tool or through APIs or action functions.

To import a large amount user-defined secure data into ASAP secure storage, it is recommended that the user compose a flat file containing essential secure data information. The format is "name:value:description" with semicolons used as delimiters. For example, a data file will reference secure data as follows:

```
TOR_NE:password1:Class B NE login info
ENG_NE:password2:Class B NE login info
```

> **Note:**
>
> The ASAP security administration tool encrypts the value field.

Refer to the *ASAP System Administrator's Guide* for more information.

Table 2-5 tbl_classB_secu

Column_name	Type	Length	Nulls	Description
name	varchar2	80	0	Name of key entry.
value	varchar2	255	0	The encrypted value.
class	integer	-	0	Reserved for future use.
s_cache	integer	-	0	Reserved for future use.
c_date	date	-	0	Creation date.
desc1	varchar2	255	1	Description.

Table indexes:

```
name, value
```

tbl_code_list

This static table is maintained within the Control database. It is used to track core and custom code used in ASAP. For instance, this table can identify code that tracks cartridges deployed within ASAP.

Table 2-6 tbl_code_list

Column_name	Type	Length	Nulls	Description
code_type	varchar2	4	0	Type of code.
code	varchar2	80	0	The code entry.
value	varchar2	4	0	The code value.
code_desc	varchar2	25	1	A description of the code.
parm1	varchar2	25	1	General purpose parameter field.
parm2	varchar2	25	1	Same as parm1.
parm3	varchar2	25	1	Same as parm1.
parm4	varchar2	25	1	Same as parm1.

Table indexes:

```
code_type, code, value
```

tbl_component

This static table contains a list of ASAP processes for each ASAP territory and system. You are responsible for populating and maintaining this table.

Table 2-7 tbl_component

Column_name	Type	Length	Nulls	Description
territory	varchar2	20	0	The name of the ASAP territory. This allows the components of all ASAP systems in all territories to be specified and maintained in a single database. It is possible to have many territories in an ASAP implementation. Territories are generally mutually exclusive since there is no communication between ASAP systems in different territories.
system	varchar2	20	0	The name of the ASAP system within an ASAP territory.
component	varchar2	40	0	The ASAP application component within this territory. This refers to the application components listed in the appl_cd column in tbl_appl_proc.

Table indexes:

```
territory, system, component
```

tbl_event_log

This dynamic table contains a log of all system events generated by ASAP applications.

You can create these entries by using the event function from within the source code.

Table 2-8 tbl_event_log

Column_name	Type	Length	Nulls	Description
appl_cd	varchar2	8	0	The logical name of the ASAP application that generated the system event.
event_type	varchar2	8	0	The event type that specifies whether a system alarm is to be generated by accessing the static table, tbl_event_type.
event_unid	number	38	0	A unique ID of this event. If an alarm is generated by this event, an alarm log entry is created in tbl_alarm_log with this event_unid value.
source_file	varchar2	14	0	The source file name where the event was generated.
source_line	number	38	0	The line in the source file where the event was generated.
reason	varchar2	80	0	A description of the system event.
event_dts	date	-	0	The date and time of the system event.
ack_user	varchar2	30	1	ID of the user who acknowledged the event.

Table indexes:

```
Indexes
```

```
event_unid
appl_cd, event_dts
```

tbl_event_type

This static table defines the system events an ASAP application may generate and if required, the system alarm code associated with the event. You can generate system events using customer-specific code. You can configure these system events to enable or disable system alarms, as required.

System events are logged in the Control database table, tbl_event_log. The ASAP core system specifies the system events generated by the core system. You are responsible for updating the existing system events and adding your own specific events.

This table is referenced by the customEvent element in ServiceModel.xsd. In other words, The customEvent must reference an event that has been defined in tbl_event_type in the control database.

Table 2-9 tbl_event_type

Column_name	Type	Length	Nulls	Description
event_type	varchar2	8	0	The ASAP event type generated by the application. Each system event must have a record in this table. The core API includes the following system events: • ABNORMAL – Abnormal Process Termination - Application Terminated Unexpectedly. • SYS_TERM – Application Self Termination upon Critical Error. • SYS_ERR – General Application Process Error. • SYS_WARN – General Application Process Warning. • SYS_INFO – General Application Information. • DISK_ERR – Critical Disk / Media Error. • RPCSPACE – Critical Database Resource Error. • RPC_ERR – General Application RPC Error. • UNIX_ERR – UNIX System Call Error. • NETWK_ERR – Application Network Connection Error. • SRVOBJER – Application Server Object Access Error. The Control server can issue the following events: • APP_STRT – ASAP Application Local or Remote Startup. • APP_STOP – ASAP Application Local or Remote Shutdown. • APP_ERR – ASAP Application Startup Error.

Table 2-9 (Cont.) tbl_event_type

Column_name	Type	Length	Nulls	Description
event_type (continued)	varchar2	8	0	The SARM can issue the following events: • WOINPROC – Warning: Work Order(s) in Progress Longer than Specified Threshold. • ROUT_ERR – Error: SARM Routing Error - Unable to Determine Host NE. The NEP can issue the following events: • MAINTNCE – Information: Host NE has gone into Maintenance Mode. • BIND_ERR – Warning: Unable to Allocate Device to Connect to NE. • CONN_ERR – Warning: NE Connection Attempt Failed. • DIAL_ERR – Warning: Dial-up Program to Connect to NE Failed. • SYS_TUNE – Informational message that ASAP is auto-tuning its connection pools. • LOGN_ERR – Warning: Login Program to Network Element Failed. • PORT_DIS – Error: Connection to NE Failed; Port/Device DISABLED.
description	varchar2	80	1	A description of the event. If an alarm is generated by this event, the description is passed to the alarm program.
alarm_code	number	8	1	The alarm code associated with the system event. If null, a database log entry is created, but no alarm is generated.
alarm_action	varchar2	2	1	The alarm action; specifically, the enabling or disabling of the associated system alarm. It allows system events to generate and stop system alarms. Possible values are: • E – Enable an alarm. • D – Disable an alarm.
notify_aims	char	0	1	Reserved for future use.

Table indexes:

```
event_type
```

The following is an example of the tbl_event_type table:

```
event_type description                                 alarm_code alarm_action
 ---------- ---------------------------------------- ---------- ------------
ABNORMAL  Abnormal Process Termination - Application     ABNORMAL   E
DB2FULL   Database Space is Critical                     CRIT_NAC   E
DISK_ERR  Critical Disk / Media Error - See Diagno       CRIT_AC    E
FS2FULL   File System Space is Critical                  CRIT_NAC   E
MAINTNCE  Information: Host NE has gone into Maint       MIN_AC     E
NTWK_ERR  Application Network Connection Error - S       MIN_AC     E
ROUT_ERR  Error: SARM Routing Error - Unable to De       MIN_AC     E
RPCSPACE  Critical Database Resource Error - See D       CRIT_NAC   E
RPC_ERR   General Application RPC Error - See Diag       MIN_AC     E
SRVOBJER  Application Server Object Access Error -       MIN_AC     E
SYS_ERR   General Application Process Error              MAJ_AC     E
```

```
SYS_INFO  General Application Information
SYS_WARN  General Application Process Warning                    E
TRANFULL  Tranlog Space is Critical                CRIT_NAC      E
UNIX_ERR  UNIX System Call Error - See Diagnostics MIN_AC        E
WOINPROC  Work Order(s) in Progress Longer than Sp MIN_AC        E
```

tbl_listeners

This table allows any ASAP server to set up socket listeners for receiving RPC requests. You must configure this table to allow the SARM to start up socket listeners for incoming SRP requests. As well, every Java-enabled NEP must maintain a dedicated connection to its JInterpreter.

Table 2-10 tbl_listeners

Column_name	Type	Length	Nulls	Description
srv_name	varchar2	8	0	Name of the server that starts a socket listener. The SARM must start a socket listener to receive incoming Java SRP requests. For a Java-enabled NEP, this is the name of the NEP ($NEP). For the Java SRP, this column contains the SARM name.
host_name	varchar2	80	0	The host name or the IP address on which the server application resides. For the JInterpreter, this value must always be localhost. For the Java SRP, the host_name identifies the location of the SARM.
listener_name	varchar2	40	0	The name of the listener thread. For a Java-enabled NEP, the listener name describes the listener in the Java process that accepts interpreter requests from the C process. This listener name must always be $NEP_jlistener. For the Java SRP, observe the naming convention of "<Java SRP application name>_jsrplistener". This column is used by the Java SRP to retrieve the listener configurations.
port	integer		0	A free port on which the server can start the socket listener.

Table indexes:

```
srv_name, listener_name
```

tbl_name_value_pair

This static table provides the facility to maintain miscellaneous name value pair information related to the Control database. You are responsible for maintaining this table.

Table 2-11 tbl_name_value_pair

Column_name	Type	Length	Nulls	Nulls
name	varchar2	40	0	The name of the parameter.
value	number	38	0	The value of the parameter.

Table indexes:

`name`

tbl_process_info

This table contains information on process system resource usage. This information is updated according to the frequency of polling, as specified by the PERF_POLL_PERIOD configuration variable.

Table 2-12 tbl_process_info

Column_name	Type	Length	Nulls	Nulls
appl_cd	varchar2	8	0	The name of the ASAP server.
info_dts	date	-	0	The date and time of the ASAP process performance polling.
sys_events	smallint	-	0	The total number of system events generated by appl_cd since ASAP start up.
user_cpu	integer	-	0	User CPU time.
system_cpu	integer	-	0	System CPU time.
proc_identity	number	20	0	The ID of the process.

Table indexes:

`appl_cd, info_dts`

> **Note:**
>
> The user_cpu and system_cpu are gathered by each ASAP server by using the times() UNIX system call. The times are in units of 1/CLK_TCK seconds.

tbl_server_info

This static table stores OCA SRP server information required for the OCA applet to connect to different sessions. When the OCA SRP server starts, it sends an HTTP request to the IORManager servlet, which extracts the following information from the HTTP request:

- OCA SRP server name
- Name of host where WebLogic Server is deployed
- Port number on which WebLogic Server is listening

The servlet then inserts the server, host name and port number string in the table.

When the OCA applet starts up, it reads the following information from the table.

Table 2-13 tbl_server_info

Column_name	Type	Length	Nulls	Description
servername	varchar2	8	0	The name of the server.
hostname	varchar2	80	0	The host on which the WebLogic server resides.
info	varchar2	512	0	The port number on which WebLogic Server is listening.

For more information on configuring the OCA, refer to the *ASAP Installation Guide*.

tbl_system_alarm

This static table describes ASAP system alarms that can be generated by ASAP system events. You must populate and maintain this table. Some initial data is provided in the core system.

Table 2-14 tbl_system_alarm

Column_name	Type	Length	Nulls	Description
alarm_code	varchar2	8	0	The alarm code.
description	varchar2	80	0	A description of the system alarm that is passed to the alarm program.
alarm_level	varchar2	8	0	The level of the alarm. Possible values are: • MINOR – minor alarm. • MAJOR – major alarm. • CRITICAL – critical alarm.
escalation_code	varchar2	8	0	Reserved for future use.
escalation_time	number	38	0	Reserved for future use.
auto_clear	char	1	Not applicable	If this flag is set to Y, it automatically clears the alarm after being generated; otherwise, the alarm continues to go off until it is either manually turned off or a system event is triggered to disable it. For more information, see "tbl_event_type." You can use a Control Server administrative RPC, alarm_stop, to turn off a particular non-auto clearing system alarm.
route#_period	number	38	Not applicable	Where # is a number between 1 - 5. The period in minutes from the alarm to the alarm center.
route#_start	number	38	Not applicable	Where # is a number between 1 - 5. The daily start time in minutes after midnight for the alarms to go to the alarm center. A value between 0 and 1440.
route#_end	number	38	Not applicable	Where # is a number between 1 - 5. The daily end time in minutes after midnight for the alarms to go to the alarm center. A value between 0 and 1440.

Table 2-14 (Cont.) tbl_system_alarm

Column_name	Type	Length	Nulls	Description
route#_center	varchar2	8	Not applicable	Where # is a number between 1 - 5. The alarm center to route alarms.

Table indexes:

```
alarm_code
alarm_code description               alarm_level escalation_code

escalation_time auto_clear route1_period route1_start route1_end
route1_Centre route2_period route2_start route2_end route2_Centre

---------- ---------------------------------------- ----------- ---------------
ABNORMAL  Abnormal Process Termination       CRITICAL

 NULL N           5      0    1440
 ADMINPGR        NULL     NULL    NULL

CRIT_NAC  Critical ASAP System Alarm        CRITICAL

 NULL N           5      0    1440
 ADMINPGR        NULL     NULL    NULL

CRIT_AC  Critical ASAP System Alarm        CRITICAL

      NULL Y           5      0    1440
 ADMINPGR        NULL     NULL    NULL

MAJ_NAC  Major ASAP System Alarm         MAJOR

      NULL N           5      0    1440
 ADMIN         NULL     NULL    NULL

MAJ_AC   Major ASAP System Alarm         MAJOR

      NULL Y           5      0    1440
 ADMIN         NULL     NULL    NULL

MIN_NAC  Minor ASAP System Alarm         MINOR

      NULL N           5      0    1440
 ADMIN         NULL     NULL    NULL

MIN_AC   Minor ASAP System Alarm         MINOR

      NULL Y           5      0    1440
 ADMIN         NULL     NULL    NULL
```

tbl_unid

You can use this dynamic table to manage unique IDs needed by other tables. It is present in most user databases and provides a method of generating a serial field.

Table 2-15 tbl_unid

Column_name	Type	Length	Nulls	Description
unid_type	char	32	1	The unique ID type. You can maintain many different types of unid values with each generating a serial field by means of a suitable function.
unid	number	38	1	The current unid value for the unid type.
pad#	char	255	1	Where # is a number between 1 - 4. Forces each record to reside on a separate database page (allowing improved database concurrency).

Table indexes:

`unid_type`

tbl_unload_param

This table is used by the utilities. It provides information about the parameters inserted or deleted by the stored procedures.

Table 2-16 tbl_unload_param

Column_name	Type	Length	Nulls	Description
seq_no	Number	38	0	Sequence Number of the table
col_no	Number	38	0	Column Number of the parameter in the table
para_name	Varchar2	80	0	Name of the Parameter
default_flag	Number	38	0	Default flag
sp_type	char	3	0	Type of the stored procedure
rows_int	Number	38	0	Number of Rows

Table indexes:

`seq_no, para_name, sp_type`

This table will be available in the Control and SARM databases. It contains data relevant to the respective database it is in.

tbl_unload_sp

This table provides information about the stored procedures used to insert and delete data from ASAP tables. This table is used by utilities which modify these ASAP tables.

Table 2-17 tbl_unload_sp

Column_name	Type	Length	Nulls	Description
seq_no	Number	38	0	Sequence Number of the table
Tbl_name	Varchar2	40	0	Name of the ASAP table

Table 2-17 (Cont.) tbl_unload_sp

Column_name	Type	Length	Nulls	Description
New_sp	Varchar2	40	0	Stored Procedure used to insert the data into the table
Del_sp	Varchar2	40	0	Stored Procedure used to delete data from the table
List_sp	Varchar2	40	0	Stored Procedure used to list the data in the table

Table indexes:

`seq_no`

This table will be available in the Control and SARM databases. It contains data relevant to the respective database it is in.

SARM Database

This section details the SARM database tables.

Work Order Audit Information

This section describes the various methods of extracting work order audit information from the following tables:

- tbl_wrk_ord – contains the current status of work orders.
- tbl_info_parm – contains information parameters that are returned to the SRP from NEP.
- tbl_srq_log – contains information pertaining to the system's interaction with NEs and resulting changes to work order status and switch history information of the NE responses.
- tbl_wo_audit – tracks the work order's status as it is processed. The level of information captured in tbl_wo_audit is governed by the ASAP configuration variable WO_AUDIT_LEVEL.

The level of logging for tbl_srq_log and tbl_wo_audit is governed by the ASAP.cfg configuration parameter WO_AUDIT_LEVEL. The following list explains the available audit levels:

- 0 – no auditing occurs. No information is placed in the tbl_wo_audit table.
- 1 – there is one audit entry per work order in the tbl_wo_audit table as it is tracked through the system.
- 2 – provides all functions of level 1 plus the audit level entries for all error states in the tbl_wo_audit table.
- 3 – provides all functions of level 2 plus it tracks the provisioning of a work order through the entire provisioning process in the tbl_wo_audit table. For example, when the ASDL was started, when it was placed in pending queue, where in the pending queue it is, when it was sent to the NEP etc.
- 4 – all events are inserted into the tbl_wo_audit table. This level is intended to debug the work order auditing process.

If the WO_AUDIT_LEVEL configuration parameter is set to 2 or greater, SRQ_ERROR_EVENTS are written to tbl_srq_log. Whenever it is set less than 2, WO_AUDIT_LEVEL events are not written into this table.

Events of other types, such as NE_RESP_EVENTs, NE_CMD_EVENTs and SRQ_INFO_EVENTs, are written into tbl_srq_log even when WO_AUDIT_LEVEL=0. These events track the information from the NEP's perspective, including:

- response logging (commands sent to and responses received from the NE)
- asdl_exit types and associated error messages
- information such as when a command was sent to the NE.

Viewing Work Order Audit Information

The OCA client provides a work order audit function that displays events related to the processing of work orders contained in tbl_wo_audit. These events include, but are not limited to, the event text and the data and time of these events.

The asap_utils utility provides additional functions that are particularly related to the processing of work orders.

- **2. SARM - In Proc Requests Summary** – lists the number of requests currently in progress in the ASAP database (this is determined to be orders in a Loading or In Progress state).
- **4. SARM - Work Order Queue Summary** – lists the number of orders in each of the SARM order queues. The queues include:
  - Ready Queue – orders currently in progress.
  - Rollback Queue – orders currently being rolled back.
  - Auto Held Queue – orders that are being held by the SARM and not released for some reason
- **5. SARM - Work Order Queue Details** – provides the order details of each order in the SARM work order queues. Such queues are global to the SARM.
- **6. SARM - Work Order Lock States** – lists the orders in progress and their respective lock states. Generally, an in progress order will have a local lock. Only in the high availability configuration will orders be remotely locked.
- **7. SARM - ASDL/NE Queue Summary** – provides summary details about each NE in the system including:
  - the NE, technology and software load
  - the NEP managing the NE
  - the current state of the NE
  - Down, Connecting, Available, Maintenance, Disabled
  - the time estimate (sec.) for ASDL processing to that NE
  - the number of ASDLs pending to that NE (in a prioritized queue), the number currently in progress, the number of connections open to that NE, and the number of ASDLs waiting to be retried to that NE.
- **8. SARM - ASDL/NE Queue Details** – provides details about each ASDL in the Pending, In Progress and Retry ASDL queues to each NE in the system.

A complete description of these and other asap_utils functions is located in the *ASAP System Server Configuration Guide*.

The level of logging is governed by the ASAP.cfg configuration parameter WO_AUDIT_LEVEL. The following list explains the available audit levels:

- 0 – no auditing occurs. No information is placed in the tbl_wo_audit table.
- 1 – there is one audit entry per work order in the tbl_wo_audit table as it is traced through the system.
- 2 – provides all functions of level 1 plus the audit level entries for all error states in the tbl_wo_audit table.
- 3 – provides all functions of level 2 plus it tracks the provisioning of a work order through the entire provisioning process in the tbl_wo_audit table. For example, when the ASDL was started, when it was placed in pending queue, where in the pending queue it is, when it was sent to the NEP etc.
- 4 – all events are inserted into the tbl_wo_audit table. This level is intended to debug the work order auditing process.

If the WO_AUDIT_LEVEL is set to 2 or greater, SRQ_ERROR_EVENTS are written to tbl_srq_log. Whenever it is set less than 2, WO_AUDIT_LEVEL events are not written into this table.

Events of other types, such as NE_RESP_EVENTs, NE_CMD_EVENTs and SRQ_INFO_EVENTs, are written into tbl_srq_log even when WO_AUDIT_LEVEL=0. These events track the information from the NEP's perspective, including:

- response logging (commands sent to and responses received from the NE)
- asdl_exit types and associated error messages
- information such as when a command was sent to the NE.

SARM Database Tables

This section details the SARM database tables.

tbl_asap_srp

This static ASAP database table defines all Service Request Processors (SRPs) currently configured in the ASAP system. Any ASAP application process that communicates with the SARM as an SRP using the SRP API must be defined in this table.

Upon start up, the SARM opens one or more network connections to each SRP defined in this table. For each SRP, you can configure the SARM to issue an ASAP system event with each work order notification transmitted back to the SRP. The following are possible system events:

- NO EVENT – No event is sent to the SRP.
- NULL – The event is sent to the SRP or no system event is generated.
- Others – The event is sent to the SRP or the system event is generated of the type specified. This allows the SARM to be configured by the SRP and not globally as it would if this was performed by means of a configuration variable.

For more information on event types, see "tbl_event_type."

The Java SRP must be configured to be a socket type connection.

You are responsible for populating and maintaining this table.

You can maintain this table in the following ways:

- Using the Service Activation Configuration Tool (SACT). Refer to the *ASAP Server Configuration Guide* for instructions.
- Using stored procedures, including:
  - **SSP_new_srp** adds new SRP definitions.
  - **SSP_del_srp** deletes SRP definitions from this table.
  - **SSP_list_srp** lists the contents of this table.

Table 2-18 tbl_asap_srp Columns

Column_name	Type	Length	Nulls	Description
srp_id	varchar2	8	0	The logical SRP server name.
srp_desc	varchar2	255	1	A description of the SRP.
srp_conn_type	char	1	0	Connection protocol from the SARM to the SRP. Valid values are: • O – for Open Client. • S – for sockets.
srp_host_name	varchar2	80	1	Name of the machine that the SRP resides upon.
srp_host_port	varchar2	6	1	The port number that the SRP is waiting on for socket connections.
wo_estimate_evt	varchar2	8	1	Work order estimate event. If you configure the SARM using the configuration variable WO_TIME_ESTIMATE_ON to perform this work order estimation calculation, the SARM determines the approximate number and processing time for ASDLs on the work order. The SARM then calculates an approximate time estimate for this work order.
wo_failure_evt	varchar2	8	1	Work order failure event.
wo_complete_evt	varchar2	8	1	Work order complete event.
wo_start_evt	varchar2	8	1	Work order start event. The default value is NO EVENT. If this event is required by the SRP, you must specify NULL.
wo_soft_err_evt	varchar2	8	1	This notification is generated whenever an ASDL response on the work order is 'Fail But Continue'. If the work order fails or completes, then this notification is followed by a Failure or Completion notification. If this event is not required by the SRP, you must specify NO_WO_EVENT_NOTIFICATION (NO EVENT).
wo_blocked_evt	varchar2	8	1	Not supported.

Table 2-18 (Cont.) tbl_asap_srp Columns

Column_name	Type	Length	Nulls	Description
wo_rollback_evt	varchar2	8	1	Generated when a work order fails and rollback is configured on one or more CSDLs on the order. It is also generated when a completed work order is cancelled and the SARM rolls back any completed ASDLs on the work order. If this event is not required by the SRP, specify NO_WO_EVENT_NOTIFICATION (NO EVENT).
wo_timeout_evt	varchar2	8	1	Work order timeout event.
ne_unknown_evt	varchar2	8	1	If defined, the system event issued by the SARM when an NE unknown notification is being transmitted back to the SRP from the SARM. This notification is transmitted to the SRP upon a Remote to Host NE, or DN to Host NE routing error.
ne_avail_evt	varchar2	8	1	No longer used.
ne_unavail_evt	varchar2	8	1	No longer used.
wo_accept_evt	varchar2	8	1	The default value is NO EVENT. If this event is required by the SRP, you must specify NULL. If defined, the system event issued by the SARM when a work order acceptance notification is being transmitted back to the SRP from the SARM.
aux_srp_id	varchar2	8	1	Specifies the auxiliary SRP to where the SARM opens network connections in the event that the primary SRP is unavailable. Set this field to NULL.
aux_srp_conn_type	char	1	1	Connection protocol for SARM to auxiliary SRP communications. Valid values are: • O – for Open Client. • S – for sockets.
aux_srp_host_name	varchar2	80	1	Name of the machine that the auxiliary SRP resides upon.
aux_srp_host_port	varchar2	6	1	The number of the port that the auxiliary SRP is waiting on for socket connections.

Table indexes:

`srp_id`

tbl_asap_stats

tbl_asap_stats provides a way of gathering and analyzing statistics related to the ASAP provisioning process.

If you enable the ASAP statistics gathering by using the NEP configuration variable ASAP_STATS_ON, the dynamic table will contain a log of all the ASAP statistical entries. Statistical entries are made by the Interpreter and are passed to relevant data fields.

Table 2-19 tbl_asap_stats Columns

Column_name	Type	Length	Nulls	Description
stats_unid	number	38	0	The unique ID of the statistical entry which is generated at insert time.
stats_dts	date	-	0	The date and time that the statistical entry was logged.
wo_id	varchar2	80	0	The work order to which this statistical entry is related. This is determined from the work order table at insert time.
wo_stat	number	1	0	The status of the associated work order. This is determined from the work order table at insert time. An update trigger updates the work order status to match that of the work order. This means that when a work order is completed, all statistical entries associated with that work order is updated to 'Completed'. The possible values for this field are the same as those in wo_stat in the SARM database table tbl_wrk_ord. See "tbl_wrk_ord (SARM)."
user_id	varchar2	64	0	The user ID associated with this log entry. The user ID is determined from the orig_login field in the SARM database table tbl_wrk_ord at insert time. See "tbl_wrk_ord (SARM)."
srq_id	number	38	0	The service request related to this statistical entry.
asdl_cmd	varchar2	80	0	The ASDL being provisioned when the statistics log entry was created.
mcli	varchar2	80	1	If set, the Remote NE.
hcli	varchar2	80	0	The Host NE to which the ASDL currently being processed is routed.
dn	varchar2	24	1	If set, the directory number currently being provisioned.
len	varchar2	16	1	If set, the line equipment number currently being provisioned.
ne_cmd	varchar2	255	1	If set, the NE command string sent to the NE.
cmd_type	varchar2	25	1	If set, the type of NE command string. The possible values which form the basis for simple statistical gathering include: • ADD – service addition. • REMOVE – service removal. • CHANGE – service change. • QUERY – query only, no provisioning performed. You can customize this field and also specify other command types to better suit your statistical requirements.

Table 2-19 (Cont.) tbl_asap_stats Columns

Column_name	Type	Length	Nulls	Description
cmd_stat	varchar2	25	1	If set, identifies the status of the NE command after transmission to the NE. The possible values which form the basis for statistical gathering include: • SUCCEED – Command completed successfully. • FAIL – Command failed. • RETRY – Command failed but retried. • MAINTENANCE – Command failed because the NE is currently unavailable to receive provisioning requests. • SOFT_FAIL – Command failed but processing continues on other ASDLs. • DELAYED_FAIL – An ASDL had failed during provisioning. The SARM skips any subsequent ASDL in the CSDL, continues provisioning at the next CSDL, and then fails the order. This field is customizable. You can specify other command states to better suit your statistical requirements. Refer to the *ASAP Cartridge Development Guide* for more detailed descriptions of these base_types.
cmd_reason	varchar2	255	1	If set, identifies the reason that the NE command failed. This is the error string returned from the NE. You can generate reports to determine the principal error conditions in the provisioning process.
parm1	varchar2	25	1	Multi-purpose text field in which customer specific information is logged. This field logs additional information in this table. To allow this in a flexible manner, these miscellaneous parameter fields have been provided.
parm2	varchar2	25	1	Same as parm1.
parm3	varchar2	25	1	Same as parm1.
parm4	varchar2	25	1	Same as parm1.
parm5	varchar2	25	1	Same as parm1.
parm6	varchar2	25	1	Same as parm1.

Table indexes:

```
Indexes

stats_unid
wo_id
wo_stat, stats_dts
cmd_stat, stats_dts
```

tbl_asdl_config

This static table defines the ASDL configuration information required to handle routing and rollback at the ASDL level. It is used by the SARM to determine which NEP to route the command to, whether rollback is required for this ASDL, and if so, the rollback ASDL to use. You are responsible for populating and maintaining this table.

- **SSP_new_asdl_defn** adds new ASDL configuration information to this table.
- **SSP_del_asdl_defn** deletes ASDL configuration information from this table.
- **SSP_list_asdl_defn** lists the contents of this table.

Table 2-20 tbl_asdl_config Columns

Column_name	Type	Length	Nulls	Description
asdl_cmd	varchar2	80	0	The ASDL command.
reverse_asdl	varchar2	80	0	Only required if rollback is configured on the ASDL. If the ASDL requires rollback, the reverse ASDL command must be invoked.
ignore_rollback	char	1	0	Ignore rollback flag. Possible values are: • Y – Rollback is ignored for this ASDL. • N – Rollback is required for this ASDL. In this case, rollback is only performed if there is a valid rollback ASDL command defined. Rollback can be initiated for this ASDL in two cases. • The work order fails and rollback is configured on one or more CSDLs on the work order. In this case, if configured, this ASDL is rolled back. • When a cancellation is received on the work order. If this ASDL has been completed and rollback is configured, the SARM initiates rollback for this ASDL.
route_flag	char	1	0	Routing of the ASDL. Possible value is: • (N) ROUTE_TO_NEP – The ASDL is routed to the NEP only. This value is defined in the header file sarm_defs.h. Deprecated. Column maintained for backward compatibility only.
description	varchar2	255	1	A description of the ASDL.
asdl_timeout	number	8	0	The maximum number of seconds for an ASDL timeout. If there is no response in the given timeout value, the ASDL will timeout. For more information, see the *ASAP System Administrator's Guide*.
asdl_retry_number	number	8	0	The maximum number of retries if an ASDL requests timeout. If the number of retries exceeds asdl_retry_number, the order is failed and rolled back. For more information, see the *ASAP System Administrator's Guide*.

Table 2-20 (Cont.) tbl_asdl_config Columns

Column_name	Type	Length	Nulls	Description
asdl_retry_interval	number	8	0	The time period in seconds between ASDL retries. For more information, see the *ASAP System Administrator's Guide*.

Table indexes:

`asdl_cmd`

tbl_asdl_log

This dynamic table is used by the SARM to log the ASDLs as they are sent to the NEP to facilitate rollback.

Table 2-21 tbl_asdl_log Columns

Column_name	Type	Length	Nulls	Description
srq_id	number	38	0	The service request ID uniquely identifying where the ASDL belongs.
asdl_stat	number	1	0	Status of the ASDL. If defined, this field is updated while processing the ASDL and its rollback ASDL. Possible values include: • ASDL_NEP_COMPLETE 10 • ASDL_NEP_FAIL 11 • ASDL_NEP_RBACK_COMP 12 • ASDL_NEP_RBACK_FAIL 13 • ASDL_INITIAL 14 • ASDL_NEP_FAIL_CONTINUE 17 • ASDL_NEP_FAIL_DELAYED 20 These values are defined in sarm_defs.h.
asdl_unid	number	38	0	A unique ID identifying the ASDL log entry. This is a unique identifier and is assigned sequentially starting at the value1.
csdl_seq_no	number	38	0	The sequence number uniquely identifying the CSDL on the SRQ that generated this ASDL.
asdl_option	varchar2	255	1	Reserved.
asdl_cmd	varchar2	80	0	The ASDL command.
rollback_asdl	varchar2	80	1	If rollback is required on this ASDL, use this command to roll back the original ASDL. It is set by the SARM once the rollback is completed as a history record. This field is NULL unless rollback of this ASDL is performed.
comp_dts	date	-	1	The completion date and time of the ASDL processing.
rollback_dts	date	-	1	The date and time of the ASDL rollback.

Table 2-21 (Cont.) tbl_asdl_log Columns

Column_name	Type	Length	Nulls	Description
host_clli	varchar2	80	0	The Host NE to which the ASDL is routed by the SARM.
queue_dts	date	-	1	The date and time when the ASDL was placed in the SARM provisioning queue.
start_dts	date	-	1	The date and time when provisioning starts for this ASDL. The difference between this value and the queue_dts represents the time the ASDL spent in the queue of pending ASDLs before being transmitted to the NEP for provisioning.
retry_count	number	38	1	A count of the number of times the ASDL was retried at the NE.

Table indexes:

```
srq_id, asdl_stat, asdl_unid
```

tbl_asdl_parm

A static table used by the SARM to define the parameter labels and values associated with a given ASDL. It also provides the mapping between the CSDL parameter labels received from the SRP (csdl_lbl) and the ASDL parameter labels (asdl_lbl) transmitted to the NEP for interpretation by the Java methods.

For each CSDL label (csdl_lbl), the SARM checks the current CSDL parameter name value pairs for a matching label. If no matching label is found, it checks for a label in the work order global parameter name value pairs. If no matching label is found in either of these parameter name value pairs and the parameter type (param_typ) which is mandatory, the default value (default_vlu), is used.

If no default value is set, the SARM registers an ASDL parameter mapping failure. If the parameter is Indexed, the csdl_lbl must contain a “++" or the SARM will not start. There is considerable translation mapping logic related to both the tbl_csdl_asdl and this table.

> **Note:**
>
> By convention, the ++ notation appears at the end of the label within square brackets. This convention makes it easy to identify the index.

An example of this translation is contained in the *ASAP Cartridge Development Guide*. Refer to the section on dynamic routing scenarios.

You are responsible for populating and maintaining this table.

- **SSP_new_asdl_parm** adds ASDL parameters to tbl_asdl_parm.
- **SSP_del_asdl_parm** adds ASDL parameters to tbl_asdl_parm.
- **SSP_list_asdl_parm** lists the contents of this table.

Table 2-22 tbl_asdl_parm Columns

Column_name	Type	Length	Nulls	Description
asdl_cmd	varchar2	80	0	The ASDL command.
parm_seq_no	number	38	0	The sequence number of the ASDL parameter within the parameter list.
asdl_lbl	varchar2	80	0	The ASDL label used in transmitting the ASDL parameter name value pairs to the NEP for interpretation by the NE Java methods.
csdl_lbl	varchar2	80	0	The CSDL or global work order parameter label, which is transmitted to the SARM by the SRP or returned by the NEP Java methods as a return CSDL global parameter. For an indexed or compound parameter, the CSDL command defined in this table is not exactly the same as the label transmitted by the SRP. This table stores the CSDL's base name.
default_vlu	varchar2	255	1	In the case when the CSDL parameter is not passed to the SARM, the SARM substitutes the default value for the csdl_lbl. This default parameter value is referenced only for Scalar parameters.

Table 2-22 (Cont.) tbl_asdl_parm Columns

Column_name	Type	Length	Nulls	Description
param_typ	char	1	0	There are three ASDL parameter formats: • SCALAR – Specifies the parameter label transmitted on the ASDL command. • COMPOUND – Specifies the base name of the Compound parameter transmitted on the ASDL command. • INDEXED – Specifies the base name of the ASDL command transmitted on the ASDL command. ASDL parameters can be either required or optional. Consequently, the possible parameter type values include: • R – required scalar parameter. • O – optional scalar parameter. • C – required compound parameter. This parameter type is also used for compound indexed parameters. For more information, see the *ASAP Cartridge Development Guide*. • N – optional compound parameter. • M – mandatory indexed parameter. • I – optional indexed parameter. • X – required XML parameter. • Y – optional XML parameter. • P – required XPATH parameter. • Q – optional XPATH parameter. • S – Parameter count. This value gives the Java method, the total number of parameters associated with this ASDL command. • + – Current index value for this ASDL. Only applicable to indexed ASDLs. For more information about XML and XPATH parameters, see *ASAP Cartridge Development Guide*.
dep_asdl_lbl	varchar2	80	0	An ASDL parameter that identifies the XML document that the XPATH is evaluated against. Applies only to parameter types P and Q.

Table indexes:

```
asdl_cmd, parm_seq_no
```

For details and examples of various parameter types, refer to the *ASAP Cartridge Development Guide*.

tbl_asdl_response

In this ASDL Loopback testing table, the Interpreter must be placed in loopback mode by means of the Interpreter configuration parameter LOOPBACK_ON.

In loopback mode, when an ASDL completes successfully, the Interpreter refers to this table to determine the action to take before completing the Interpreter ASDL invocation.

This table specifies a time interval to wait, (useful to test time-out conditions), for ASDL states returned by the Interpreter processing. You only need to perform the test to populate this table when loopback testing is being performed.

Table 2-23 tbl_asdl_response Columns

Column_name	Type	Length	Nulls	Description
asdl	varchar2	80	0	The ASDL for which loopback processing is provided.
parm_option	varchar2	80	0	Reserved for future use.
exit_status	varchar2	20	0	The exit status returned to the Interpreter from the simulation. The possible values include: • MAINT – Simulates the NE having gone into Maintenance mode and allows testing of the conditions. • FAIL – Simulates a hard error. • SUCCEED – Default successful completion pf running the scripts. • STOP – Stops a work order after running the script is complete.
asdl_status	varchar2	20	1	The ASDL status returned to the SARM. This column is only used when exit_status is FAIL. The possible values include: • RETRY – The ASDL is retried by the SARM. You can configure both the retry period and number of retries within the SARM using the configuration variables (NUM_TIMES_RETRY and RETRY_TIME_INTERVAL). • CONT or SOFT_FAIL – The ASDL fails but provisioning continues, for example, attempting to remove a service that is not present. • DELAYED_FAIL – The ASDL fails but provisioning on the rest of the order continues, at the end of which the order will fail. • FAIL – The ASDL fails and all provisioning on this work order ceases, for example, an unrecoverable error occurred. • STOP – Stops a work order after running the script is complete. Refer to the *ASAP Cartridge Development Guide* for more detailed descriptions of these base_types.
asdl_time	number	38	0	The time interval, in seconds, that the Interpreter waits before returning a response to the invoking server. This interval is used to simulate a delay in the processing of the NE responses to mimic usual run-time conditions.

Table 2-23 (Cont.) tbl_asdl_response Columns

Column_name	Type	Length	Nulls	Description
error_text	varchar2	255	1	An error text description sent back to the SARM detailing the cause of the error. Error text can be viewed by the front-end user interface. If it is not explicitly specified, generic descriptive text is supplied.

Table indexes:

```
asdl, asdl_option
```

tbl_aux_wo_prop

tbl_aux_wo_prop is a class A dynamic table that serves as an extension to tbl_wrk_ord. tbl_aux_wo_prop was designed to accommodate additional pre-defined work order properties to supplement the ones contained in tbl_wrk_ord. Currently, the only extended property supported in tbl_aux_wo_prop is WO_SECURITY_PROP. This property is maintained for each work order. If WO_SECURITY_PROP = 0, then work order information is eligible to be output to diagnostic files. If WO_SECURITY_PROP = 1, then no work order information is written to diagnostic files.

Refer to the *ASAP System Administrator's Guide* for more information on secure work order information.

When the work orders in the tbl_wrk_ord are deleted, the corresponding records in tbl_aux_wo_prop are deleted. This table can be purged using function **SSP_db_admin**. For guidelines and instructions on database purging, refer to the *ASAP System Administrator's Guide.*

Table 2-24 tbl_aux_wo_prop Columns

Column Name	Type	Length	Nulls	Nulls
wo_id	varchar2	80	0	The ID of the work order property.
name	varchar2	80	0	The name of the work order property.
value	varchar2	255	1	The value given to the work order property.

Table indexes:

```
Non-unique

wo_id
```

tbl_blackout

This static table contains date and time periods for which a particular Host NE is deemed unavailable by the user for ASAP updates.

The NEP reads this table using the current time. If the current time is within a blackout period, the NEP returns a maintenance mode condition, therefore, disconnects from the Host NE. This process is continually retried until the Host NE is no longer blacked out. You can use this table when there are systems competing for limited ports on the Host NE. You are responsible for populating and maintaining this table.

Note that if the dayname entry is present, then the configuration is considered to be Static. The implementation will check for the start time and end time alone, and assume that both fall on the same day. When configuring a blackout period that spans from one day to the next (e.g. from 22:00 until 01:00 the next day) you must configure two separate lines in tbl_blackout: one for 22:00:00 until 00:00:00 and one from 00:00:00 until 01:00:00.

Table 2-25 tbl_blackout Columns

Column_name	Type	Length	Nulls	Description
host_clli	varchar2	80	0	The Host NE that is blacked out.
dayname	varchar2	10	1	If this field contains the valid name of a day of the week (for example, Sunday or Monday) then the date portions of start_tm and end_tm are ignored. If this field is empty than both the date and time portions from start_tm and end_tm will be used. If this field contains an invalid value, the blackout will not occur.
start_tm	date	-	0	The start date and time for the blackout interval.
end_tm	date	-	0	The end date and time for the blackout interval.
description	varchar2	255	1	A description of the NE blackout reason.

Table indexes:

```
host_clli, start_tm, dayname
```

tbl_clli_route

This static table contains the mapping between a Remote NE and its Host NE. It is used to determine the Host NE when a Remote NE is provided on the work order. Given the Remote NE and ASDL, this table provides the Host NE to which this ASDL is routed.

If no Remote NE is present on the work order, the table tbl_nep_rte_asdl_nxx is used to route the ASDL by DN. This is only performed if the parameter MCLI is present on the ASDL and is set to the special literal SKIPCLLI. You are responsible for populating and maintaining this table.

- **SSP_new_clli_map** adds new CLLI-to-Host CLLI mapping definitions to tbl_clli_route.
- **SSP_del_clli_map** deletes CLLI-to-Host CLLI mapping definitions in tbl_clli_route.
- **SSP_list_clli_map** lists the contents of this table.

Table 2-26 tbl_clli_route Columns

Column_name	Type	Length	Nulls	Description
mach_clli	varchar2	80	0	Remote NE.
host_clli	varchar2	80	0	Host NE to which the Remote NE is connected.
asdl_cmd	varchar2	80	1	The ASDL command for which this routing definition is valid. This is useful for services that require ASDL-dependent routing.

Table indexes:

```
mach_clli, asdl_cmd
```

tbl_comm_param

This static table contains communication parameters required to communicate with various external systems. It is possible to specify such communication parameters by device type, for example, serial, TCP/IP Telnet, Generic, and so on.

You can also specify communication parameters for:

- Host NEs and all devices.
- Host NEs and a particular device.
- Specific Host NE and all devices.
- Specific Host NE and a specific device.

If an entry exists for the Host NE and device of a particular Command processor in the NEP, it overrides any of the previous entries. You can also specify parameters in this table for particular Host NEs and devices or combinations.

Each parameter is made available to the Java method. tbl_comm_param specifies various parameters that are specific to the Host NE or device. It allows Java methods to employ processing that is specific to the Host NE or device, where required. You are responsible for populating and maintaining this table.

- **SSP_new_comm_param** adds new communication parameters for a specified device type, host, and device into tbl_comm_param.
- **SSP_del_comm_param** deletes parameter information from tbl_comm_param.
- **SSP_list_comm_param** lists the contents of this table.
- **SSP_get_async_ne** lists NEs which have an ASYNC_CONN communication parameter defined with a value of TRUE or FALSE.

Table 2-27 tbl_comm_param Columns

Column_name	Type	Length	Nulls	Description
dev_type	char	1	0	The device line type. The possible values include: • D – Serial Port Dialup • F – TCP/IP FTP Connection • G – Generic Terminal Based Connection • H – Serial Port Hardwired • M – Generic Message Based Connection • S – TCP/IP Socket Connection • T – TCP/IP Telnet Connection • W – LDAP Connection
host	varchar2	80	0	The Host NE for which this parameter value applies. If specifying communication parameters for all Host NEs, set this parameter to COMMON_HOST (COMMON_HOST_CFG).
device	varchar2	40	0	The device for which this parameter value applies. If specifying communication parameters for all devices, set this parameter to COMMON_DEVICE (COMMON_DEVICE_CFG).

Table 2-27 (Cont.) tbl_comm_param Columns

Column_name	Type	Length	Nulls	Description
param_label	varchar2	80	0	The parameter label. There are many communication parameters that you must specify in this table in order for the ASAP communication library to function correctly. These communication parameters and their usage are detailed in the *ASAP System Administrator's Guide*.
param_value	varchar2	255	0	The parameter value.
param_desc	varchar2	255	1	Description of the communication parameter.

Table indexes:

```
dev_type, host, device, param_label
```

tbl_cp_mux

Specifies the mapping between command processor logical devices and NEP multiplexing devices.

Table 2-28 tbl_cp_mux Columns

Column_name	Type	Length	Nulls	Description
cp_device	varchar2	40	0	A logical device used in the communications between a command processor thread and a multiplexing device. This device must be first configured in the table tbl_resource_pool.
mux	varchar2	40	0	The name of the multiplexing device that the specified command processor logical device maps to. The multiplexing device must be first configured in the table tbl_nep_mux.

Table indexes:

```
cp_device
```

tbl_csdl_asdl

This static table used by the SARM to map and sequence CSDL commands to ASDL commands. For each ASDL associated with a CSDL, SARM checks if the ASDL is valid for the CSDL. The final determination of whether the ASDL is valid depends on the ASDL parameter translation process specified by tbl_asdl_parm. To perform this translation, certain conditions must apply. These conditions are identified in columns cond_flag and eval_exp. An ASDL is valid only if both conditions are satisfied.

Examples of various configurations and instructions on configuring CSDL-to-ASDL translation are described in the *ASAP System Administrator's Guide*.

You are responsible for populating and maintaining this table using one of the following mechanisms:

- using XML schemas and deploying these services using the Service Activation Deployment Tool (SADT). Refer to the *ASAP Cartridge Development Guide* for more information.
- using stored procedures
  - **SSP_new_CSDL_asdl** adds a new CSDL-to-ASDL mapping definitions to tbl_csdl_asdl
  - **SSP_del_CSDL_asdl** deletes CSDL-to-ASDL mapping definitions from tbl_csdl_asdl
  - **SSP_list_csdl_asdl** lists the contents of this table.

Table 2-29 tbl_csdl_asdl Columns

Column_name	Type	Length	Nulls	Description
csdl_cmd	varchar2	80	0	CSDL command that translates into one or more ASDL commands.
asdl_seq_no	number	38	0	The sequence number of the ASDL command associated with the CSDL. This enables one CSDL to generate more than one ASDL command.
asdl_cmd	varchar2	80	0	ASDL command to which the CSDL translates.
pnr	number	38	1	Values are: • 0 (default) – This ASDL is not the 'point of no return' for rollback purposes • 1 – This ASDL is the 'point of no return' for partial rollback. If rollback occurs, and the process has continued beyond this point, roll back to this ASDL but no further. • 2 – 'point of no return' for no rollback. Once past this ASDL, no rollback can occur.

Table indexes:

```
csdl_cmd, asdl_seq_no
```

tbl_csdl_asdl_eval

This table contains CSDL to ASDL mappings and the multiple condition expressions defined for the mappings

Table 2-30 tbl_csdl_asdl_eval Columns

Column_name	Type	Length	Nulls	Description
csdl_cmd	varchar2	80	0	Foreign key to tbl_csdl_asdl.csdl_cmd.
asdl_seq_no	varchar2	80	0	The sequence number of the ASDL command associated with the CSDL. This enables one CSDL to generate more than one ASDL command.

Table 2-30 (Cont.) tbl_csdl_asdl_eval Columns

Column_name	Type	Length	Nulls	Description
cond_flag	char	1	0	The conditions to be met for the ASDL to be run for the CSDL. The values for this field are as follows: • A – Always run the ASDL for this CSDL when the expression is NULL (no expression), when the expression contains the string TRUE, or when the expression is evaluated to be true. • E – Check for the associated label to be present for the CSDL parameter label/value pairs and check that its value is equal to the label value. Similar to "=" or LIKE in the algebraic expression. • D – Check that the label is defined. Similar to ISDEF operations in the algebraic expression. • N – Check that the label is not defined. Checks that the label is similar to NOTDEF. The condition is checked for each ASDL associated with the CSDL. If the condition is satisfied, the ASDL is added to the list of ASDLs for the CSDL. If not, the ASDL is not run. Validation is performed in the parameter validation stage in referencing the table tbl_asdl_parm.
label	varchar2	80	1	The parameter label to test in the condition flag.
value	varchar2	255	1	The parameter value associated with the label which is tested in the condition flag.
eval_exp	varchar2	255	1	Contains combination of parameter names, operators, and values to which the parameters are compared.
apply_from	number	38	1	The first indexed ASDL that this rule should apply to: Valid range is from 1 to 9999. Must be less than or equal to the value specified in column **apply_to.** If is not specified, then this rule will be applied to any indexed ASDL up to and including the one specified in column **apply_to**.
apply_to	number	38	1	The last indexed ASDL that this rule should apply to: Valid range is from 1 to 9999. Must be greater than or equal to the value specified in column **apply_from.** If is not specified, then this rule will be applied to any indexed ASDL starting from the one specified in column **apply_from**.

tbl_csdl_config

This static table contains the static CSDL configuration information that determines if the CSDL is configured for rollback, its provisioning sequence, and whether a failure of the CSDL generates a system event. You are responsible for populating and maintaining this table.

- **SSP_new_csdl_defn** adds new CSDL definitions to tbl_csdl_config.
- **SSP_del_csdl_defn** deletes CSDL definitions from tbl_csdl_config.
- **SSP_list_net_elem** lists the contents of this table.

Table 2-31 tbl_csdl_config Columns

Column_name	Type	Length	Nulls	Description
csdl_cmd	varchar2	80	0	The CSDL command.
rollback_req	char	1	0	A flag indicating if rollback is required for this CSDL. In certain circumstances, it may be preferable for no rollback to occur. If any of the CSDLs on a work order require rollback, the dynamic work order structure in SARM's memory is flagged and the entire work order is rolled back if any CSDL fails. Possible values are: • Y – Rollback required. • N – No rollback required. In the SRP database, **wo_rback** must be set to Y or D for the CSDL to rollback.
csdl_level	smallint	1	0	The level of the CSDL in the SRQ. An integer between 0 and 255 that indicates the sequence level for the CSDL command within the work order. The SARM uses this integer to determine the order in which to provision CSDL commands from an SRP. The SARM then provisions CSDL commands that have lower level numbers first. Sequence levels are only relevant for inter-dependent CSDL commands As the SARM receives the CSDLs from the SRP, it reorders them in a sequence that corresponds to their respective CSDL levels to ensure the correct provisioning sequence. This reordered sequence may not be the same as the sequence in which the SRP transmitted the CSDLs to the SARM. The field csdl_id in the table tbl_srq_csdl contains the SRP CSDL sequence of the CSDLs transmitted to the SARM.
fail_event	varchar2	8	1	If set, the system event to be triggered upon CSDL failure. This event can log or print error messages, trigger system alarms, etc. Use this column to immediately notify personnel of CSDL failure.
complete_event	varchar2	8	1	If set, the system event triggered upon CSDL completion.

Table 2-31 (Cont.) tbl_csdl_config Columns

Column_name	Type	Length	Nulls	Description
option_asdl	number	38	0	Reserved for future use.
description	varchar2	255	1	Description of the CSDL command. This description field is displayed in front-end user interfaces only.

Table indexes:

`csdl_cmd`

tbl_err_threshold

This static table specifies the threshold for the number of consecutive hard errors for a particular Host NE and ASDL. If the threshold is exceeded, the SARM requests the NEP to disable the specified Host NE. Such hard errors can be generated directly from user-defined exit types that map to hard errors. You are responsible for populating and maintaining this table.

- **SSP_new_err_threshold** adds a new threshold for a specific NE and ASDL command in tbl_err_threshold.
- **SSP_del_err_threshold** deletes a threshold from tbl_err_threshold.
- **SSP_list_err_threshold** lists the contents of this table.

Table 2-32 tbl_err_threshold Columns

Column_name	Type	Length	Nulls	Description
host_clli	varchar2	80	0	The Host NE.
asdl_cmd	varchar2	80	0	The ASDL commands to be used to track consecutive hard errors.
threshold	number	38	0	The threshold number of consecutive hard errors that must occur for the SARM to disable the specified Host NE.

Table indexes:

`host_clli, asdl_cmd`

tbl_event_dataset

This table contains sets of parameters to be returned for events. Note that multiple trigger criteria (stored in "tbl_event_template") can map to a single set of return parameters. Also, the table can contain multiple rows for given **template_name**, with each row defining a particular parameter.

Table 2-33 tbl_event_dataset Columns

Column_name	Type	Length	Nulls	Description
template_name	varchar2	20	0	The name of the event template dataset. This is a unique user-defined identifier for a distinct return parameter dataset.
parameter_type	varchar2	20	0	Identifies the type of the parameter to be added to the event template dataset: • GLOBAL_PARAMETER • CSDL_PARAMETER • INFO_PARAM • EXTENDED_PROPERTY
csdl	varchar2	80	1	The CSDL to apply the event template dataset to. The csdl parameter is only required if the **parameter_type** is CSDL_PARAMETER or INFO_PARAM:
parameter_name	varchar2	20	0	Name of the parameter to be added to the event template dataset.

Table indexes:

`template_name, parameter_type, cscl, parameter_name`

tbl_event_template

This table maps event template names to event type / CSDL combinations. The **name** field maps to event template entries in tbl_event_dataset which detail extended event information to be returned in the work order header.

See "tbl_event_dataset" for more information.

> **Note:**
>
> The combination of **event_type**, **csdl**, **parameter_name**, and **parameter_value** within the table **tbl_event_template** must be unique to return the parameters appropriately. If the data matches, SARM considers those templates to be duplicates and returns the parameters of only one template.

Table 2-34 tbl_event_template Columns

Column_name	Type	Length	Nulls	Description
name	varchar2	20	0	The name of the event template. Points to a set of return parameters in tbl_event_dataset thus allowing for the case where multiple trigger criteria could map to a single set of return parameters.

Table 2-34 (Cont.) tbl_event_template Columns

Column_name	Type	Length	Nulls	Description
event_type	varchar2	20	0	(Mandatory) The event type to apply the event template name to. Following are the event types: • Order Startup event: Returns order parameters, extended work order property. • Order Complete event: Returns order parameters, information parameters, extended work order parameters, and service action parameters. • Order Timeout event: Returns order parameters, information parameters, extended work order parameters, and service action parameters. • Order Fail event: Returns order parameters, information parameters, extended work order parameters, and service action parameters. **Note:** For each service action or just for the given service action, service action parameters are returned.
csdl	varchar2	80	1	The CSDL to apply the event template to. Optional. **Note**: This field is applicable only when the event type is set to **orderFailEvent**.

Table indexes:

`event_type, csdl`

tbl_ext_method_lib

This table provides the external method library details required by the SRT.

Table 2-35 tbl_ext_method_lib Columns

Column_name	Type	Length	Nulls	Description
name	varchar2	80	0	Name of the library
type	char	1	0	Type of the library • S - represents Script • Q - represents SQL • J - represents Jar
library	blob	-	1	The binary form of the library.

tbl_host_clli

This static table contains the Host NE, technology, and software load of each Host NE in the ASAP system. It also contains records for each Host NE to which the NEPs interface. You are responsible for populating and maintaining this table

- **SSP_new_net_host** adds new network element definitions to this table.

- **SSP_del_net_host** deletes network element definitions from this table.
- **SSP_list_net_host** lists the contents of this table.

Table 2-36 tbl_host_clli Columns

Column_name	Type	Length	Nulls	Description
host_clli	varchar2	80	0	The Host NE identifier.
tech_type	varchar2	16	0	Technology of the Host NE or SRP.
sftwr_load	varchar2	16	0	Software version of the Host NE, for example, BCS33, PAC4, or the token software load of the SRP. The technology and software load must be consistent with the values given in the table tbl_nep_asdl_prog in order to ensure that JInterpreter are able to translate an ASDL.

Table indexes:

`host_clli`

tbl_id_routing

This static table is a routing database table that defines the mapping between an ASDL and the Host NE when using ID_ROUTING as an ASDL routing scheme.

- **SSP_new_id_routing** adds a host NE and the ID_ROUTING mapping record to tbl_id_routing.
- **SSP_del_id_routing** deletes a host NE and the ID_ROUTING mapping record from tbl_id_routing.
- **SSP_list_id_routing** lists the contents of this table.

Table 2-37 tbl_id_routing Columns

Column_name	Type	Length	Nulls	Description
host_clli	varchar2	80	0	The Host NE which the ASDL command is routed to.
asdl_cmd	varchar2	80	1	The ASDL command that is being routed.
id_routing_from	varchar2	255	0	The starting point of a range of ID_ROUTING.
id_routing_to	varchar2	255	0	The end point of a range of ID_ROUTING.

If a CSDL maps to multiple ASDLs that are routed to different host NEs, you must ensure that tbl_ne_routing contains an entry for each ASDL. Each entry references a different host NE. For example:

Table 2-38 ID Routing example

fromRange	toRange	asdl_cmd	host_clli
100	200	A-ADD_POTS_LINE	HOUSTON
100	200	A-OPTION_ON	DALLAS

The following example shows how ID_ROUTING operates for IP addresses:

Table 2-39 ID Routing example

fromRange	toRange	asdl_cmd	host_clli
10.9.1.0	10.9.10.255	A-ADD_POTS_LINE	TORONTO
10.9.11.0	10.9.18.255	A-ADD_POTS_LINE	DALLAS

In this case, if the ASDL references an IP address of 10.9.3.25, the ASDL would be routed to the TORONTO host NE. The IP address 10.9.12.255 would be routed to DALLAS.

Refer to $ASAP_BASE\samples\ASDL_ROUTE for a sample ID routing based on IP addresses.

Table index:

```
host_clli
```

Once the SARM starts, tbl_id_routing is loaded into memory from the SARM database.

For exact matching, the value of id_routing_from, and id_routing_to must be configured identically.

tbl_info_parm

This dynamic table contains information parameters that are returned to the SRP from NEP. These parameters contain information that is returned to the requesting external system as Compound parameters. The SRP uses a query API function to retrieve the Information parameters associated with a particular work order.

Table 2-40 tbl_info_parm Columns

Column_name	Type	Length	Nulls	Description
wo_id	varchar2	80	0	The work order ID.
parm_lbl	varchar2	80	0	The parameter label.
parm_group	varchar2	80	1	NE parameter group information returned to the Host system. The SRP API routines specify parameters belonging to particular parameter groups in the Information parameter retrieval.
parm_vlu	varchar2	255	1	The value associated with the parameter label. All control characters x0 to x1F, except x9, xA, and xD (tab, cr and nl), are stripped from the event text after it is retrieved from the database and before used in the XML document. This behavior is in compliance with XML 1.0 specification.
csdl_seq_no	number	38	0	The CSDL sequence number in the work order that is being processed when the Information parameter is generated.
csdl_cmd	varchar2	80	0	The CSDL command being processed when the Information parameter is generated from the Interpreter.

Table 2-40 (Cont.) tbl_info_parm Columns

Column_name	Type	Length	Nulls	Description
csdl_id	number	38	0	The ID of the CSDL being processed when the Information parameter is generated. This CSDL ID is the same as in table tbl_srq_csdl and enables the SRP to track which CSDL the information parameter belongs to.
vlu_hint	char	1	0	Indicates whether the parameter value is a real value or an XML document or XPath expression. Can be one of the following: • X - indicates a reference to XML document • P - indicates a reference to XPath expression • T - indicates a reference to large text (reserved for future use)

Table indexes:

```
wo_id, parm_lbl, parm_group
```

tbl_label_value

This dynamic table stores event subscription information from the old CORBA SRP clients to handle event servers between CORBA SRP client and server. The new OCA SRP does not use this table.

Table 2-41 tbl_label_value Columns

Column_name	Type	Length	Nulls	Description
key_1	varchar2	16	0	Work Order ID.
key_2	number	-	1	Event type. Possible values are defined oca.idls as follows: • (1) ASC_WO_COMPLETE_EVT • (2) ASC_WO_FAILURE_EVT • (3) ASC_WO_COMPLETE_FAILURE_EVT • (4) ASC_WO_OTHER_EVT • (5) ASC_WO_ALL_EVT • (6) ASC_WO_SOFT_ERR_EVT • (8) ASC_WO_TIMEOUT_EVT • (9) ASC_WO_ROLLBACK_EVT • (10) ASC_WO_ACCEPT_EVT • (11) ASC_WO_STARTUP_EVT • (12) ASC_WO_ESTIMATE_EVT • (13) ASC_NE_UNKNOWN_EVT
label	varchar2	600	0	An external system ID that is passed with the Work Order. It is not used by the SARM, but it is used by the SRP for proper routing to upstream systems.

Table 2-41 (Cont.) tbl_label_value Columns

Column_name	Type	Length	Nulls	Description
value	varchar2	128	1	A sister external system ID that is passed with the Work Order. It is not used by the SARM, but it is used by the SRP for proper routing to upstream systems.

Table indexes:

```
key_1, key_2, label
```

tbl_large_data

This table stores XML data, XPath expressions or other large data for future use by the SRT.

Table 2-42 tbl_large_data Columns

Column_name	Type	Length	Nulls	Description
ref_id	number	20	0	Reference ID returned by the Database for the XML Order Data.
data	blob	-	1	Raw XML Data, XPath Expression or other large data.

tbl_msg_convert

This static table contains the language and format of events logged by the SARM as it processes requests. Log messages are sent back to the Host system, and therefore, it is important they are in the user's native language. This table provides you with a mechanism to insert your own event text in your own language by populating this table with suitable data.

The core system provides the base event text in the language USA English. If you want to use another language, you are responsible for populating this table with the events in that language.

- **SSP_new_intl_msg** adds a new international message record to tbl_msg_convert.
- **SSP_del_intl_msg** deletes an international message from tbl_msg_convert.
- **SSP_list_intl_msg** lists the contents of this table.

Table 2-43 tbl_msg_convert Columns

Column_name	Type	Length	Nulls	Description
lang_cd	varchar2	3	0	The language code. This code determines the text message and format the SARM utilizes. The language code that the SARM uses is set by the SARM configuration variable LANGUAGE_OF_MSG. The default is USA.
msg_id	number	38	0	Unique message identifier for messages referenced within the SARM.

Table 2-43 (Cont.) tbl_msg_convert Columns

Column_name	Type	Length	Nulls	Description
msg_type	char	1	0	This field specifies the type of message formatting. Possible values are: • (D) Dynamic – The SARM performs parameter substitution into the formatted string in the message field which allows dynamic customizing of the message text. • (S) Static – The SARM does not perform any parameter substitution; instead it uses the raw message in the next message field.
message	char	255	0	The message text. If the type field is S, this field resembles a printf() format string. If the type field is D, no parameter substitution is performed.
var_description	char	40	1	Description of the substitutable fields within the message.
wo_audit	char	1	1	The destination for this log message. Possible values are: • ASAP_LOG_SRQ (srq_log) • ASAP_LOG_WOA (work order audit) • ASAP_SRQWOA (both) • ASAP_LOG_NO (none)

Table indexes:

```
lang_cd, msg_id
```

tbl_ne_config

This static table, which is used by the SARM and the NEP, contains configuration information for each Host NE within ASAP. It also identifies the NEP that manages each Host NE.

This table is read by the SARM to determine the NEs managed by each NEP. This allows the SARM to route the ASDLs to the appropriate NEP that is managing the Host NE. It defines attributes that are particular to each Host NE regardless of the number and nature of the connections to that NE.

This table is read by the NEP to determine the NEs to be managed by that NEP and the Session Managers to be spawned to control all interaction with the NEs. It also specifies the primary resource pools of devices used to connect to the NEs.You are responsible for populating and maintaining this table.

- **SSP_new_net_elem()** adds new network element definitions to this table.
- **SSP_del_net_elem()** deletes network element definitions from this table.
- **SSP_list_net_elem()** lists the contents of this table.
- **SSP_set_ne_loopback()** updates the table when the loopback state is set to ON, OFF, or GLOBAL through asap_utils.
- **SSP_new_net_element()** creates an extra throughput column with the default value of 0.
- **SSP_set_ne_throughput()** sets the minimum time for a transaction in milliseconds on the NE.

Table 2-44 tbl_ne_config Columns

Column_name	Type	Length	Nulls	Description
host_clli	varchar2	80	0	The Host NE. This is the name of the SessionManager thread within the NEP which manages all interaction with the Host NE and all connections to it. For dynamic routing, this is the name of the template.
nep_svr_cd	varchar2	8	0	The logical name of the NEP server that connects to the Host NE.
primary_pool	varchar2	8	1	The primary resource pool used by the NEP managing the Host NE. It determines the primary devices used to connect to the Host NE. The primary pool is dedicated to a specific NE. The Host NE Session Manager within the NEP uses entries in this primary pool before attempting to use entries in the NEP's secondary resource pool to connect to this Host NE, as defined in tbl_nep.
max_connections	number	2	0	The maximum number of concurrent connections allowed to the Host NE. This includes connections from both primary and secondary pools. For multiple dedicated Command Processors to a Host NE, configure the spawn threshold to 1 and the terminate threshold to 0. This ensures that multiple Command Processors are used. For dynamic routing, this information can be provided on each order.

Table 2-44 (Cont.) tbl_ne_config Columns

Column_name	Type	Length	Nulls	Description
drop_timeout	number	2	0	The maximum provisioning activity idle time, in minutes, that must elapse before the Session Manager managing the Host NE disconnects the primary connection to the NE. Until this threshold is exceeded, the Session Manager managing the NE maintains a dedicated Command Processor with a dedicated connection to the NE when there are no pending ASDL requests (maximum idle time). This time-out interval only applies to the primary connection to the NE, as all auxiliary connections have reached their kill_threshold by this point. For a busy Host NE, this time interval should be set high as there is a high probability of an incoming ASDL for the Host NE in that period. However, this value should be set low for a less active Host NE as the resource pool device in use by the Command Processor may be required by another Session Manager to communicate with another Host NE. This is most likely the case when the device is from the NEP's resource pool (the secondary pool which is also accessible to all Session Managers). For dynamic routing, drop timeout information can be provided on each order.
throughput	smallint	1	0	NE instance throughput control – the minimum number of milliseconds a transaction takes per NE instance. The value 0 disables NE instance thoughput control.
spawn_threshold	number	2	0	The number of ASDL requests in the SARM's “ASDL Ready Queue" destined to the NE. When this number is exceeded, the SARM requests the NEP Session Manager to open a new auxiliary connection to the NE. As the number of ASDLs in the SARM ASDL Ready Queue continues to exceed this spawn threshold, the SARM continues to request the NEP establishment of auxiliary connections to the NE. For example, the spawn threshold to a particular NE is 10. Once the ASDL Ready Queue size reaches 11, the SARM requests the NEP to establish an auxiliary connection to the NE. If the ASDL Ready Queue size reaches 12, the SARM requests another auxiliary connection be established by the NEP, and so on. This spawn threshold value should always be above the terminate threshold if multiple connections are required to a particular NE. For dynamic routing, spawn threshold information can be provided on each order.

Table 2-44 (Cont.) tbl_ne_config Columns

Column_name	Type	Length	Nulls	Description
kill_threshold	number	2	0	Upon receiving an ASDL completion, the SARM determines whether the current ASDL Ready queue size is less than the terminate threshold. If so, it checks whether the ASDL was completed by an auxiliary connection. If so, the SARM transmits a disconnect request to the NEP Session Manager which disconnects the auxiliary connection. If the ASDL completion was performed on the primary connection, the SARM will not issue a disconnect request. For example, if the spawn_threshold is 10 and the terminate threshold is 5, when the SARM receives an ASDL completion when its ASDL Ready queue size is 4 or less, it checks whether the ASDL completion was performed by an auxiliary connection. If so, it issues a disconnect request to the NEP Session Manager managing this NE. If the ASDL completion was not performed by an auxiliary connection, as would be the case when the spawn threshold was never exceeded, the SARM will not issue a disconnect request. For multiple dedicated connections, set the relevant number of connections in max_connections, the spawn threshold to 1, the terminate threshold to 0, and the appropriate number of devices in the primary and secondary resource pools. For dynamic routing, terminate threshold information can be provided on each order.
template_flag	Char ('Y','N')	1	0	Flag to indicate if this network element entry identifies a static NE (N) or a dynamic network element template (Y). The default value is 'N'.
loopback_on	Char ('Y', 'N', 'G')	1	0	Y: The NE is in loopback state regardless of the LOOPBACK_ON ASAP configuration parameter. N: The NE is not in loop back state regardless of the LOOPBACK_ON ASAP configuration parameter. G: The NE loop back state is determined by the LOOPBACK_ON ASAP configuration parameter. The default value is 'G'.
request_timeout	number	8	0	The maximum number of seconds for an NE request timeout. If there is no response in the given timeout value, the associated ASDL will timeout. For more information, see the *ASAP System Administrator's Guide*.
request_retry_number	number	8	0	The maximum number of retries, if the NE requests timeout. If the number of retries exceeds request_retry_number, the order is failed and rolled back. For more information, see the *ASAP System Administrator's Guide*.

Table 2-44 (Cont.) tbl_ne_config Columns

Column_name	Type	Length	Nulls	Description
request_retry_interval	number	8	0	The time period in seconds between NE retries. For more information, see the *ASAP System Administrator's Guide*.

Table indexes:

```
nep_svr_cd, host_clli
```

tbl_ne_event

This table contains NE-related log messages.

Table 2-45 tbl_ne_event Columns

Column_name	Type	Length	Nulls	Description
event_identity	number	20	0	Unique ID of this log message in this table.
host_clli	varchar2	80	0	The Host NE identifier if an NE or the SRP name if an SRP.
event_dts	date	-	0	The time when the event was generated.
state	varchar2	80	0	The current state of the NE. Possible states are: • Available • Down • Connecting • Maintenance • Port Failure • Disabled • Unknown NE State
evt_text	varchar2	255	0	Description of the event.

Table indexes:

```
Indexes

evt_identity
host_clli, event_dts
event_dts
```

tbl_ne_monitor

Maintains the status of all NEs.

> **Note:**
>
> The tbl_ne_monitor table has been deprecated from ASAP 4.6.x onwards due to impact on the performance and so NE state information is no longer populated in it by default. However, this functionality can be turned on when you set the variable `WO_TIME_ESTIMATE_ON` to **1** in the **ASAP.cfg** file. On turning this configuration variable on the SARM database table tbl_ne_monitor gets populated with the NE state information and a new event WO_ESTIMATE_EVT is added to the SARM database table tbl_wo_event_queue. Oracle recommends not to use this feature in production.

Table 2-46 tbl_ne_monitor Columns

Column_name	Type	Length	Nulls	Description
host_clli	varchar2	80	0	The Host NE where the ASDL is routed by the SARM. Even though one Host NE is specified on the CSDL, different ASDLs on that CSDL can be routed to different Host NEs.
nep_svr_cd	varchar2	8	0	Name of the NEP server.
state	varchar2	80	0	The current state of the NE. Possible states are: • Available • Down • Connecting • Maintenance • Port Failure • Disabled • Unknown NE State
asdl_time_est	number	38	0	Estimated time to process an ASDL.
pending_count	number	38	0	Number of ASDLs pending.
in_progress_count	number	38	0	Number of ASDLs in progress.
connect_count	number	38	0	Number of connections to the NE.
retry_count	number	38	0	The number of times the ASDL was retried at the NE.
held_count	number	38	0	Number of ASDLs held.
consec_fail_count	number	38	0	Number of consecutive failures.
connect_dts	date	-	0	Date/time stamp of the last connection.
disconnect_dts	date	-	0	Date/time stamp of the last disconnect.
avail_dts	date	-	0	Date/time stamp of the last time mode available.
maint_dts	date	-	0	Date/time stamp of the last time it was put into maintenance mode.
err_disable_dts	date	-	0	Date/time stamp of the last time an error disabled the NE.
err_enable_dts	date	-	0	Date/time stamp of the last time the NE was enabled after an error.
adm_disable_dts	date	-	0	The last time an administrator disabled the NE.
adm_enable_dts	date	-	0	The last time an administrator enabled the NE.

Table 2-46 (Cont.) tbl_ne_monitor Columns

Column_name	Type	Length	Nulls	Description
swd_sessions	number	38	1	Number of switch direct sessions.
swd_start_dts	date	-	1	The last time a session started.
swd_end_dts	date	-	1	The time when a switch direct session ended.
swd_user_id	varchar2	64	1	Switch direct user.
pad1	char	255	0	Padding to make a table row occupy a page. This reduces concurrence on the database data page by different database processes.
pad2	char	255	0	Same as pad1.
pad3	char	255	0	Same as pad1.
pad4	char	255	0	Same as pad1.

Table indexes:

`host_clli`

tbl_ne_strsub

Enables you to configure substitutions for unknown or unwanted control characters.

Table 2-47 tbl_ne_strsub Columns

Column_name	Type	Length	Nulls	Description
strsub_type	varchar2	20	0	TBD
description	varchar2	255	1	TBD
asdl	varchar2	80	1	Name of the ASDL associated with the character. If no value is provided, the substitution will be made for all ASDL commands.
csdl	varchar2	80	1	Name of the CSDL associated with the character. If no value is provided, the substitution will be made for all CSDL commands.
ne_vendor	varchar2	255	1	Name of the software vendor for the network element associated with the character. If no value is provided, the substitution will be made for all network element vendors.
tech_type	varchar2	255	1	Technology type for the network element associated with the character. If no value is provided, the substitution will be made for all technology types.
sftwr_load	varchar2	16	1	Software load for the network element associated with the character. If no value is provided, the substitution will be made for all software loads.
ne_str_pattern	varchar2	255	1	TBD
ne_replace_pattern	varchar2	255	1	TBD

Table indexes:

```
strsub_type, csdl, asdl, ne_vendor, tech_type, sftwr_load, ne_str_pattern,
ne_replace_pattern
```

tbl_nep

This static table, referenced by the SARM and NEP, maintains the relationship between the NEP and the secondary pool of devices which is used by the NEP to establish auxiliary connections to Host NEs.

The SARM references this table upon start up to determine the NEPs configured within the system. For each NEP, the SARM opens one or more network connections to that NEP. You can configure the number of connections.

Each NEP references this table upon start up to determine the secondary pool of devices available to all Session Managers within that NEP. It spawns a Command Processor thread for each device in this secondary pool of devices. You are responsible for populating and maintaining this table.

- **SSP_new_nep** adds a new pool of devices to this table.
- **SSP_del_nep** deletes a pool of devices from this table.
- **SSP_list_nep** lists the contents of this table.

Table 2-48 tbl_nep Columns

Column_name	Type	Length	Nulls	Description
nep_svr_cd	varchar2	8	0	The logical name of the NEP managing the secondary pool of devices. It is not the physical environment specific name listed in the interfaces file.
dialup_pool	varchar2	8	1	Secondary pool of devices available to all Session Managers within the NEP. If specified, you must define this resource pool in tbl_resource_pool. If null, there are no secondary devices available in the NEP.

Table indexes:

```
nep_svr_cd
```

tbl_nep_asdl_prog

This static table is used by the JInterpreter within an application server to determine Java class to invoke for:

- a given technology such as DMS, AXE, S12, and so on.
- software load such as BCS33, PAC4, and so on.
- ASDL

It is referenced by any process using the Interpreter library. It is also referenced by NEPs, SRPs, and ISPs within ASAP.

This is useful when sftwr_load and tech have no meaning.

Any invocations of the CHAIN or CALL Interpreter actions require that the ASDL being chained or called has a mapping specified in this table. You are responsible for populating and maintaining this table.

- **SSP_new_asdl_map** adds a new ASDL mappings to tbl_nep_asdl_prog.
- **SSP_del_asdl_map** deletes ASDL mappings from tbl_nep_asdl_prog.
- **SSP_list_asdl_map** lists the contents of this table.

Table 2-49 tbl_nep_asdl_prog Columns

Column_name	Type	Length	Nulls	Description
tech	varchar2	16	0	The technology or type of the NE with which the Interpreter interacts, for example, DMS, S12, etc.
sftwr_load	varchar2	16	0	The Software Load of the software currently running on the NE. Since this field is non-null, you must place a value in the field even if there is no defined software load for this NE type. The table tbl_host_clli also adheres to the same convention for the software load value.
asdl_cmd	varchar2	80	0	The ASDL command passed to the Interpreter that determines the Interpreter program to be passed to the JInterpreter that determines the Java program to be run. ASAP reserves some special ASDL commands to map special requirements, such as LOGIN or CONNECTION_HANDLER, to a program.
program	varchar2	255	0	Name of the Java connection class or Java provisioning method for JInterpreter to run.
interpreter_type	varchar2 ['J']	1	0	A value of "J" indicates a JInterpreter.

Table indexes:

```
tech, sftwr_load, asdl_cmd
```

tbl_nep_mux

Stores the configuration of the NEP multiplexing devices.

Table 2-50 tbl_nep_mux Columns

Column_name	Type	Length	Nulls	Description
mux	varchar2	40	0	The name of the multiplexing device. Communication parameters for this device are specified in tbl_comm_param.
host_clli	varchar2	80	0	The NE that the multiplexing device communicates with. You must first configure the NE in tbl_ne_config.
mux_dev_type	char	1	0	The type of communications protocol or interface used in the interactions between the multiplexing device type and the NE.

Table 2-50 (Cont.) tbl_nep_mux Columns

Column_name	Type	Length	Nulls	Description
cp_dev_type	char	1	0	The type of communications protocol used in the interactions between the multiplexing device type and the command processor threads.

Table indexes:

```
mux
```

tbl_nep_jprogram

Table 2-51 tbl_nep_jprogram Columns

Column_name	Type	Length	Nulls	Description
program	varchar	255	0	Name of the java class in the form package.class name.
jclass	blob			The binary form of the java class.

Table indexes:

```
primary key

program
```

tbl_nep_program

This static table is used by the Interpreter within an application server information required to perform specific processing. For example, in the NEP, interact using MML commands and responses with the appropriate NE.

Records are maintained in NPG flat files for easier editing and then compiled into their respective table records.

You are responsible for populating and maintaining the NPG files.

- **SSP_new_nep_program** adds actions based on the specified program name and/or line number to tbl_nep_program.
- **SSP_del_nep_program** deletes actions based on the specified program name and/or line number from tbl_nep_program.
- **SSP_list_nep_program** lists the contents of this table

Table 2-52 tbl_nep_program Columns

Column_name	Type	Length	Nulls	Description
program	varchar2	255	0	The name of the Program. With this program the ASDL, Technology, and Software load map in the tbl_nep_asdl_prog static database table.
line_no	number	38	0	The line number in the program which acts as a label for that instruction. The line number is similar to the ones used in BASIC programs.

Table 2-52 (Cont.) tbl_nep_program Columns

Column_name	Type	Length	Nulls	Description
action	varchar2	32	0	An action string to identify the particular action performed by the Interpreter in the Command Processor. A number of actions are provided by various ASAP component libraries, for example, Interpreter, NEP, SRP, and so on. You can add action functions and overwrite existing action functions as required.
act_string	varchar2	255	1	The action string associated with the Java methods. This field is required depending on the specified action.
act_int	number	38	1	The action integer which, if set, represents the next line number in the program. The next line number is where the program should continue or there is a numeric field specific to the particular action function. If the action integer points to an invalid line number or is absent from the program, the Interpreter registers a run time error and fails the operation. This field is required depending on the specified action.

Table indexes:

`program, line_no`

tbl_nep_program_source

Stores the source code. The compiled version is saved in tbl_nep_program.

Table 2-53 tbl_nep_program_source Columns

Column_name	Type	Length	Nulls	Description
program	varchar2	255	0	The name of the program. With this program the ASDL, Technology, and Software load, map in the tbl_nep_asdl_prog static database table.
line_no	number	38	0	The line number in the program which acts as a label for that instruction. The line number is similar to the ones used in BASIC programs.
seq	number	38	0	Order of the source code.
source	varchar2	255	1	Line of the source code.

Table indexes:

`program, line_no, seq`

tbl_nep_rte_asdl_nxx

ASAP can receive work orders with no Remote NE information. This static table was created to route the ASDLs to the relevant Host NE by means of a DN and ASDL command. If the Host

system is unable to determine the Remote NE, the routing logic involving this table is employed in the SARM.

This table is not read into the internal memory within the SARM because such routing is rarely used in comparison to the Remote to Host NE routing mechanism. A parameter with label MCLI and value, and SKIPCLLI must be present on the ASDL for this routing logic to be employed. You are responsible for populating and maintaining this table

SSP_new_dn_map adds new ASDL command routings by directory number to tbl_nep_rte_asdl_nxx.

SSP_del_dn_map deletes ASDL command routings from tbl_nep_rte_asdl_nxx.

SSP_list_dn_map lists the contents of this table

Table 2-54 tbl_nep_rte_asdl_nxx Columns

Column_name	Type	Length	Nulls	Description
asdl_cmd	varchar2	80	0	The ASDL command that provides ASDL specific routing capabilities by telephone number. This is important for such services as Voice Mail which is routed to separate NEs based on the ASDL.
npa	varchar2	3	0	The NPA is the first three digits in a telephone number.
nxx	varchar2	3	0	The NXX is the second three digits in a telephone number.
from_line	varchar2	4	0	The starting point of a range of telephone LINE numbers. It is the remaining four numbers in a telephone number which provide routing.
to_line	varchar2	4	0	The end point of a range of telephone LINE numbers to provide routing.
cont_typ	varchar2	1	0	Reserved for future use.
cont_nm	varchar2	8	0	Reserved for future use.
queue_nm	varchar2	64	0	The Host NE to which this ASDL is routed. The SARM determines the NEP managing this Host NE and routes the ASDL appropriately.

Table indexes:

```
asdl_cmd, npa, nxx, from_line, to_line, cont_typ
```

tbl_order_events

This table provides the order translation details for the SRT.

Table 2-55 tbl_order_events Columns

Column_name	Type	Length	Nulls	Description
translation_name	varchar2	255	0	Name of the Translation Object.
query_type	varchar2	80	0	Type of the Query.
type	char	1	0	Type of the object. • J - JMS • X - XPATH

Table 2-55 (Cont.) tbl_order_events Columns

Column_name	Type	Length	Nulls	Description
parm	varchar2	255	1	Name of the parameter.
value	varchar2	255	1	Value for the parameter.

tbl_order_translation

This table contains the order translation script details for the SRT.

Table 2-56 tbl_order_translation Columns

Column_name	Type	Length	Nulls	Description
name	varchar2	255	0	A unique name given to the translation script as an identifier.
script_name	varchar2	255	1	Name of the script file that implements translation.
translation_type	varchar2	255	0	Type of Translation to be run. Possible values are XSLT and DO_NOT_FORWARD.
type	char	1	0	Type of the translation object
parm	varchar2	255	1	Name of the parameter
value	varchar2	255	1	Value of the parameter
message_direction	varchar2	80	1	Translation Direction which indicates the whether this translation is for incoming orders, responses or events. Possible values are UPSTREAM, EVENT and RESPONSE.
script	BLOB	-	1	Binary form of the library/script.

tbl_resource_pool

This static table defines collections of devices which may be used by the NEP to establish connections to NEs. Such groups of devices are called resource pools.

Each NE configuration (tbl_ne_config record) determines a primary resource pool which defines one or more devices the NEP uses to connect to that NE. Such devices are not used to connect to other NEs.

Each NEP has a secondary resource pool (defined in tbl_nep) containing devices used by the NEP to establish connections to any NE managed by that NEP. Such primary and secondary resource pools are defined in this table. You are responsible for populating and maintaining this table.

- **SSP_new_resource** adds new device definitions to this table.
- **SSP_del_resource** deletes device definitions from this table.
- **SSP_list_resource** lists the contents of this table

Table 2-57 tbl_resource_pool Columns

Column_name	Type	Length	Nulls	Description
asap_sys	varchar2	8	0	This is the environment in which this database record is to be used. As the ASAP databases are environment independent, only logical representations of physical entities exist within them. • tbl_resource_pool – The table that contains specific environment-dependent information that varies between environments. • asap_sys – Distinguishes different environments. It takes the value of the environment variable, ASAP_SYS, TEST, PROD, etc., in the current environment. At run time, only those records with this field defined to be the same as the environment variable asap_sys, are loaded by the NEP.
pool	varchar2	8	0	The name of the pool of devices. It is referenced by: • tbl_nep – The secondary resource pool of the entire NEP. • tbl_ne_config – The primary resource pool of the NE.
device	varchar2	40	0	The device name. This is the name of the logical device used to establish a connection to an NE. The device corresponding to this logical device is specified by means of the communications parameter table, tbl_comm_param.
line_type	char	1	0	The communication protocol used by this device. The possible values include: • C – CORBA • D – Serial Port Dialup • F – TCP/IP FTP Connection • G – Generic Terminal Based Connection • H – Serial Port Hardwired • M – Generic Message Based Connection • S – TCP/IP Socket Connection • T – TCP/IP Telnet Connection • W – LDAP Connection This is enforced by the associated data rule on the datatype. For the X.25 protocol, you can use both G and M. For the X.29, you can only use G because the X.29 does not support message-based type. Such definitions are defined in the header file nep_core.h.
vs_key	number	38	-	Reserved. The shared memory segment identifier for the Virtual Screen buffer.

Table indexes:

```
Indexes
```

```
asap_sys, pool
asap_sys, device
```

tbl_srq

This dynamic table is used by the SARM and contains Service Requests (SRQs) created by the SARM from details passed by an SRP during work order translation. There is a one-to-one mapping between work orders and SRQs in the SARM. The SRQ provides a unique ID that references to the work order within ASAP.

Table 2-58 tbl_srq Columns

Column_name	Type	Length	Nulls	Description
srq_id	number	38	0	SRQ unique ID.
srq_dd	date	-	0	Due date and time of this SRQ. This is the same as the sched_dts in the work order table.
grp_cd	char	1	0	The action of the SRQ. Possible values include: • A – ADD: addition of service. • R – REMOVE: removal of service. • C – CHANGE: change/update of existing service. • Q – QUERY: query existing service. This action field is important to the order in which SRQs are processed within the same work order and between work orders. These values are defined in asap_core.h.
srq_pri	char	1	0	Priority of this SRQ as assigned by the Host order system. It is equal to the work order priority. Possible values include: • (1) ASAP_SRQ_HIGH_PRIO – High priority SRQ. • (5) ASAP_SRQ_NORMAL_PRIO – Normal priority SRQ. • (9) ASAP_SRQ_LOW_PRIO – Low priority SRQ. These values are defined in asap_core.h. This priority field is the first field in the internal composite ASAP SRQ priority within the ASAP core followed by the SRQ due date and the SRQ action such as Remove, Change, or Add.

Table 2-58 (Cont.) tbl_srq Columns

Column_name	Type	Length	Nulls	Description
srq_stat	number	38	0	SRQ status that is updated while the SRQ is being processed by the SARM. Possible values include: • (0) HELD – SRQ is held awaiting manual intervention. • (1) INITIAL – SRQ is yet to begin provisioning. • (2) SRQ_ABORTED – SRQ has been stopped due to order update or cancel. • (3) NEP_UNAVAIL – NEP routing for the SRQ's current ASDL is temporarily unavailable. • (7) IN_PROCESSING – SRQ in process. • (12) COMPLETED – SRQ successfully completed. It may have exceptions and/or revisions. • (13) FAILED – SRQ failed. See the SRQ log details to determine the cause of failure. • (14) TRANSLATION_ERROR – The SRP could not translate the SRQ correctly but transmitted it to the SARM. • (15) SRQ_ROLLBACK – SRQ has been rolled back. • (16) REVIEW – SRQ is in Reviewed status. • (60) CMD_RETRY – An ASDL on this SRQ is currently in the ASDL Retry queue. The SARM waits for a configured period or number of retries. These definitions are defined in sarm_defs.h.
evt_dt_tm	date	-	0	Last date and time that this SRQ record was updated.
wo_id	varchar2	80	0	The ID of the work order that this SRQ belongs to.
srq_chg	char	1	1	Indicates whether the SRQ has been altered since being received by ASAP. Possible values include: • N – No revisions on the SRQ. • Y – Revisions in ASAP not to be reflected back to the Host System. • C – Revisions in ASAP to be reflected back to the Host System.
proc_typ	char	1	1	Type of processing required for this SRQ. Possible values include: • (I) IMMEDIATE – Immediate requests to be provisioned as they are received. • (D) DELAYED – Delayed or batch requests due in the future. Batch requests have lower priority than immediate requests. Values are defined in sarm_defs.h.

Table 2-58 (Cont.) tbl_srq Columns

Column_name	Type	Length	Nulls	Description
cur_csdl_seq_no	number	38	1	The sequence number of the current CSDL being processed within this SRQ. This serves as a pointer into the CSDL table which enables the SARM to determine which CSDL it had been processing. This is used by the SARM upon restarting an In Progress work order.
cur_csdl_st	number	38	1	The status of the current CSDL being processed within this SRQ. This field and the sequence number are updated by the SARM as the SRQ is being processed. Possible values of this field are detailed in the csdl_st field of the tbl_srq_csdl table.
host_clli	varchar2	80	1	The Host NE associated with the latest ASDL on the SRQ. This is updated by the SARM to the value of the current ASDL Host NE value. There may be multiple Host NEs associated with a CSDL since each ASDL may be routed to a different Host NE.

Table indexes:

```
Indexes

srq_id
wo_id
```

tbl_srq_asdl_parm

This dynamic table contains parameter name value pairs associated with a Service Request (SRQ). It allows the ASDL and rollback parameters to be defined and, where possible, provides a pointer to their location in other database tables.

Table 2-59 tbl_srq_asdl_parm Columns

Column_name	Type	Length	Nulls	Description
srq_id	number	38	0	The SRQ ID that the parameter name value pairs are associated.
unid	number	38	0	If the ASDL type is ROLLBACK_TYPE, (according to tbl_asdl_log) then this field is set to the value of the ASDL unique ID (asdl_unid) in the ASDL log table (tbl_asdl_log). This allows the ASDL log table to determine the rollback parameters associated with a particular ASDL. These parameter types are defined in the header file sarm_defs.h.
parm_lbl	varchar2	80	0	The parameter label, for example, LEN, or DN.
parm_vlu	varchar2	255	1	The parameter value associated with the parameter label.

Table 2-59 (Cont.) tbl_srq_asdl_parm Columns

Column_name	Type	Length	Nulls	Description
vlu_hint	char	1	0	Indicates whether the parameter value is a real value or an XML document or XPath expression. Can be one of the following: • X - indicates a reference to XML document • P - indicates a reference to XPath expression • T - indicates a reference to large text (reserved for future use)

Table indexes:

```
srq_id, unid, parm_lbl
```

tbl_srq_csdl

This dynamic table contains the CSDLs for each Service Request (SRQ) listed in the SRQ table. Each SRQ can have multiple CSDLs.

Table 2-60 tbl_srq_csdl Columns

Column_name	Type	Length	Nulls	Description
srq_id	number	38	0	Unique ID of the SRQ that is associated with the CSDL.
csdl_seq_no	number	38	0	Sequence number that distinguishes CSDLs within the SRQ.
actn_noun_lbl	varchar2	80	0	CSDL command.
csdl_st	number	38	0	Status of the CSDL. Possible values include: • (100) HELD_STATE – The CSDL is Held awaiting manual release within ASAP. • (101) INITIAL_STATE – The CSDL is yet to begin processing. • (102) FAILED_CSDL – The CSDL has failed. • (103) ABORTED_CSDL – The CSDL has been stopped (usually by an OCA user). • (104) COMPLETED_CSDL – The CSDL has completed successfully. • (106) ROLLBACK_COMPLETED_CSDL – The CSDL has been successfully rolled back. CSDL status values are defined in sarm_defs.h.
asdl_seq_no	number	38	0	The sequence number of the ASDL currently being processed by the SARM on the CSDL. It is updated dynamically by the SARM as the ASDLs are processed. The SARM uses this field to determine the current ASDL to be provisioned upon restart.

Table 2-60 (Cont.) tbl_srq_csdl Columns

Column_name	Type	Length	Nulls	Description
index_parm_cnt	number	38	0	Current index value in Indexed ASDL parameters. This field is dynamically updated by the SARM as it provisions an ASDL with a set of Indexed parameters. When the SARM restarts an In Progress order, it uses the value in this field to determine the relevant set of Indexed ASDL parameters to transmit with this ASDL. Use the getParam method to retrieve the value of the given input parameter as passed down to the JInterpreter by the SARM. Use the getIntParam method to retrieve the value of the given input parameter as passed down to the JInterpreter by the SARM and casts this value to an integer. Refer to the *ASAP Online Reference* for more information.
csdl_id	number	38	0	The CSDL ID in the SRP transmitted to the SARM. When the SRP transmits a CSDL to the SARM, it passes this CSDL ID which SARM stores in the field. With this, the SARM generates the csdl_seq_no field value once it receives all CSDLs from the SRP and orders them according to their respective levels. This field is maintained by the SARM so that the SRP can query the SARM for CSDL specific information. It is used by the SRP to correlate the CSDL in the SRP with the CSDL in the SARM.
asdl_route	char	1	0	Specifies the routing of the current ASDL on the CSDL. The CSDL is mapped to one or more ASDLs by the table tbl_csdl_asdl. Once the SARM acknowledges the ASDL, it looks up the appropriate entry in the tbl_asdl_config to determine the routing for this ASDL. Possible values include: • (?) TO_BE_DETERMINED – The SARM has not yet determined the routing of the first ASDL on this CSDL. • (N) ROUTE_TO_NEP – ASDL routed to the NEP. This value is defined in the header file sarm_defs.h.
csdl_type	char	1	1	The type of CSDL.
orig_seq_no	number	38	1	The original sequence number of the CSDL on the SRQ.
estimate	number	38	1	The time estimate to provision the CSDL.
start_dts	date	-	1	The date and time that the CSDL started provisioning.

Table 2-60 (Cont.) tbl_srq_csdl Columns

Column_name	Type	Length	Nulls	Description
abort_dts	date	-	1	The date and time that the CSDL was stopped.
failure_dts	date	-	1	The date and time that the CSDL failed provisioning.
comp_dts	date	-	1	The date and time that the CSDL completed provisioning.
update_dts	date	-	1	The date and time that the CSDL was last updated.
update_uid	varchar2	64	1	The user who last updated the CSDL.
prov_sequence	number	38	1	The provisioning sequence of the CSDL on the SRQ. You can specify an alternate provisioning sequence to the one originally received from the originating system.

Table indexes:

```
srq_id, csdl_seq_no
```

tbl_srq_log

This dynamic table contains information logged for each SRQ. The log is a history of events that occurred on each SRQ, including Switch History information of the NE responses. If the Switch History is greater than 195 characters, the entry splits into two entries.

Through an API, the SARM and NEP write to this database table during the provisioning process. The SRP queries this table through an API to retrieve selected records for a particular work order.

You can view audit log information through the OCA Client. For more information, refer to the *ASAP OCA User Guide*.

Table 2-61 tbl_srq_log Columns

Column_name	Type	Length	Nulls	Description
srq_id	number	38	0	ID of the Service Request.
srq_log_identity	number	20	0	Unique ID of the log message in this table. This is an identity field automatically generated by the RDBMS upon insertion. This replaces the earlier field for performance reasons.
evt_dt_tm	date	-	0	Date and time of the logged event.
csdl_seq_no	number	38	0	Sequence number of the CSDL within the SRQ that the event is associated with. This is the same as the csdl_seq_no field in the CSDL table.
srq_stat	number	38	0	Status of the SRQ at the time the event occurred. For possible values, see srq_stat in tbl_srq, "tbl_srq."

Table 2-61 (Cont.) tbl_srq_log Columns

Column_name	Type	Length	Nulls	Description
csdl_st	number	38	0	Status of the CSDL when the event occurred. For possible values, see csdl_st in tbl_srq_csdl, "tbl_srq_csdl."
srq_evt	varchar2	8	0	The SRQ Log Event. The SRP may inquire for the SRQ log by specifying particular SRQ events of interest in the inquiry RPCs. Possible values include: • SRQ_INFO_EVENT "INFO" – Information messages. • SRQ_ERROR_EVENT "ERROR" – Error messages. • NE_CMD_EVENT "NE_CMD" – Command entered to the NE. • NE_RESP_EVENT "NE_RESP" – Response from the NE. These values are defined in the header file sarm_defs.h. You must set WO_AUDIT_LEVEL in ASAP.cfg to 2 to generate SRQ_ERROR_EVENTS.
evt_text	varchar2	255	1	Text description of the event. For NE responses, it contains 255 characters at a time (including newline characters) from the NE generated report. This should be considered when the information is being displayed by a front-end user interface. This allows each log record to contain more NE response information than if one NE response record was contained in each SRQ log record. If generated by an application, this text describes the event as displayed to you. All control characters x0 to x1F, except x9, xA, and xD (tab, cr and nl), are stripped from the event text after it is retrieved from the database and before used in the XML document. This behavior is in compliance with XML 1.0 specification.
asdl_unid	number	38	1	The ID of the ASDL that generated the log entry. If there is no current ASDL for the log entry, set this field to null.

Table indexes:

```
srq_id, srq_log_identity
```

tbl_srq_parm

This dynamic table contains global and CSDL name value pairs associated with an SRQ. It allows different types of parameters to be defined and where possible, provides a pointer to their location in other database tables. There is an index on this table which allows efficient query access to the global and CSDL parameters.

Table 2-62 tbl_srq_parm Columns

Column_name	Type	Length	Nulls	Description
srq_id	number	38	0	The SRQ ID that is associated with the parameter name value pairs.
parm_typ	char	1	0	The type of SRQ parameter. Possible values include: • (P) GLOBAL_TYPE – Parameters are set by one CSDL in an SRQ and referenced by others (they are global within the SRQ). • (C) CSDL_TYPE – Parameters are local to the current CSDL being processed within the SRQ. They are referenced by other ASDLs within the same CSDL. These parameter types are defined in the header file sarm_defs.h.
unid	number	38	0	If the parameter type is CSDL_TYPE, this field equals the CSDL sequence number (csdl_seq_no) in the CSDL tbl_srq_csdl. This field is not used if the parameter type is GLOBAL_TYPE.
parm_lbl	varchar2	80	0	The parameter label, for example, LEN or DN.
parm_vlu	varchar2	255	1	The parameter value associated with the parameter label.
parm_subvlu	varchar2	80	1	Contains the first 32 bytes of the global or CSDL parameter value.
vlu_hint	char	1	0	Indicates whether the parameter value is a real value or an XML document or XPath expression. Can be one of the following: • X - indicates a reference to XML document • P - indicates a reference to XPath expression • T - indicates a reference to large text (reserved for future use)

Table indexes:

```
Indexes

srq_id, parm_typ, unid, parm_lbl
parm_subvlu, parm_lbl
```

tbl_srt_bundle

This table contains SRT service bundles and/or its spawning details for the SRT.

Table 2-63 tbl_srt_bundle Columns

Column_name	Type	Length	Nulls	Description
service_id	varchar2	80	0	Service Identifier.
description	varchar2	1024	1	Description of the bundle or service action.

Table 2-63 (Cont.) tbl_srt_bundle Columns

Column_name	Type	Length	Nulls	Description
spawn_parm	varchar2	255	1	Name of the spawning parameter on the order that will cause this service bundle or service action to be added to the order if it has the correct value
spawn_value	varchar2	255	1	Value of the spawning parameter that will cause the service bundle or service action to be added to the order.
service_type	char	1	0	The service_type field indicates whether the table entry is a service bundle or a service action. • B – Service bundle • C – Service action

tbl_srt_bundle_csdl

This table contains the service action spawning information for the SRT.

Table 2-64 tbl_srt_bundle_csdl Columns

Column_name	Type	Length	Nulls	Description
service_id	varchar2	80	0	Service Identifier.
csdl_cmd	varchar2	80	0	Name of the CSDL.
csdl_seq_no	number	20	0	Sequence number of the CSDL.
cond_flag	char	1	0	Conditional flag for the CSDL. Possible values are: • A - ALWAYS • E - EQUALS • D - DEFINED • N - NOT_DEFINED
label	varchar2	80	1	The parameter label.
value	varchar2	255	1	Value of the parameter.
eval_exp	varchar2	255	1	Contains combination of parameter names, operators, and values to which the parameters are compared.
inc_ord_resp	char	1	1	Identifies whether order data need to be included in the response or not. • Y – Yes • N – No
description	varchar2	1024	1	Description of the service bundle or service action.

tbl_srt_config_reload

This table specifies the SRT configuration reload time.

Table 2-65 tbl_srt_config_reload Columns

Column_name	Type	Length	Nulls	Description
load_dts	date	-	0	Date and time at which the SRT configuration was reloaded.

tbl_srt_correlation

This table provides the correlation details for the SRT work order.

Table 2-66 tbl_srt_correlation Columns

Column_name	Type	Length	Nulls	Description
correlation_id	varchar2	80	0	SRT Correlation identifier
asap_id	varchar2	80	0	ASAP Work Order identifier

tbl_srt_csdl_parm

This table provides the service action parameter mapping details for the service bundle.

Table 2-67 tbl_srt_csdl_parm Columns

Column_name	Type	Length	Nulls	Description
service_id	varchar2	80	0	Service Identifier.
csdl_cmd	varchar2	80	0	The CSDL command.
csdl_seq_no	number	20	0	Sequence number of the CSDL.
parm_seq_no	number	20	0	Sequence number of the parameter.
bundle_label	varchar2	80	0	The upstream label used in transmitting the parameters for provisioning.
csdl_label	varchar2	80	0	The CSDL label used in transmitting the parameters for provisioning.
default value	varchar2	255	1	Default Value.
parm_type	char	1	0	Type of the parameter. Possible values include: • R - Required Scalar • O - Optional Scalar • C - Required Compound • N - Optional Compound • M - Required Indexed • I - Optional Indexed

tbl_srt_ctx

This table provides the correlation details for the SRT work order.

Table 2-68 tbl_srt_ctx Columns

Column_name	Type	Length	Nulls	Description
asap_id	varchar2	80	0	ASAP identifier.
name	varchar2	1024	0	Name of the parameter.
value	varchar2	1024	0	Value of the parameter.
type	varchar2	1024	0	Type of the parameter.

srt_header_mapping

This table enumerates XPath names and values to allow additional event XML message body data to be placed in the JMS header properties. This supports the inclusion of extended event data in JMS headers without additional database queries. If there are no entries in this table, then the default behavior occurs for JMS header creation. See "tbl_event_dataset" and "tbl_event_template."

The table contains of XPath names and values. The SRT iterates through the table entries and runs the configured XPaths. The returned name/value pairs are added to the JMS header properties.

If the XPath in the configuration name attribute returns multiple results, all returned values are added to the header. The values of these parameters are the result list of the value attribute. The name/value pairs are paired in the order they were returned by the XPath functions.

If the parameter name XPath returns more results than the value XPath the remaining values will be left blank.

If the parameter name XPath returns fewer results than the value XPath results, extra value XPath results are ignored.

Note: An XPath within single quotes represents a constant.

Table 2-69 srt_header_mapping Columns

Column_name	Type	Length	Nulls	Description
xpath_name	varchar2	1024	0	xpath to generate a name.
xpath_value	varchar2	1024	0	xpath to generate a value

Examples:

To return a specific information parameter on all event types, create a record with values similar to the following:

Name – 'IMSI'

Value – /*[name()=mslv-sa:completeEvent]//extendedWoProperties/
extendedWoProperty[name='IMSI']/value

To return all extendedWoProperties on a completed event, create a record with values similar to the following:

Name – /*[name()=mslv-sa:completeEvent]//extendedWoProperties//name

Value – /*[name()=mslv-sa:completeEvent]//extendedWoProperties//value

tbl_srt_lookup

This table provides lookup details used by the SRT.

Table 2-70 tbl_srt_lookup Columns

Column_name	Type	Length	Nulls	Description
name	varchar2	255	0	Name of the lookup.
type	varchar2	128	0	Type of the lookup.
cache_scope	varchar2	32	1	Scope of the lookup. Scope can be either: NONE, NODE or SYSTEM.
cache_timeout	varchar2	20	1	The amount of time in milliseconds to cache a value if the scope is SYSTEM.
cache_max_size	number	20	1	The maximum number of entries to cache if the scope is SYSTEM. Once MaxSize is reached, least recently used values will be dropped.

tbl_srt_lookup_input

This table provides lookup input details used by the SRT.

Table 2-71 tbl_srt_lookup_input Columns

Column_name	Type	Length	Nulls	Description
lookup_name	varchar2	255	0	Name of the lookup
parm_name	varchar2	255	0	Name of the input parameter
parm_type	char	1	0	Type of the parameter. Possible values are: • V - represents Adapter Properties Value • L - represents Lookup
parm_value	varchar2	255	0	Value of the parameter associated with the parameter name. If parm_type is L then parm_value contains an XPath, otherwise parm_value contains an instance string value.
parm_source	varchar2	255	0	Indicates the source XML document for the order data. Possible values are: • if parm_type is L, then parm_source contains the name of a data provider / Lookup • otherwise, if parm_source is ASAP_SRT_ORDER, this means that the source XML document contains the order data

tbl_srt_lookup_output

This table provides lookup output details used by the SRT.

Table 2-72 tbl_srt_lookup_output Columns

Column_name	Type	Length	Nulls	Description
lookup_name	varchar2	255	0	Name of the lookup.
parm_name	varchar2	255	0	Name of the output parameter.
parm_type	char	1	0	Type of the parameter. Possible values is: • X - represents XPATH
parm_value	varchar2	255	0	Value of the parameter.

tbl_srt_query_spawn

This table provides query spawning details for the SRT.

Table 2-73 tbl_srt_query_spawn Columns

Column_name	Type	Length	Nulls	Description
parm_name	varchar2	255	0	Name of the parameter
req_exp	varchar2	1024	0	Regular expression used by the SRT.
eval_cond	char	1	0	Evaluation condition

tbl_stubs

This dynamic table is used in the ASAP High Availability configuration to maintain a copy of the critical data relating to any work order in either ASAP site. As the SARM processes a work order locally, the other SARM is notified of any critical updates performed on the order by means of SARM to SARM communication. The other SARM updates its stub table with the critical update information.

Table 2-74 tbl_stubs Columns

Column_name	Type	Length	Nulls	Description
wo_id	varchar2	80	0	The Work order ID
wo_stat	number	38	0	The status of the work order. The possible values include: • (I) SRP_WO_INIT – Initial state work order. • (H) SRP_WO_HELD – Held work order. • (R) SRP_WO_REVIEW – Reviewed state work order. • (X) SRP_WO_CANCELLED – Cancelled work order. • (E) SRP_WO_TRAN_ERROR – Work order with a translation error. • (F) SRP_WO_FAIL – Failed work order. • (C) SRP_WO_COMP – Completed work order.
wo_cmd	number	38	0	The work order command.
srp_id	varchar2	8	0	The SRP from which this work order originated.
sched_dts	date	-	0	The due date and time of the work order.

Table 2-74 (Cont.) tbl_stubs Columns

Column_name	Type	Length	Nulls	Description
crit_seq_no	number	38	1	The critical sequence number representing the current critical update that was last applied to the work order.
parent_wo	varchar2	80	1	The parent work order of this order.

Table indexes:

```
Wo_id, parent_wo
```

tbl_test_rpc_parm

This table provides the details about the RPC parameters used by SRT.

Table 2-75 tbl_test_rpc_parm Columns

Column_name	Type	Length	Nulls	Description
rpc	varchar2	80	0	The Work order ID
seq_no	number	38	0	The status of the work order.
parm_lbl	varchar2	80	0	The work order command.
parm_typ	char	1	0	The SRP from which this work order originated
default_vlu	varchar2	255	1	The due date and time of the work order

Table indexes:

```
rpc, seq_no
```

tbl_stat_text

Stores labels for the OCA client.

- **SSP_new_stat_text** adds new static text labels to tbl_stat_text.
- **SSP_del_stat_text** deletes static text labels from tbl_stat_text.
- **SSP_list_stat_text** lists the contents of this table.

Table 2-76 tbl_stat_text Columns

Column_name	Type	Length	Nulls	Description
stat_id	varchar2	10	0	ID for a group of labels.
status	number	38	1	Integer key field for grouping.
code	varchar2	20	1	String key field for grouping.
stat_text	varchar2	100	0	Text describing the label.

Table indexes:

```
Indexes
```

```
stat_id, status
stat_id, code
```

tbl_unload_sp

This table provides information about the stored procedures used to insert/delete/ data to/from the ASAP tables. This table will be used by the utilities to load/unload the data in the ASAP tables.

This table is available in both the Control and SARM databases, containing data appropriate to the respective database.

Table 2-77 tbl_unload_sp Columns

Column_name	Type	Length	Nulls	Description
seq_no	number	38	0	Sequence number of the table.
tbl_name	varchar2	40	0	Name of the ASAP table.
new_sp	varchar2	40	1	Stored procedure used to insert the data into the table.
del_sp	varchar2	40	1	Stored procedure used to delete the data from the table.
list_sp	varchar2	40	1	Stored procedure used to list the data in the table.

Table indexes:

```
seq_no
```

tbl_unload_param

This table is used by utilities. It provides information about the parameters inserted/deleted by stored procedures.

This table is available in both the Control and SARM databases, containing data appropriate to the respective database.

Table 2-78 tbl_unload_param Columns

Column_name	Type	Length	Nulls	Description
seq_no	number	38	0	Sequence number of the table.
col_number	number	38	0	Column number of the parameter in the table.
para_name	varchar2	80	0	Name of the Parameter.
default_flag	number	38	0	Default Flag.
sp_type	char	3	0	Type of the stored procedure.
rows_int	number	38	0	Number of rows.

Table indexes:

```
seq_no, para_name, sp_type
```

temp_wrk_ord

This dynamic table details the supplemental work order information required by the ASAP functionality.

Table 2-79 temp_wrk_ord Columns

Column_name	Type	Length	Nulls	Description
wo_id	varchar2	80	0	The work order ID that uniquely identifies the work order in the ASAP core.
sched_dts	Date	-	0	The scheduled date and time for provisioning to occur on work orders due in the future. For immediate work orders, set it to the current time.
wo_stat	number	38	0	The status of the work order. It is updated by the SARM as the work order is being processed.
comp_dts	Date	-	1	The completion date of all provisioning associated with the work order in ASAP.
srp_id	varchar2	8	0	The logical name of the front-end SRP that notifications and results are sent to for a particular work order.
update_dts	Date	-	0	The date and time of the last update on the work order within the ASAP core.
org_unit	varchar2	8	0	The Organization Unit of the person or group to whom notification is sent should particular events occur on the work order.
orig_login	varchar2	64	1	The original login ID of the user who initiated the work order in the Host order system. This information is used for display, notification, and diagnostic purposes.
revs_flag	char	1	0	The revisions flag on the work order to indicate if the work order was revised by the OCA client.
exceptions	char	1	0	Exceptions flag indicating to the SRP if there are any exceptions in the completion of the work order. Such exceptions are generally the result of a “Fail but Continue" status being returned to the SARM for one of the ASDLs on the work order. This field is set by the SARM and communicated to the relevant SRP, which then requests the exception details. The possible values include: • (Y) ASAP_WO_EXCEPTIONS – The work order completed with exceptions. • (N) ASAP_WO_NO_EXCEPTIONS – The work order completed without any exceptions. The flag is passed back to the SRP by the SARM in the Work Order Completion Event. The values are defined in the header file asap_core.h.
pend_cancel	char	1	1	This flag indicates whether there is a pending order cancellation for this work order.

Table 2-79 (Cont.) temp_wrk_ord Columns

Column_name	Type	Length	Nulls	Description
rollback_stat	number	38	0	The rollback status of the work order.
command	number	38	0	This field is transmitted by the SRP to the SARM and informs the SARM processing command to apply it to the work order.
crit_seq_no	number	38	1	The sequence number of the last critical update performed on this work order. This is updated each time a critical update to the work order has been performed.
lock_uid	varchar2	64	1	The user who last locked the work order for updating.
lock_dts	Date	-	1	The date and time that the work order was last locked for updating.
start_dts	Date	-	1	The date and time that the work order started provisioning.
asdl_timeout	number	38	1	If set, the ASDL time-out interval to be used on the work order instead of the system-wide SARM default which is specified by the configuration parameter ASDL_TIMEOUTS.
parent_wo	varchar2	80	1	If set, the parent work order on which the work order is dependent. The parent order must be completed for the work order to begin provisioning.
wo_timeout	number	38	1	If set, the work order time-out interval to be used on the work order instead of the system-wide SARM default which is specified by the configuration parameter ORDER_TIMEOUT.
asdl_retry_num	number	38	1	If set, the number of ASDL retries to be used on the work order instead of the system-wide SARM default which is specified by the configuration parameter NUM_TIMES_RETRY.
asdl_retry_int	number	38	1	If set, the time period in seconds between ASDL retries to be used on the work order instead of the system-wide SARM default.
wo_rback	char	1	1	If set to: • (Y)es – A flag specifies whether to explicitly roll back the order in the event of failure. • (N)o – The order is not rolled back. • (D)efault – The SARM receives the setting from the SRP.
stub_update_req	char	1	1	Indicates whether Stub update is required or not.
asdl_delay_fail	char	1	1	If set, a flag which specifies whether or not to treat hard errors encountered on the work order as delayed failures, therefore allowing the work order to finish processing before failing.
max_delay_fail	number	38	1	If set, the number of ASDL delayed failures to allow before failing a batch work order. This allows the overriding of the system-wide SARM default which is specified by the configuration parameter, BATCH_THRESHOLD.

Table 2-79 (Cont.) temp_wrk_ord Columns

Column_name	Type	Length	Nulls	Description
srq_id	number	38	1	The ID of the latest SRQ associated with the work order. There is a single SRQ related to a given work order.
is_future_dated	char	1	1	If set to: • Y – It indicates that the work order is future-dated. • N – The work order is not future-dated. This is used by the ADMIN Server.
batch_group	varchar2	80	1	The batch group ID of the work order.
extsys_di	varchar2	128	1	An external system ID that is passed with the work order. It is not used by the SARM, but is used by the SRP for proper routing to upstream systems.
rollback_exceptions	char	1	0	The rollback exceptions flag returned by the SARM on the work order rollback completion notification. Indicates whether there are exceptions (i.e. ASDL failure) during rollback of a work order.
point_of_no_return	number	38	0	Values are: • -1 – No rollback if work order fails • 0 – (default). Normal rollback behavior. No 'point of no return' functionality. • >0 – ASDL_SEQ_NO. This ASDL is the 'point of no return' for partial rollback. If rollback occurs, and the processing has continued beyond this point, roll back to this ASDL but no further.
failure_reason	varchar2	255	1	Provides the failure reason for a work order that fails during provisioning.

Table indexes:

```
wo_id, srq_id
```

tbl_uid_pwd

If the SARM configuration variable SECURITY_CHECK is enabled, this static table is referenced to authorize access to the SARM from the SRP as part of the SRP to SARM protocol.

For the security check to be validated, the user ID and password on the work order must be defined in the table with an active status. If a security violation is detected, the SARM rejects the work order with a security violation, ASAP_STAT_SECURITY_VIOLATION (103).

This provides a central security mechanism to ensure that only properly validated work orders are received by the SARM from all SRPs in the system.

- **SSP_new_userid** adds a new user account for the SARM to control access from the SRP in tbl__uid_pwd.
- **SSP_del_userid** deletes a user account for the SARM to control access from the SRP in tbl__uid_pwd.

- **SSP_list_userid** lists the contents of this table.

Table 2-80 tbl_uid_pwd Columns

Column_name	Type	Length	Nulls	Description
userid	varchar2	64	0	The user ID for the security check.
pwd	varchar2	30	0	The associated password.
status	varchar2	40	1	Your current status. If set to ACTIVE, then you can access. If not, access is denied.

Table indexes:

```
userid
```

tbl_unid

This dynamic table manages unique IDs required by other tables. It is present in most user-created databases and provides a method of generating a serial field.

Table 2-81 tbl_unid Columns

Column_name	Type	Length	Nulls	Description
unid_type	varchar2	8	0	A unique code identifying the UNID type. This allows there to be many different UNID values for different types of UNID.
unid	number	38	0	The UNID value for a particular type.
pad1	char	255	0	Padding to make a table row occupy a page. This reduces concurrence on the database data page by different database processes.
pad2	char	255	0	Same as pad1.
pad3	char	255	0	Same as pad1.
pad4	char	255	0	Same as pad1.

Table indexes:

```
Unique

unid_type
```

tbl_user_err

This static table provides a mechanism to define user exit codes and map them to one of the base ASDL exit types. For more information on user exit types, see the *ASAP Cartridge Development Guide*.

If the Java methods return a user-defined exit type, the NEP checks whether there is a base exit type defined for the specified ASDL and exit type. If so, then the base exit type is used. If not, the NEP determines whether there is a user-defined error type associated with the user-defined exit type.

Some initial data is provided as part of the core system. You are responsible for populating and maintaining this table.

Pattern matching is ordered by length (search_pattern), csdl, asdl, ne_vendor, sftwr_load and tech_type. Pattern matching starts from the most specific specification to the least specific. The most specific specification is one that has all columns within the table filled.

- **SSP_new_err_type** adds a new mapping of user-defined error types.
- **SSP_del_err_type** deletes the mapping of user-defined error types.
- **SSP_list_err_type** list the contents of this table

Table 2-82 tbl_user_err Columns

Column_name	Type	Length	Nulls	Description
csdl	varchar2	80	1	The CSDL that is run. Error types can be defined for user_type and CSDL combinations.
asdl	varchar2	80	1	The ASDL that is run. Error types can be defined for user_type and ASDL combinations.
vendor	varchar2	255	1	The vendor of the network element.
tech_type	varchar2	255	1	The technology of the network element.
sftwr_load	varchar2	16	1	Software version of the host network element.
search_pattern	varchar2	255	1	Regular expression pattern that is used to match on network element responses.
user_type	varchar2	20	0	User-defined ASDL exit type.
base_type	varchar2	20	0	The base ASDL exit type where this user specified ASDL exit type maps to. The base types include: • SUCCEED – ASDL run successfully. • FAIL – ASDL encountered a hard error. • RETRY – ASDL to be retried in future. • MAINTENANCE – ASDL failed because the NE is currently unavailable to receive provisioning requests. • SOFT_FAIL – ASDL generates an error occurs that should not halt the processing of the order. • DELAYED_FAIL – ASDL failed during provisioning. The SARM skips any subsequent ASDL in the CSDL, continues provisioning at the next CSDL, and then fails the order. • STOP – ASDL is stopped. Refer to the *ASAP Cartridge Development Guide* for more detailed descriptions of these base_types.
description	varchar2	255	1	Description of the user exit type.

Table indexes:

```
Unique

csdl, asdl, ne_vendor, tech_type, sftwr_load, search_pattern, user_type
```

tbl_user_err_threshold

With this static table you can specify thresholds for a specific user exit from the Interpreter by Host NE and ASDL command.

The SARM maintains three counters for each Host NE, ASDL, and user exit type. If a counter exceeds its user configured threshold, the SARM issues the appropriate event, if defined. In the Control database you can configure any system events generated from this table to map to relevant system alarms as required.

- **SSP_new_user_err_threshold** adds a new user-defined error threshold or set of thresholds to tbl_user_err_threshold.
- **SSP_del_user_err_threshold** deletes a user-defined error threshold or set of thresholds from tbl_user_err_threshold.
- **SSP_list_user_err_threshold** lists the contents of this table

Table 2-83 tbl_user_err_threshold Columns

Column_name	Type	Length	Nulls	Description
host_clli	varchar2	80	0	The Host NE.
asdl_cmd	varchar2	80	0	The ASDL command.
user_type	varchar2	20	0	The user exit type.
minor_threshold	number	4	1	The number of user exits for this Host NE and ASDL before the minor event is generated.
minor_event	varchar2	8	1	The minor system event to be generated.
major_threshold	number	4	1	The number of user exits for this Host NE and ASDL before the major event is triggered.
major_event	varchar2	8	1	The major system event generated.
critical_threshold	number	4	1	The number of user exits for this Host NE and ASDL before the critical event is triggered.
critical_event	varchar2	8	1	The critical system event generated.

Table indexes:

```
Unique

host_clli, asdl_cmd, user_type
```

tbl_usr_wo_prop

This table is used to dynamically configure work order properties. It consists of WO ID and name/value pair for a user-defined work order properties. This table supports only string type user-defined work order properties.

Table 2-84 tbl_usr_wo_prop Columns

Column Name	Type	Length	Nulls	Description
wo_id	varchar2	80	0	The ID of the work order.
name	varchar2	80	0	The name of the work order property.

Table 2-84 (Cont.) tbl_usr_wo_prop Columns

Column Name	Type	Length	Nulls	Description
value	varchar2	255	1	The value given to the work order property.
value_hint	char	1	0	Indicates whether the parameter value is a real value (XML document) or a reference ID (XPATH expression). Can be one of the following: • X - indicates a reference to XML document • P - indicates a reference to XPath expression • T - indicates a reference to large text (reserved for future use)

Table indexes:

```
Unique

wo_id, name
```

tbl_wo_audit

This dynamic table tracks the work order status. It is populated based on the wo_audit column in "tbl_msg_convert" and the WO_AUDIT_LEVEL parameter in ActivationConfig.xsd and ASAP.cfg.

You can view audit log information through the OCA Client. For more information, refer to the *ASAP OCA User Guide*.

Table 2-85 tbl_wo_audit Columns

Column_name	Type	Length	Nulls	Description
wo_id	varchar2	80	0	The ID of the work order.
wo_audit_log_identity	number	20	0	Unique ID of the audit message. This is an identity field automatically generated by the RDBMS upon insertion.

Table 2-85 (Cont.) tbl_wo_audit Columns

Column_name	Type	Length	Nulls	Description
wo_stat	smallint	1	0	The status of the work order internally within ASAP. It is updated by the SARM as the work order is being processed. Possible values for this field include: • (101) WO_LOADING – The work order is being loaded into the SARM from the SRP. • (102) WO_INIT – The work order is in the Initial state awaiting provisioning. • (103) WO_IN_PROGRESS – The work order is currently being provisioned. • (104) WO_COMPLETE – The work order has been completed. • (200) WO_GET_STATUS – Transient state within the SARM. • (221) WO_STOP_WAIT – The work order has been stopped and is being rolled back. • (222) WO_STOPPED – The work order is currently stopped. • (246) WO_REVIEW – The work order is in a Review state (similar to held). • (247) WO_CANCEL_WAIT – The work order has a cancellation request awaiting. • (249) WO_LOCK – The work order is in a Locked state. • (250) WO_ABORT – The work order has been stopped. • (251) WO_TIME_OUT – Work order processing has exceeded the time-out interval. • (252) WO_CANCELLED – The work order has been cancelled, usually from the Host system. • (253) WO_FAILED – The work order provisioning has failed. • (254) WO_HELD – The work order is placed in a Held state. • (255) WO_TRANSLATION_FAIL – The work order translation failed in the SRP, but was transmitted to the SARM to facilitate manual intervention. These values are defined in the header file sarm_defs.h

Table 2-85 (Cont.) tbl_wo_audit Columns

Column_name	Type	Length	Nulls	Description
srq_evt	varchar2	8	1	The SRQ Log Event. The SRP may inquire for the SRQ log by specifying particular SRQ events of interest in the inquiry RPCs. Possible values include: • SRQ_INFO_EVENT "INFO" – Information messages. • SRQ_ERROR_EVENT "ERROR" – Error messages. • NE_CMD_EVENT "NE_CMD" – Command entered to NE. • NE_RESP_EVENT "NE_RESP" – Response from NE. These values are defined in the header file sarm_defs.h.
evt_dt_tm	date	-	1	Last date and time the SRQ record was updated. This value is referenced when performing audit log queries in the OCA client.
evt_text	varchar2	255	1	Description of the event.
user_id	varchar2	64	1	The user ID.
sched_dts	date	-	1	The scheduled date and time for provisioning to occur on this work order. It is used for work orders that are due in the future. For immediate work orders, set it to the current time. This value is referenced when performing work order queries in the OCA client.
priority	varchar2	21	0	Priority of the SRQ as assigned by the Host order system. It is equal to the work order priority. Possible values include: • (1) ASAP_SRQ_HIGH_PRIO – High priority SRQ. • (5) ASAP_SRQ_NORMAL_PRIO – Normal priority SRQ. • (9) ASAP_SRQ_LOW_PRIO – Low priority SRQ. These values are defined in asap_core.h. This priority field is the first field in the internal composite ASAP SRQ priority within the ASAP core followed by the SRQ due date and the SRQ action such as Remove, Change, or Add.
batch_group	varchar2	80	1	The batch group ID of the work order.
parent_wo	varchar2	80	1	If set, the parent work order on which this order is dependent. The parent work order must be completed for provisioning to begin.

Table 2-85 (Cont.) tbl_wo_audit Columns

Column_name	Type	Length	Nulls	Description
org_unit	varchar2	8	0	The Organization Unit of the person or group to whom notification is sent should particular events occur on the work order. It is used for notification purposes and user group determination.
grp_cd	char	1	0	The action of the SRQ. Possible values include: • (A) ADD – Addition of service. • (R) REMOVE – Removal of service. • (C) CHANGE – Change/update of existing service. • (Q) QUERY – Query existing service. This field determines the order in which SRQs are processed within the same work order and SRQs between work orders. These values are defined in asap_core.h.
wo_event_time	date	-	1	The date and time that the work order event was placed into the audit log.
wo_event_location	varchar2	255	1	The location of the work order event. For example, which pending queue the ASDL was placed in.

Table indexes:

```
wo_id, wo_audit_log_identity
```

tbl_wo_event_queue

This dynamic table stores a back-up copy of all SRP events generated. Completed events are purged on a periodic basis.

Table 2-86 tbl_wo_event_queue Columns

Column_name	Type	Length	Nulls	Description
wo_id	varchar2	80	0	The work order ID.
event_unid	number	4	0	A unique ID of this event. If an alarm is generated by this event, an alarm log entry in tbl_alarm_log is created with this event unid value.
event_type	number	4	0	The event type. It specifies if a system alarm is generated by accessing the static table, tbl_event_type.
event_status	number	4	0	The current status of the event.
srp_id	varchar2	8	0	The logical name of the front end SRP where notifications and results are sent for this particular work order, for example, SRP_EMUL.
event_dts	date	-	0	The date and time of the system event.

Table 2-86 (Cont.) tbl_wo_event_queue Columns

Column_name	Type	Length	Nulls	Description
start_dts	date	-	0	The date and time of when provisioning starts for the ASDL. The difference between this value and queue_dts represents the time the ASDL spent in the queue of pending ASDLs to that Host NE before being transmitted to the NEP for provisioning.
complete_dts	date	-	0	The date and time of when the ASDL provisioning is completed.
estimate	number	4	1	The work order processing estimate returned to the SRP by the SARM after the work order was transmitted to the SARM by the SRP.
misc	varchar2	80	1	This field specifies miscellaneous information received by the SARM through the asap_wo_begin RPC. It contains another work order identifier associated with the primary work order.
rev_flag	char	1	1	The revisions flag returned by the SARM on the work order completion notification.
exceptions	char	1	1	The exceptions flag returned by the SARM on the work order completion notification.
mach_clli	varchar2	80	1	Remote NE.
host_clli	varchar2	80	1	Host NE to which the Remote NE is connected.
reason	varchar2	80	1	A description of the system event.
csdl_seq_no	number	4	1	Sequence number of the CSDL within the logical work order.
csdl_id	number	4	1	The ID of the CSDL being processed when the Information parameter is generated. This CSDL ID is the same as the ID in tbl_srq_csdl. It allows the SRP to track the CSDL that this information parameter belongs to.
timeout_status	number	4	1	The status of the work order when a timeout occurs. Possible values include: • (90) ASAP_TIMEOUT_EXECUTING – The work order timed out but was still running. • (91) ASAP_TIMEOUT_FAIL – The work order timed out and failed.
queue_tm	float	8	1	Timestamp when the event was generated.

Table 2-86 (Cont.) tbl_wo_event_queue Columns

Column_name	Type	Length	Nulls	Description
xaction_type	varchar2	16	1	The type of SRP event to be sent. Possible values include: • WO_ACCEPT • WO_ESTIMATE • WO_STARTUP • WO_COMPLETION • WO_FAILURE • WO_SOFT_ERROR • WO_BLOCKED • WO_ROLLBACK • WO_TIMEOUT • NE_UNKNOWN • NE_UNAVAILABLE • NE_AVAILABLE The WO_ACCEPT and WO_STARTUP events are turned off by default. See tbl_asap_srp for turning on the events.
old_wo_stat	number	4	1	The previous status of the work order in the SARM.
new_wo_stat	number	4	1	Status of the new work order.
status	number	4	1	The current status of the work order in the SARM.
extsys_id	varchar2	128	1	An external system ID passed with the work order. It is not used by the SARM, but is used by the SRP for proper routing to upstream systems.
rollback_exceptions	char	1	1	The rollback exceptions flag returned by the SARM on the work order rollback completion notification.

Table indexes:

```
Indexes

wo_id, event_unid
event_status, srp_id
```

tbl_wrk_ord (SARM)

This dynamic table details the essential work order information required by the ASAP core functionality. It contains details required by the provisioning process but does not contain customer-specific work order details.

Table 2-87 tbl_wrk_ord (SARM) Columns

Column_name	Type	Length	Nulls	Description
wo_id	varchar2	80	0	The work order ID that uniquely identifies the work order in the ASAP core. This ID is the same as the one used in the Host system and allows easier user reference to orders in the OCA client.

Table 2-87 (Cont.) tbl_wrk_ord (SARM) Columns

Column_name	Type	Length	Nulls	Description
sched_dts	date	-	0	The scheduled date and time for provisioning to occur on work orders due in the future. For immediate work orders, set it to the current time.
wo_stat	number	1	0	The status of the work order. It is updated by the SARM as the work order is being processed. Possible values include: • (101) WO_LOADING – The work order is being loaded into the SARM from the SRP. • (102) WO_INIT – The work order is in the Initial state awaiting provisioning. • (103) WO_IN_PROGRESS – The work order is being provisioned. • (104) WO_COMPLETE – The work order is completed. • (200) WO_GET_STATUS – Transient state within the SARM. • (221) WO_STOP_WAIT – The work order has stopped and is being rolled back. • (222) WO_STOPPED – The work order is stopped. • (246) WO_REVIEW – The work order is in a Reviewed state, similar to Held. • (247) WO_CANCEL_WAIT – The work order has a cancellation request waiting. • (249) WO_LOCK – The work order is in a Locked state. • (250) WO_ABORT – The work order has been stopped. • (251) WO_TIME_OUT – Work order processing has exceeded the timeout interval. • (252) WO_CANCELLED – The work order has been cancelled, usually from the Host system. • (253) WO_FAILED – The work order provisioning has failed. • (254) WO_HELD – The work order is placed in a Held state. • (255) WO_TRANSLATION_FAIL – The work order translation failed in the SRP, but was transmitted to the SARM to facilitate manual intervention. These values are defined in the header file sarm_defs.h.
comp_dts	date	-	1	The completion date of all provisioning associated with the work order in ASAP.
srp_id	varchar2	8	0	The logical name of the front-end SRP that notifications and results are sent to for a particular work order.
update_dts	date	-	0	The date and time of the last update on the work order within the ASAP core.

Table 2-87 (Cont.) tbl_wrk_ord (SARM) Columns

Column_name	Type	Length	Nulls	Description
org_unit	varchar2	8	0	The Organization Unit of the person or group to whom notification is sent should particular events occur on the work order.
orig_login	varchar2	64	1	The original login ID of the user who initiated the work order in the Host order system. This information is used for display, notification, and diagnostic purposes.
revs_flag	char	1	0	The revisions flag on the work order to indicate if the work order was revised by the OCA client. The possible values include: • (Y) ASAP_WO_REVISIONS – The work order has associated revisions. • (N) ASAP_WO_NO_REVISIONS – The work order has no associated revisions. This flag is passed back to the SRP by the SARM in the Work Order Completion Event. Such values are defined in the header file asap_core.h.
exceptions	char	1	0	Exceptions flag indicating to the SRP if there are any exceptions in the completion of the work order. Such exceptions are generally the result of a “Fail but Continue" status being returned to the SARM for one of the ASDLs on the work order. This field is set by the SARM and communicated to the relevant SRP, which then requests the exception details. The possible values include: • (Y) ASAP_WO_EXCEPTIONS – The work order completed with exceptions. • (N) ASAP_WO_NO_EXCEPTIONS – The work order completed without any exceptions. The flag is passed back to the SRP by the SARM in the Work Order Completion Event. The values are defined in the header file asap_core.h.
pend_cancel	char	1	1	This flag indicates if there is a pending order cancellation for this work order. Upon completion of the next ASDL on the order, the SARM checks this flag. If set, the SARM initiates rollback if configured at the ASDL level on the work order. The possible values are: • Y – Cancellation pending on the work order. • N – No cancellation received.

Table 2-87 (Cont.) tbl_wrk_ord (SARM) Columns

Column_name	Type	Length	Nulls	Description
rollback_stat	number	1	0	The rollback status of the work order. If any CSDL on the work order requires rollback, this field is set to (201) RBACK_REQUIRED. Otherwise, it is set to (200) BACK_NOT_REQUIRED and no rollback takes place if the work order fails. If rollback is required, it starts on the work order and the SARM updates this field to (202) RBACK_IN_PROGRESS. This is a transient state and at the end of the rollback (Complete or Failed), the rollback status reverts back to its original state. If the rollback procedure fails, the field is updated to (204) RBACK_FAILED, otherwise (203) RBACK_COMPLETE to denote successful rollback. These values are defined in the header file sarm_defs.h.
command	number	4	0	This field is transmitted by the SRP to the SARM and informs the SARM processing command to apply it to the work order. Possible values are: • ASAP_CMD_WO_UPDATE – The work order is either new or an update to an existing one. • ASAP_CMD_WO_CANCEL – The work order is a cancellation request on an existing order. • ASAP_CMD_WO_TRAN_ERROR – Indicates to the SARM that a translation error occurred on the work order. • ASAP_CMD_WO_HELD – The work order is to be held by the SARM until released by either the originating system with an update request or through the user interface. • ASAP_CMD_WO_REVIEW – The work order is to be held in a Reviewed state by the SARM until released by either the originating system through an update request or through the user interface. • ASAP_CMD_WO_REPLACE – Rollback an existing work order and submit with the same wo_id but a different set of data These values are defined in the header file asap_core.h.
crit_seq_no	number	4	1	The sequence number of the last critical update performed on this work order. This is updated each time a critical update to the work order has been performed.
lock_uid	varchar2	64	1	The user who last locked the work order for updating.
lock_dts	date	-	1	The date and time that the work order was last locked for updating.

Table 2-87 (Cont.) tbl_wrk_ord (SARM) Columns

Column_name	Type	Length	Nulls	Description
start_dts	date	-	1	The date and time that the work order started provisioning.
asdl_timeout	number	4	1	If set, the ASDL time-out interval to be used on the work order instead of the system-wide SARM default which is specified by the configuration parameter ASDL_TIMEOUTS.
parent_wo	varchar2	80	1	If set, the parent work order on which the work order is dependent. The parent order must be completed for the work order to begin provisioning.
wo_timeout	number	4	1	If set, the work order time-out interval to be used on the work order instead of the system-wide SARM default which is specified by the configuration parameter ORDER_TIMEOUT. The order timeout behavior is governed by two parameters: the wo_timeout parameter on the work order and the ORDER_TIMEOUT configuration parameter in ASAP.cfg. If wo_timeout has a value greater than one, it is used. If wo_timeout has a value of zero, work orders do not time out. If wo_timeout has a value less than zero, ORDER_TIMEOUT is used. If wo_timeout has a value less than zero and ORDER TIMEOUT has a value of zero or less than zero, work orders do not time out. The work order/ASDL timer starts after the work order has been submitted and the first ASDL starts provisioning. This threshold can be exceeded if, for example, the connection to an NE is interrupted after the connection has been established.
asdl_retry_num	number	4	1	If set, the number of ASDL retries to be used on the work order instead of the system-wide SARM default which is specified by the configuration parameter NUM_TIMES_RETRY.
asdl_retry_int	number	4	1	If set, the time period in seconds between ASDL retries to be used on the work order instead of the system-wide SARM default.
wo_rback	char	1	1	If set to: • (Y)es – A flag specifies whether to explicitly roll back the order in the event of failure. • (N)o – The order is not rolled back. • (D)efault – The SARM receives the setting from the SRP.
stub_update_req	-	-	-	-

Table 2-87 (Cont.) tbl_wrk_ord (SARM) Columns

Column_name	Type	Length	Nulls	Description
asdl_delay_fail	char	1	1	If set, a flag which specifies whether or not to treat hard errors encountered on the work order as delayed failures, therefore allowing the work order to finish processing before failing. This is used to override the behavior of the NEP.
max_delay_fail	number	4	1	If set, the number of ASDL delayed failures to allow before failing a batch work order. This allows the overriding of the system-wide SARM default which is specified by the configuration parameter, BATCH_THRESHOLD.
srq_id	number	4	1	The ID of the latest SRQ associated with the work order. There is a single SRQ related to a given work order.
is_future_dated	char	1	1	If set to: • Y – It indicates that the work order is future-dated. • N – The work order is not future-dated. This is used by the ADMIN Server.
batch_group	varchar2	80	1	The batch group ID of the work order.
extsys_id	varchar2	128	1	An external system ID that is passed with the work order. It is not used by the SARM, but is used by the SRP for proper routing to upstream systems.
rollback_exceptions	char	1	0	The rollback exceptions flag returned by the SARM on the work order rollback completion notification. Indicates whether there are exceptions (i.e. ASDL failure) during rollback of a work order.
point_of_no_return	number	38	0	Values are: • -1 – No rollback if work order fails • 0 – (default). Normal rollback behavior. No 'point of no return' functionality. • >0 – ASDL_SEQ_NO. This ASDL is the 'point of no return' for partial rollback. If rollback occurs, and the processing has continued beyond this point, roll back to this ASDL but no further.
failure_reason	varchar2	255	1	Provides the failure reason for a work order that fails during provisioning.

Table indexes:

```
Indexes

wo_id
wo_stat, sched_dts, org_unit
parent_wo
batch_group
```

temp_csdl_estim

A temporary storage table for the Oracle **SSP_csdl_list** function. Data from this table is automatically maintained by the function; therefore, you should not manually add or remove data.

Table 2-88 temp_csdl_estim Columns

Column_name	Type	Length	Nulls	Description
sess_id	number	22	1	Oracle session ID of the session running the function.
csdl_seq_no	number	4	1	CSDL sequence number.
estimate	number	4	1	CSDL estimate.

Table indexes:

```
sess_id
```

temp_csdl_list

A temporary storage table for the Oracle **SSP_csdl_list** function. Data from this table is automatically maintained by the function; therefore, you should not manually add or remove data.

Table 2-89 temp_csdl_list Columns

Column_name	Type	Length	Nulls	Description
sess_id	number	22	1	Oracle session ID of the session running the function.
csdl_cmd	varchar2	80	1	CSDL command.
csdl_st	number	1	1	CSDL status.
csdl_seq_no	number	4	1	CSDL sequence number.
csdl_id	number	4	1	CSDL ID.
description	varchar2	255	1	Description of the CSDL.
prov_sequence	number	4	1	Provisioning sequence of the CSDL.
asdl_seq_no	number	4	1	ASDL sequence number.

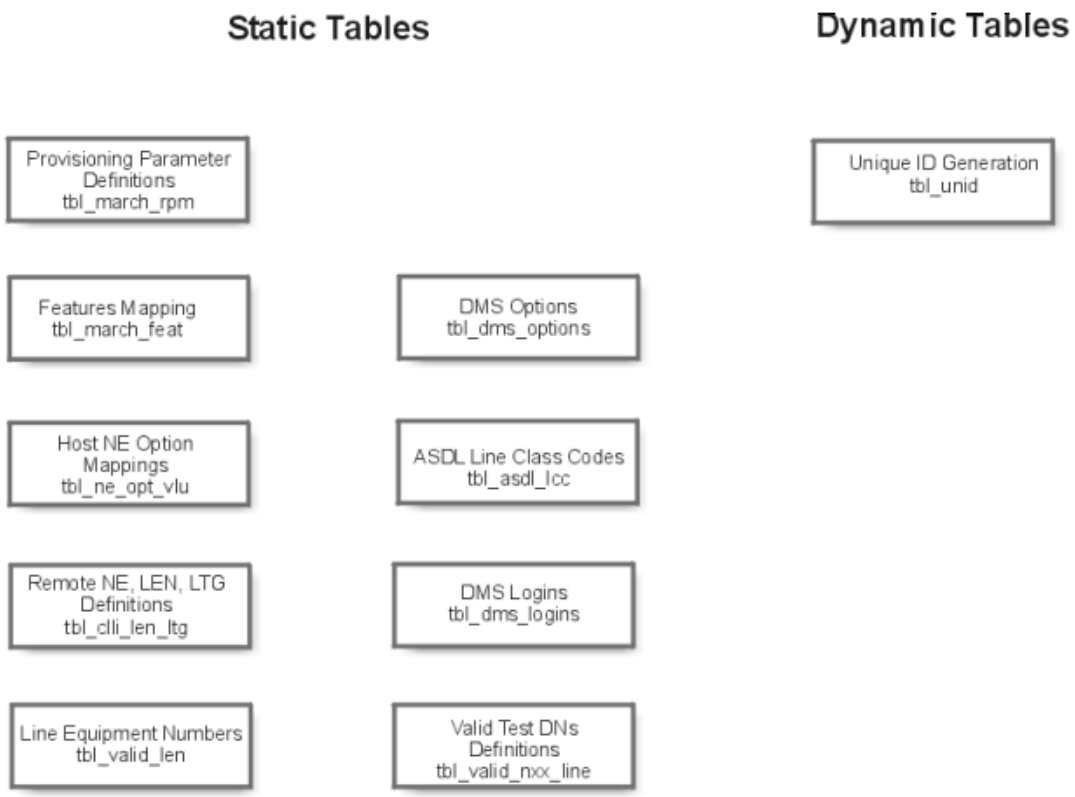
Table indexes:

```
sess_id
```

NEP database

This section details the NEP Database tables.

Figure 2-1 NEP database tables



User-created Database Tables

Following is a list of user-created database tables.

tbl_asdl_lcc

This static table contains the line class codes for certain ASDLs. It is only used by NE queries.

Table 2-90 tbl_asdl_lcc Columns

Column_name	Type	Length	Nulls	Description
asdl	char	80	0	The ASDLs that add an access line.
lcc	char	5	0	The Line Class Code (LCC) for a specific ASDL adding an access line.

Table indexes:

`asdl`

tbl_clli_len_ltg

This static table contains specific line information. For example, site code, line treatment group, intercept information, and so on.

Table 2-91 tbl_clli_len_ltg Columns

Column_name	Type	Length	Nulls	Nulls
mach_clli	varchar2	128	0	Remote NE.
nxx	varchar2	3	0	The NXX or Exchange working from this Remote NE.
site	char	4	1	Site code associated with this Remote NE and NXX.
ltg	number	38	1	Line treatment group for determining if calls are toll calls.
pub_incpt	char	4	1	Indicates if the intercept on the line is for a directory-published number.
non_pub_incpt	char	4	1	Indicates if the intercept on the line is for a non-published directory number.
clli_desc	varchar2	255	1	Description on the Remote NE.
from_len	varchar2	7	0	Start of LEN (Line Equipment Number) range.
to_len	varchar2	7	0	End of LEN range.
host_clli	char	80	0	The Host NE managing the Remote NE.

Table indexes:

```
Indexes

mach_clli, nxx, from_len
host_clli, site, from_len
```

tbl_dms_logins

This static table is used by the NEP and contains the login user IDs and passwords for the connections to the DMS NEs. You are responsible for populating and maintaining this table.

Table 2-92 tbl_dms_logins Columns

Column_name	Type	Length	Nulls	Description
host_clli	char	80	0	Host NE.
login_id	char	30	0	The login user ID that ASAP uses for this Host NE.
password1	char	30	0	The first NE password in a sequence of two passwords which are alternated.
password2	char	30	0	The second NE password.
last_changed	date	-	0	Date and time that the password was last changed.

Table indexes:

```
host_clli, login_id
```

tbl_dms_options

This static table is used to specify DMS specific options.

Table 2-93 tbl_dms_options Columns

Column_name	Type	Length	Nulls	Description
tech	char	16	0	The technology of the Host NE.
sftwr_load	char	16	0	The software load of the Host NE.
dms_option	varchar2	10	0	The DMS option.

Table indexes:

```
tech, sftwr_load, dms_option
```

tbl_march_feat

This static table provides the ability to map a generic option to a switch specific option.

Table 2-94 tbl_march_feat Columns

Column_name	Type	Length	Nulls	Description
generic_feat	varchar2	20	0	The generic feature.
tech	varchar2	16	0	The technology of the Host NE.
switch_feat	varchar2	20	0	The switch specific option.

Table indexes:

```
generic_feat, tech
```

tbl_march_rpm

This static table provides NEP access to provisioning parameters that are specific to a particular Host NE.

Only the values of the parameters (ASDL parameters) are used, therefore the order of the parameters is fixed. The following is the order of the various parameter types:

- (%TYPE, %MCLI)
- (%TYPE, %MCLI, %NXX)
- (%TYPE, %MCLI, %USOC)
- (%TYPE, %MCLI, %USOC, %NXX)
- (%TYPE, %MCLI, %FEAT)
- (%TYPE, %MCLI, %PIC)
- (%TYPE, %MCLI, %USOC, %LCC)
- (%TYPE, %MCLI, %USOC, %NXX, %LCC)

Table 2-95 tbl_ne_opt_vlu Columns

Column_name	Type	Length	Nulls	Description
host_clli	char	80	0	The Host NE.

Table 2-95 (Cont.) tbl_ne_opt_vlu Columns

Column_name	Type	Length	Nulls	Description
param_type	varchar2	10	0	The type of provisioning parameter. The following values include: • H – Host parameters. • HN – Host/nxx parameters. • HU – Host/usoc parameters. • HUN – Host/usoc/nxx parameters. • HF – Host/feature parameters. • HP – Host pic conversion. • HUL – Host/usoc/lcc parameters. • HUNL – Host/usoc/nxx/lcc parameters. • RCCF – Remote activation CCF's for a 5ESS host. • NACT – No activate CCF's for a 5ESS host.
usoc	varchar2	10	1	The recent change Universal Service Order Code (USOC).
nxx	char	3	1	The NXX (Network Number Exchange).
lcc	varchar2	10	1	The line class code.
feat	varchar2	20	1	Switch feature name.
param_lbl	varchar2	80	0	Parameter label.
param_vlu	varchar2	30	1	Parameter value.

Table indexes:

```
host_clli, param_type, usoc, nxx, lcc, feat, param_lbl
```

tbl_ne_opt_vlu

This static table contains the option string to be sent to the Network Element.

Table 2-96 ttbl_ne_opt_vlu Columns

Column_name	Type	Length	Nulls	Description
tech	char	255	0	Technology of the Host Network Element.
sftwr_load	char	16	0	Software Version of the Host Network Element
asdl_cmd	char	80	0	The ASDL Command for which the option string needs to be sent to the NE.
mask_lbl	char	10	1	A masking label for the option format
opt_vlu	char	80	1	Option Value
ne_opt_vlu	char	80	0	The option value for the particular NE Type and Software Load.

Table indexes:

```
tech, sftwr_load, asdl_cmd, mask_lbl, opt_vlu
```

tbl_unid

This dynamic table provides a method of generating a serial field. You can manage unique IDs required by other tables. It is present in most user-created databases.

Table 2-97 tbl_unid Columns

Column_name	Type	Length	Nulls	Description
unid_type	varchar2	12	0	A unique code identifying the UNID type. This allows many different UNID values for different types of UNID.
unid	number	38	0	The UNID value for the type.
pad1	char	255	0	Padding to make a table row occupy a page. This is to reduce concurrence on the database data page by different database processes.
pad2	char	255	0	Same as pad1.
pad3	char	255	0	Same as pad1.
pad4	char	255	0	Same as pad1.

Table indexes:

```
unid_type
```

tbl_valid_len

This static table contains a set of line equipment numbers that ASAP accesses in a non-production environment. This table is checked from the CHECK_DATA Interpreter ASDL, when the ASAP_SYS environment variable does not equal PROD, and the Interpreter is not in loopback mode.

The CHECK_DATA program and tbl_valid_len database table are included as samples so that you may create your own data checking tables.

tbl_valid_len is used to safeguard line equipment numbers (other than those listed in this table) when system testing is being conducted on a production NE. If validation is requested and a particular LEN is not found in this check table, then the order is failed.

Table 2-98 tbl_valid_len Columns

Column_name	Type	Length	Nulls	Description
mcli	char	80	0	Remote NE of the LEN.
len	char	7	0	Valid testing Line Equipment Number (LEN).

Table indexes:

```
mcli, len
```

tbl_valid_nxx_line

This static table contains a set of telephone numbers that the ASAP system accesses in a non-production environment. This is a search table for the CHECK_DATA Interpreter, when the

ASAP_SYS environment variable does not equal PROD and the Interpreter is not in loopback mode.

The CHECK_DATA program and tbl_valid_nxx_line database table are included as samples so that you may create your own data checking tables.

This table is to safeguard telephone numbers (other than those listed in this table) when system testing is being conducted on a production NE. If validation is requested and a particular DN is not found in this check table, the ASDL is failed.

Table 2-99 tbl_valid_nxx_line Columns

Column_name	Type	Length	Nulls	Description
nxx	char	3	0	Valid test NXX.
line	char	4	0	Valid test line.

Table indexes:

```
nxx, line
```

3
Shared Libraries

This chapter contains information on the ASAP libraries that are shared by the provisioning (upstream) interface and the downstream interface. This chapter consists of the following sections:

- Control Configuration Interface
- XML JMX Interface

Control Configuration Interface

This section describes the functions for the Control subsystem.

The Control subsystem supports static table configuration. Use functions instead of SQL insert scripts to interface with the static configuration database tables.

The function-based interface reduces the dependency between administrators who configure the system and product developers who need to make changes to the static tables to support new functionality.

> **Note:**
>
> If you invoke an CSP_del_* function without parameters, all rows in the table are deleted.
>
> If you invoke the CSP_list_* functions without parameters, all rows are listed.

Interface Definitions

This section lists the syntax, descriptions, parameters, and results for Control configuration actions.

CSP_db_admin

This function purges all performance data that have been stored for more than a specified number of days. The default value of a_days is 3 days if it is not provided.

For more information about using functions, see "Oracle Examples."

Affected tables:

- tbl_alarm_log
- tbl_event_log
- tbl_process_info

Table 3-1 CSP_db_admin Parameters

Name	Description	Req'd	(I)nput/(O)utput
days	Specifies the age (in days) of log data to delete. All data older than the specified number of days is deleted.	Yes	I

CSP_del_alarm

This function deletes a system alarm code from tbl_system_alarm.

For more information about using functions, see "Oracle Examples."

Table 3-2 CSP_del_alarm Parameters

Name	Description	Req'd	(I)nput/ (O)utput
alarm_code	System alarm code identifier.	No	I

CSP_del_appl

This function deletes ASAP application registration information from the Control database (tbl_appl_proc).

For more information about using functions, see "Oracle Examples."

Table 3-3 CSP_del_appl Parameters

Name	Description	Req'd	(I)nput/ (O)utput
appl_cd	Logical name of the ASAP application server.	No	I

CSP_del_center

This function deletes an alarm center definition from the control database (tbl_alarm_center).

For more information about using functions, see "Oracle Examples."

Table 3-4 CSP_del_center Parameters

Name	Description	Req'd	(I)nput/ (O)utput
alarm_center	The alarm center to be deleted.	No	I

CSP_del_code

This function deletes an administration system code from the database (tbl_code_list).

For more information about using functions, see "Oracle Examples."

Table 3-5 CSP_del_code Parameters

Name	Description	Req'd	(I)nput/ (O)utput
code_type	Type of code. For example: “DB": database script related entry.	No	I
code	The code.	No	I
value	Value of the code.	No	I

CSP_del_component

This function deletes an ASAP component from tbl_component.

For more information about using functions, see "Oracle Examples."

Table 3-6 CSP_del_component Parameters

Name	Description	Req'd	(I)nput/ (O)utput
territory	The ASAP territory.	No	I
system	The ASAP system.	No	I
component	The ASAP component.	No	I

CSP_del_db_thresh

This function deletes a database threshold definition from tbl_db_threshold.

For more information about using functions, see "Oracle Examples."

Table 3-7 CSP_del_db_thresh Parameters

Name	Description	Req'd	(I)nput/ (O)utput
asap_sys	The ASAP environment (“TEST", “PROD", etc.).	No	I
db_name	The database name.	No	I

CSP_del_event

This function deletes an ASAP event type from the database (tbl_event_type).

For more information about using functions, see "Oracle Examples."

Table 3-8 CSP_del_event Parameters

Name	Description	Req'd	(I)nput/ (O)utput
event_type	The event type. For example, “ABNORMAL", “SYS_ERR", etc.	No	I

CSP_del_fs_thresh

This function deletes a file system threshold definition from tbl_fs_threshold.

For more information about using functions, see "Oracle Examples."

Table 3-9 CSP_del_fs_thresh Parameters

Name	Description	Req'd	(I)nput/ (O)utput
asap_sys	The ASAP environment (TEST, PROD, etc.).	No	I
file_system	The UNIX file system for which the threshold definition is to be deleted.	No	I

CSP_del_listener

This function deletes a listener entry from tbl_listeners.

For more information about using functions, see "Oracle Examples."

Table 3-10 CSP_del_listeners Parameters

Name	Description	Req'd	(I)nput/ (O)utput
srv_name	Name of the server that starts a socket listener. The SARM must start a socket listener to receive incoming Java SRP requests. For a Java-enabled NEP, this is the name of the NEP ($NEP). For the Java SRP, this column contains the SARM name.	Yes	I
listener_name	The name of the listener thread. For a Java-enabled NEP, the listener name describes the listener in the Java process that accepts interpreter requests from the C process. This listener name must always be $NEP_jlistener. For the Java SRP, observe the naming convention of “<Java SRP application name>_jsrplistener". This column is used by the Java SRP to retrieve the listener configurations.	Yes	I

CSP_del_nvp

This function deletes a name/value pair from the database (tbl_name_value_pair).

For more information about using functions, see "Oracle Examples."

Table 3-11 CSP_del_nvp Parameters

Name	Description	Req'd	(I)nput/ (O)utput
name	Name of the name/value pair.	No	I

CSP_get_listener

This function lists listener entries associated with an NEP (tbl_listeners).

For more information about using functions, see "Oracle Examples."

Table 3-12 CSP_get_listener Parameters

Name	Description	Req'd	(I)nput/ (O)utput
srv_name	Name of the server that starts a socket listener. The SARM must start a socket listener to receive incoming Java SRP requests. For a Java-enabled NEP, this is the name of the NEP ($NEP). For the Java SRP, this column contains the SARM name.	Yes	I

CSP_list_alarm

This function lists system alarms contained in tbl_system_alarm.

For more information about using functions, see "Oracle Examples."

Table 3-13 CSP_list_alarm Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
alarm_code	System alarm code.	No	I

Table 3-14 CSP_list_alarm Results

Name	Datatype	Description
alarm_code	TYP_code	The alarm code.
description	TYP_desc	Brief description of the system alarm.
alarm_level	TYP_alarm_level	Level of the alarm.
escalation_code	TYP_code	Escalation code of the alarm.
escalation_time	TYP_time	Escalation time.
auto_clear	TYP_yes_no	Determines whether the alarm must be automatically cleared.
route#_period	TYP_short	Interval in minutes for the alarm to be sent to the alarm center. # can be a number between 1 and 5 to designate up to five routings.
route#_start	TYP_time	Daily start time in minutes after midnight. # can be a number between 1 and 5 to designate up to five routings.

Table 3-14 (Cont.) CSP_list_alarm Results

Name	Datatype	Description
route#_end	TYP_time	Daily end time in minutes after midnight. # can be a number between 1 and 5 to designate up to five routings.
route#_center	TYP_code	Alarm center to route alarm to. # can be a number between 1 and 5 to designate up to five routings.

CSP_list_appl

This function lists ASAP application registration information for the specified appl_cd or all applications from the Control database (tbl_appl_proc).

For more information about using functions, see "Oracle Examples."

Table 3-15 CSP_list_appl Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
appl_cd	The logical name of the ASAP application server.	No	I

Table 3-16 CSP_list_appl Results

Name	Datatype	Description
start_seq	TYP_start_seq	Controls the sequence in which the applications are started. For example, certain client applications may be required to start before server applications, and other client applications after the server applications.
appl_type	TYP_appl_type	Specifies whether the ASAP application is an application server or a client application. Specify: • S – For server • C – For client • M – For primary control server • R – For remote secondary control server
appl_cd	TYP_code	ASAP logical client/server name, for example, SARM, NEP01, NEP02.
control_svr	TYP_code	The logical ASAP application control server that spawns this application and monitors its behavior.
description	TYP_desc	Brief description of the ASAP application.
diag_file	TYP_unix_file	The name of the diagnostics logfile to which diagnostic messages are written. This file is created in the $LOGDIR directory under a dated directory, for example, in the $LOGDIR/yymmdd format.
auto_start	TYP_yesno	An autostart flag that determines if the application is to be started automatically when ASAP starts.

Table 3-16 (Cont.) CSP_list_appl Results

Name	Datatype	Description
program	varchar(40)	The name of the UNIX program that runs to start the ASAP application. The UNIX program must reside in the $PROGRAMS directory and be executable.
diag_level	TYP_diag_level	The diagnostic level of the ASAP application. The diagnostic level is used to determine whether or not to log diagnostic information based on the diagnostic level of the ASC_diag() API function call.
isactive	TYP_yesno	A yes/no flag denoting whether the ASAP server is currently active.
last_start	datetime	The last start date and time of the ASAP server.
last_halt	datetime	The last halt or terminate date and time of the ASAP server.
last_abnormal	datetime	The last abnormal termination of the Control server.
svr_type	varchar(8)	This field defines the type of server.

CSP_list_center

This function lists alarm center definitions from the control database (tbl_alarm_center).

For more information about using functions, see "Oracle Examples."

Table 3-17 CSP_list_center Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
alarm_center	The alarm center to be deleted.	No	I

Table 3-18 CSP_list_center Results

Name	Datatype	Description
alarm_center	TYP_code	The unique code representing the alarm center.
control_prog	TYP_unix_file	The program to be run to communicate the alarm to the alarm center.
description	TYP_desc	Brief description of the alarm center.
opt#_type	TYP_option	First option to the control program, where # represents a value between 1 and 5.
opt#_value	TYP_opt_value	Argument to the first option, where # represents a value between 1 and 5.

CSP_list_code

This function lists Administration System code(s) from tbl_code_list.

If you invoke the function without any parameters, all rows in the table are listed.

For more information about using functions, see "Oracle Examples."

Table 3-19 CSP_list_code Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
code_type	Type of code. For example, DB for a database script related entry.	No	I
code	The code entry.	No	I
value	Value of the code.	No	I

Table 3-20 CSP_list_code Results

Name	Datatype	Description
code_type	TYP_code_type	Type of code.
code	TYP_code_text	Type of code.
value	TYP_code_value	Parameter value associated with the label.
code_desc	TYP_desc	Brief description of the code.
parm#	TYP_code_parm	General purpose parameter, where # represents a value between 1 and 4.

CSP_list_component

This function lists ASAP components contained in tbl_component.

If you invoke the function without any parameters, all rows in the table are listed.

For more information about using functions, see "Oracle Examples."

Table 3-21 CSP_list_component Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
territory	Identifies the ASAP territory.	No	I
system	Identifies the ASAP system.	No	I
component	Identifies the ASAP component.	No	I

Table 3-22 CSP_list_component Results

Name	Datatype	Description
territory	varchar(20)	ASAP territory.
system	varchar(20)	ASAP system.
component	varchar(40)	ASAP component.

CSP_list_db_thresh

This function lists database threshold definition(s) contained in tbl_db_threshold.

If you invoke the function without any parameters, all rows in the table are listed.

For more information about using functions, see "Oracle Examples."

Table 3-23 CSP_list_db_thresh Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
asap_sys	The ASAP environment (for example, TEST, PROD etc.)	No	I
db_name	Database name.	No	I

Table 3-24 CSP_list_db_thresh Results

Name	Datatype	Description
asap_sys	TYP_code	ASAP environment.
db_name	varchar(80) TYP_desc	Database name.
data_threshold	int	Database threshold, in Mb.
tran_threshold	int	Transaction log threshold.
data_event	TYP_code	Event issued if the database threshold is exceeded.
tran_event	TYP_code	Event to be issued if the transaction log threshold is exceeded.

CSP_list_event

This function lists ASAP event definitions contained in tbl_event_type.

If you invoke the function without any parameters, all rows in the table are listed.

For more information about using functions, see "Oracle Examples."

Table 3-25 CSP_list_event Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
event_type	Event type. For example, the core uses some of the following event types: ABNORMAL, SYS_ERR, etc.	No	I

Table 3-26 CSP_list_event Results

Name	Datatype	Description
event_type	TYP_event	The event type. The core module includes some of the following event types: ABNORMAL, SYS_ERR, etc.
description	varchar(40) TYP_desc	Brief description of the event.
alarm_code	TYP_code	The alarm code associated with the event.
alarm_action	TYP_alarm_action	Specifies whether the alarm is enabled or disabled.

CSP_list_fs_thresh

This function lists file system threshold definitions contained in tbl_fs_threshold.

If you invoke the function without any parameters, all rows in the table are listed.

For more information about using functions, see "Oracle Examples."

Table 3-27 CSP_list_fs_thresh Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
asap_sys	The ASAP environment (TEST, PROD, etc.).	No	I
file_system	The UNIX file system for which the threshold definition is to be deleted.	No	I

Table 3-28 CSP_list_fs_thresh Results

Name	Datatype	Description
asap_sys	TYP_code	The ASAP environment.
file_system	varchar(100) TYP_desc	File system name.
full_threshold	int	File system full threshold.
full_event	TYP_code	Event to be generated if the full threshold is exceeded.

CSP_list_nvp

This function lists name/value pairs from the database (tbl_name_value_pair).

If you invoke the function without any parameters, all rows in the table are deleted.

For more information about using functions, see "Oracle Examples."

Table 3-29 CSP_list_nvp Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O

Table 3-29 (Cont.) CSP_list_nvp Parameters

Name	Description	Req'd	(I)nput/ (O)utput
name	Name of the name-value pair.	No	I

Table 3-30 CSP_list_nvp Results

Name	Datatype	Description
name	varchar(40)	Name of the parameter.
value	int	Parameter value associated with the label.

CSP_new_alarm

This function defines a system alarm which may be generated by ASAP system events. This includes the start time, interval, and end time for the alarm.

This function populates tbl_system_alarm.

Example:

The following example creates an alarm for the abnormal termination of an application process to the ADMINPGR center that is continuous on a five minute period any time of the day, type the following:

```
var retval number;
exec :retval := CSP_new_alarm ('ABNORMAL', 'Abnormal process termination', 'CRITICAL',
'',NULL, 'N', 5, 0, 1440, 'ADMINPGR');

print retval;
```

For more information about using functions, see "Oracle Examples."

Table 3-31 CSP_new_alarm Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
alarm_code	System alarm code.	Yes	I
description	Brief description of the system alarm.	Yes	I
alarm_level	The level of the alarm, for example, MAJOR, MINOR, CRITICAL.	Yes	I
escalation_code	The escalation code for the alarm if the alarm is not corrected within the escalation time (defined next).	Yes	I
escalation_time	Time after the alarm was raised that the escalation of the alarm must take place.	Yes	I
auto_clear	Flag that determines if the alarm should be automatically cleared upon generation.	Yes	I

Table 3-31 (Cont.) CSP_new_alarm Parameters

Name	Description	Req'd	(I)nput/ (O)utput
route#_period	Interval in minutes for the alarm to be sent to the alarm center, where # represents a value between 1 and 5.	Yes (if # = 1), No (if # = 2 to 5)	I
route#_start	The daily start time in minutes since midnight for alarms to go to this alarm center, where # represents a value between 1 and 5.	Yes (if # = 1), No (if # = 2 to 5)	I
route#_end	The daily end time in minutes since midnight, where # represents a value between 1 and 5.	Yes (if # = 1), No (if # = 2 to 5)	I
route#_center	Alarm center to route alarm to, where # represents a value between 1 and 5.	Yes (if # = 1), No (if # = 2 to 5)	I

CSP_new_appl

This function defines a new ASAP client or server application in tbl_appl_proc.

For more information about using functions, see "Oracle Examples."

Table 3-32 CSP_new_appl Parameters

Name	Description	Req'd	(I)nput/ (O)utput
start_seq	Specifies the ASAP startup sequence. This determines the sequence in which applications are started.	Yes	I
appl_type	Specifies the ASAP application server: • S – For server • C – For client • M – For primary control server • R – For remote secondary control server	Yes	I
appl_cd	The logical ASAP application code, for example, SARM, NEP01, NEP02.	Yes	I
control_svr	Logical ASAP Control server.	Yes	I
auto_start	An autostart flag.	Yes	I
program	The name of the UNIX program to start the ASAP application.	Yes	I
diag_level	The diagnostic level of the ASAP application. This is used to determine whether or not to log certain diagnostic information based on the diagnostic level of the **ASC_diag()** API function call. See the associated rule for possible values.	Yes	I
diag_file	The name of the diagnostics file in which diagnostic messages are to be placed. This file is created in the $LOGDIR directory under a dated directory, for example, using the $LOGDIR/yymmdd format.	Yes	I
description	A description of the ASAP application.	Yes	I

Table 3-32 (Cont.) CSP_new_appl Parameters

Name	Description	Req'd	(I)nput/ (O)utput
svr_type	ASAP server type. Possible values include: • CTRL – Control server • MASTER – primary Control server (must be only one per system) • SARM – SARM server • SRP – SRP server • NEP – NEP server • OTHER	No	I

Example:

To configure ASAP to manage and monitor this processes, enter the following commands. This example assumes this processes is run automatically at startup and the **Low** level diagnostics are active.

```
var retval number;
exec :retval := CSP_new_appl (4, 'S', 'NEPAXE', 'CONTROL2','Y', 'LOW', 'NEPAXE.diag',
'NEP for AXE Switches')
```

Once the control servers are started on their respective machines, you can start and shut down the application processes automatically from the primary system.

CSP_new_center

This function defines an alarm center to which alarm notifications are sent by the Control server. This function populates tbl_alarm_center.

Syntax:

```
var retval number;
exec :retval :=  CSP_new_center ('alarm_center', 'control_prog', ['description'], ...);
```

Example:

In the following example, **admin.sh** is the ADMIN center and **adminpg** is the ADMINPGR center. You must place the final versions of these programs in the ASAP programs directory and identify them using the environment variable **$PROGRAMS**.

```
var retval number;
exec :retval :=  CSP_new_center ('ADMIN', 'admin.sh', 'General Admin. Center');
```

For more information about using functions, see "Oracle Examples."

Table 3-33 CSP_new_center Parameters

Name	Description	Req'd	(I)nput/ (O)utput
alarm_center	A unique code representing the alarm center.	Yes	I
control_prog	The program to be run to communicate the alarm to the alarm center.	Yes	I

Table 3-33 (Cont.) CSP_new_center Parameters

Name	Description	Req'd	(I)nput/ (O)utput
description	A brief description of the alarm center or control program.	No	I
opt#_type	Option name to be passed to the control program, where # represents a value between 1 and 5.	No	I
opt#_value	Value of the option, where # represents a value between 1 and 5.	No	I

CSP_new_code

This function populates tbl_code_list with core or custom code used by ASAP. For instance, this function to identify codes that track the cartridges deployed within ASAP.

For more information about using functions, see "Oracle Examples."

Table 3-34 CSP_new_code Parameters

Name	Description	Req'd	(I)nput/ (O)utput
code_type	Type of code. For example, DB for database script related entry.	Yes	I
code	The code.	Yes	I
value	Value of the code.	Yes	I
code_desc	Brief description of the code.	No	I
parm#	General purpose parameter, where # can be a number between 1 and 4.	No	I

CSP_new_component

This function defines an ASAP component in a territory and adds it to database table tbl_component.

For more information about using functions, see "Oracle Examples."

Table 3-35 CSP_new_component Parameters

Name	Description	Req'd	(I)nput/ (O)utput
territory	Identifies the ASAP territory.	Yes	I
system	Identifies the ASAP system.	Yes	I
component	Identifies an ASAP system component within a territory and system. For example, SRP, SARM, etc.	Yes	I

CSP_new_db_thresh

This function defines database and/or transaction log thresholds to be used by the Control server, and writes the information to tbl_db_threshold. The Control server monitors the database/transaction log size and issues the appropriate data event/transaction event when the threshold is exceeded.

Syntax:

```
var retval number;
exec :retval :=  CSP_new_db_thresh ('asap_sys', 'db_name', 'db_threshold',
'tran_threshold', 'data_event', 'tran_event')
```

Example:

```
var retval number;
exec :retval :=  CSP_new_db_thresh ('PROD', 'SDB_P01_asap', 80, 20, 'DB2FULL',
'TRANFULL';
```

in this example, the database threshold is set to 80 percent. If the database becomes more than 80 percent full, the DB2FULL system event is issued. This command also sets the transaction log threshold for the database. When the transaction log for the database exceeds 20 MB, the TRANFULL system event is issued.

Database thresholds must be defined in the component table.

For more information about using functions, see "Oracle Examples."

Table 3-36 CSP_new_db_thresh Parameters

Name	Description	Req'd	(I)nput/ (O)utput
asap_sys	The ASAP environment (for example, TEST, PROD, etc.).	Yes	I
db_name	Database name.	Yes	I
db_threshold	Database threshold, in Mb.	Yes	I
tran_threshold	Transaction log threshold.	Yes	I
data_event	Data event.	Yes	I
tran_event	Transaction log event.	Yes	I

CSP_new_event

This function defines an “event type" within ASAP, and optionally, the associated “alarm code" that is associated with the event. System events can be generated when an error condition is encountered in ASAP. CSP_new_event populates tbl_event_type.

For more information about using functions, see "Oracle Examples."

Table 3-37 CSP_new_event Parameters

Name	Description	Req'd	(I)nput/ (O)utput
event_type	Event type. For example, the core uses some of the following event types: ABNORMAL, SYS_ERR, etc.	Yes	I
description	A brief description of the ASDL command.	No	I
alarm_code	The system alarm code associated with the event. If NULL, no alarm will be generated and only the database log entry will be created.	No	I
alarm_action	Specifies the alarm action, and may specify whether the alarm is presently enabled or disabled.	No	I
notify_aims	This is not currently implemented.		

Syntax:

```
var retval number;
exec :retval :=  CSP_new_event ('event_type', ['description'], ['alarm_code'],
[“alarm_action'];
```

Example:

The following example shows an ABNORMAL system event mapped to the ABNORMAL alarm.

```
var retval number;
exec :retval :=  CSP_new_event ('ABNORMAL', 'Abnormal Process Termination Event',
'ABNORMAL', 'E';
```

CSP_new_fs_thresh

This function defines a file system threshold to be used by the Control server. The Control server monitors the file system size and if the threshold is exceeded, the appropriate system event is generated. File system threshold information is stored in tbl_fs_threshold.

For more information about using functions, see "Oracle Examples."

CSP_new_listener

This function adds a listener entry to tbl_listeners. You must configure this table to allow the SARM to start up socket listeners for incoming SRP requests. As well, every Java-enabled NEP must maintain a dedicated connection to its JInterpreter.

For more information about using functions, see "Oracle Examples."

Table 3-38 CSP_new_listener Parameters

Name	Description	Req'd	(I)nput/ (O)utput
srv_name	Name of the server that starts a socket listener. The SARM must start a socket listener to receive incoming Java SRP requests. For a Java-enabled NEP, this is the name of the NEP ($NEP). For the Java SRP, this column contains the SARM name.	Yes	I
host_name	The host name or the IP address on which the server application resides. For the JInterpreter, this value must always be localhost. For the Java SRP, the host_name identifies the location of the SARM.	Yes	I
listener_name	The name of the listener thread. For a Java-enabled NEP, the listener name describes the listener in the Java process that accepts interpreter requests from the C process. This listener name must always be $NEP_jlistener. For the Java SRP, observe the naming convention of “<Java SRP application name>_jsrplistener". This column is used by the Java SRP to retrieve the listener configurations.	Yes	I
port	A free port on which the server can start the socket listener.	Yes	I

CSP_new_nvp

This function defines parameters (name value pairs) that are required to maintain the control database. Typical parameters include “audit trail window", “log retention window", etc. This information is stored in tbl_name_value_pair.

For more information about using functions, see "Oracle Examples."

Table 3-39 CSP_new_nvp Parameters

Name	Description	Req'd	(I)nput/ (O)utput
name	Name of the parameter.	Yes	I
value	Value of the parameter.	Yes	I

XML JMX Interface

For overview information on the JMX interface, refer to the *ASAP Server Configuration Guide*. For complete reference information on the JMX interface, refer to the *ASAP Online Reference*.

4
Provisioning Interfaces

This chapter describes the following provisioning (upstream) interfaces:

- SARM Configuration Interface

SARM Configuration Interface

This section covers the functions for the SARM and includes the following subsections:

- Static Table Configuration
- Error Management
- Switch Blackout Processing
- Stop Work Order Interface
- Localizing International Messages

Static Table Configuration

To interface to the static configuration database tables, use the function-based interface instead of SQL insert scripts.

The function-based interface reduces the dependency between administrators who configure the system and product developers who need to make changes to the static tables to support new functionality.

This section lists the syntax, descriptions, parameters, and results for the SARM configurations and includes the delete, list, and new procedures interface definitions.

SSP_db_admin

This function performs customer-specific database administration functions, such as purging the database of completed orders, and removing orphaned definitions from the SARM database.

For more information on this function and database purging, refer to the *ASAP System Administrator's Guide*.

For more information about using functions, see "Oracle Examples."

Affected tables:

- tbl_wrk_ord
- tbl_asap_stats
- tbl_info_parm
- tbl_srq
- tbl_srq_csdl
- 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

Table 4-1 SSP_db_admin Parameters

Name	Description	Req'd	(I)nput/ (O)utput
days	Specifies the age (in days) of work orders to delete. All completed work orders older than the specified number of days are deleted.	Yes	I

SSP_gather_asap_stats

Gathers statistics for objects in the database (tbl_wrk_ord). The information includes the distribution of data, the number of rows in the table and other important statistics. Statistics gathering is governed by the following parameters in ASAP.cfg:

- GATHER_STATS
- GATHER_STATS_PROC
- DB_PCT_ANALYZE
- DB_PCT_ANALYZE_IDX
- GATHER_DEGREE
- DB_ADMIN_TIME

For more information about using functions, see "Oracle Examples."

Syntax:

```
SSP_gather_asap_stats(
a_tab_estimate_pct number,
a_ind_estimate_pct number,
a_degree number)
```

Table 4-2 SSP_del_asdl_defn Parameters

Name	Description	Req'd	(I)nput/ (O)utput
a_tab_estimate_pct number	This parameter applies to Oracle only. It is used to update statistics on all user-defined tables. The updates are done when the database administrations tasks are performed. (See also DB_ADMIN_TIME.) This parameter is used to optimize the database query performance. The Oracle SQL statement is “analyze table table_name estimate statistics sample DB_PCT_ANALYZE percent". For further information, refer to “Analyze Command" in the Oracle SQL Reference manual.	Yes	I

Table 4-2 (Cont.) SSP_del_asdl_defn Parameters

Name	Description	Req'd	(I)nput/ (O)utput
a_ind_estimate_pct number	Percentage of the index to analyze when gathering statistics.	Yes	I
a_degree number	Degree of parallelism when gathering statistics. The degree parameter can take the value of auto_degree. When you specify the auto_degree, Oracle determines the degree of parallelism automatically. It will be either 1 (Running serially) or default_degree (the system default value based on number of CPUs and initialization parameters), according to the size of the object.	Yes	I

SSP_del_asdl_defn

This function deletes ASDL definitions from tbl_asdl_config. Wildcards are permitted.

> **Note:**
>
> If you do not specify an ASDL command, all ASDL command definitions are deleted from the configuration records.

Syntax:

```
var retval number;
exec :retval :=  SSP_del_asdl_defn (['asdl_cmd'])
```

Example:

```
var rc refcursor;
var retval number;
exec :retval :=  SSP_del_asdl_defn (:rc,'M-CREATE_SINGLE_LINE_ACCESS');
```

This example removes the **M-CREATE_SINGLE_LINE_ACCESS** configuration record from the static configuration tables.

For more information about using functions, see "Oracle Examples."

Table 4-3 SSP_del_asdl_defn Parameters

Name	Description	Req'd	(I)nput/ (O)utput
asdl_cmd	The ASDL command to be deleted.	No	I

SSP_del_asdl_map

This function deletes ASDL mappings from tbl_nep_asdl_prog. The mapping is based on the technology and software load.

For more information about using functions, see "Oracle Examples."

Syntax:

```
var retval number;
exec :retval :=  SSP_del_asdl_map ['tech'] [, 'sftwr_load'] [, 'asdl_cmd']
```

Table 4-4 SSP_del_asdl_map Parameters

Name	Description	Req'd	(I)nput/ (O)utput
tech	The technology type of NE or SRP with which the Interpreter is to interact.	No	I
sftwr_load	The version of the software currently running on the NEP or SRP.	No	I
asdl_cmd	The ASDL command.	No	I

SSP_del_asdl_parm

This function deletes an ASDL parameter from tbl_asdl_parm.

> **Note:**
>
> If you do not enter a sequence number, all parameters associated with this ASDL command are deleted.

For more information about using functions, see "Oracle Examples."

This example removes the ASDL command parameter with the sequence number 4 from the configuration table for the command **M-CREATE_SINGLE_LINE_ACCESS**.

Table 4-5 SSP_del_asdl_parm Parameters

Name	Description	Req'd	(I)nput/ (O)utput
asdl_cmd	The ASDL command.	No	I
parm_seq_no	The parameter sequence number.	No	I

SSP_del_clli_map

This function deletes a remote CLLI to host CLLI mapping in tbl_clli_route.

For more information about using functions, see "Oracle Examples."

Table 4-6 SSP_del_clli_map Parameters

Name	Description	Req'd	(I)nput/ (O)utput
mach_clli	The remote NE.	No	I
asdl_cmd	ASDL command.	No	I

SSP_del_comm_param

This function deletes communication parameter information from tbl_comm_param.

For more information about using functions, see "Oracle Examples."

Table 4-7 SSP_del_comm_param Parameters

Name	Description	Req'd	(I)nput/ (O)utput
dev_type	The device type. Choose the type of connection from the following: • D – Serial Port Dialup • F – TCP/IP FTP Connection • G – Generic Terminal Based Connection • H – Serial Port Hardwired • M – Generic Message Based Connection • S – TCP/IP Socket Connection • T – TCP/IP Telnet Connection • W – LDAP Connection • C – CORBA	No	I
host	Host CLLI. Set to COMMON_HOST_CFG or the host CLLI associated with the command processor. For a common host, the parameter value is the default value, otherwise, it is host-specific.	No	I
device	The physical or logical device name. Set to COMMON_DEVICE_CFG or the device associated with the command processor. For a common device, the parameter value is the default value, otherwise, it is device-specific.	No	I
param_label	Specifies the communication parameter label.	No	I
param_value	Specifies the communication parameter value.	No	I
param_desc	Specifies the communication parameter description.	No	I

SSP_del_csdl_asdl

This function deletes a CSDL-to-ASDL mapping definition from tbl_csdl_asdl.

> **Note:**
>
> If you do not specify an ASDL command sequence number, all mapping relationships for the specified CSDL command are removed from the configuration tables. If you do not specify a CSDL command, all mapping relationships are removed.

For more information about using functions, see "Oracle Examples."

Table 4-8 SSP_del_csdl_asdl Parameters

Name	Description	Req'd	(I)nput/ (O)utput
csdl_cmd	The CSDL command.	No	I
asdl_seq_no	The sequence number of the ASDL command.	No	I

SSP_del_csdl_defn

This function deletes CSDL definitions from tbl_csdl_config.

> **Note:**
>
> If you do not specify a CSDL command, all CSDL command definitions are removed.

Syntax:

```
var retval number;
exec :retval :=  SSP_del_csdl_defn ['csdl_cmd']
```

Example:

```
var retval number;
exec :retval :=  SSP_del_csdl_defn 'M-CREATE_BUS_LINE'
```

The configuration record for M-CREATE_BUS_LINE is removed from the configuration tables.

For more information about using functions, see "Oracle Examples."

Table 4-9 SSP_del_csdl_defn Parameters

Name	Description	Req'd	(I)nput/ (O)utput
csdl_cmd	The CSDL command.	No	I

SSP_del_dn_map

This function deletes a directory number mapping from tbl_nep_rte_asdl_nxx.

For more information about using functions, see "Oracle Examples."

Table 4-10 SSP_del_dn_map Parameters

Name	Description	Req'd	(I)nput/ (O)utput
asdl_cmd	The ASDL command identifier.	No	I
npa	The Numbering Plan Area code.	No	I
nxx	The Central Office code.	No	I
from_line	The lowest line number in the range.	No	I

Table 4-10 (Cont.) SSP_del_dn_map Parameters

Name	Description	Req'd	(I)nput/ (O)utput
to_line	The highest line number in the range.	No	I

SSP_del_id_routing

This function deletes a host NE and the ID_ROUTING mapping record from tbl_id_routing. Use this function when routing by ID_ROUTING is used.

For more information about using functions, see "Oracle Examples."

Table 4-11 SSP_del_id_routing Parameters

Name	Description	Req'd	(I)nput/ (O)utput
host_clli	The host NE identifier.	Yes	I
asdl_cmd	The ASDL command.	Yes	I
id_routing_from	The starting point of a range of ID_ROUTING.	Yes	I
id_routing_to	The end point of a range of ID_ROUTING.	Yes	I

SSP_del_intl_msg

This function deletes an international message record from the tbl_msg_convert.

If you do not specify a message identifier, all messages with the specified language code are deleted.

For more information about using functions, see "Oracle Examples."

Table 4-12 SSP_del_intl_msg Parameters

Name	Description	Req'd	(I)nput/ (O)utput
lang_cd	The language code.	No	I
msg_id	The unique message identifier for the message to be removed from the SARM database.	No	I

The following example shows how to delete an international message:

```
var retval number;
exec :retval :=  SSP_del_intl_msg 'USA', 1
```

This deletes American English message 1 from the SARM database.

SSP_del_ne_host

This function deletes a host NE definition from tbl_host_clli.

For more information about using functions, see "Oracle Examples."

Table 4-13 SSP_del_ne_host Parameters

Name	Description	Req'd	(I)nput/ (O)utput
host_clli	The host NE identifier of an NE or SRP.	No	I

SSP_del_nep

This function deletes an NEP secondary pool definition from tbl_nep.

For more information about using functions, see "Oracle Examples."

Table 4-14 SSP_del_nep Parameters

Name	Description	Req'd	(I)nput/ (O)utput
nep_svr_cd	The NEP managing the secondary pool of devices.	No	I

SSP_del_nep_program

This function deletes the actions based on the specified program name and/or line number from tbl_nep_program.

If the line number is not supplied, all actions with positive line numbers are removed.

For more information about using functions, see "Oracle Examples."

Table 4-15 SSP_del_nep_program Parameters

Name	Description	Req'd	(I)nput/ (O)utput
program	The program identifier.	Yes	I
line_no	The line number to delete. If set to NULL, all lines of the program are deleted.	No	I

SSP_del_net_elem

This function deletes an NE definition for an NEP from the SARM database.

This function deletes a network elements from tbl_ne_config.

For more information about using functions, see "Oracle Examples."

Table 4-16 SSP_del_net_elem Parameters

Name	Description	Req'd	(I)nput/ (O)utput
host_clli	The host NE identifier of an NE or SRP.	No	I
nep_svr_cd	The logical name of the NEP server that connects to the host NE.	No	I

SSP_del_resource

This function deletes a device from **tbl_resource_pool.**

For more information about using functions, see "Oracle Examples."

Table 4-17 SSP_del_resource Parameters

Name	Description	Req'd	(I)nput/ (O)utput
asap_sys	The ASAP environment (TEST, PROD, and so on).	No	I
device	The physical or logical device name.	No	I

SSP_del_srp

This function deletes an SRP definition from the tbl_asap_srp.

For more information about using functions, see "Oracle Examples."

Table 4-18 SSP_del_srp Parameters

Name	Description	Req'd	(I)nput/ (O)utput
srp_id	The logical SRP name.	No	I

SSP_del_stat_text

This function deletes static text labels used in the OCA GUI client from tbl_stat_text.

For more information about using functions, see "Oracle Examples."

Table 4-19 SSP_del_stat_text Parameters

Name	Description	Req'd	(I)nput/ (O)utput
stat_id	The logical group of static text messages such as WO_STATE. If this is not specified, all entries in tbl_stat_text are deleted.	No	I
status	The integer identifier for member of logical grouping.	No	I
code	The string identifier for member of logical grouping.	No	I

SSP_del_user_err_threshold

This function deletes a user-defined error threshold or set of thresholds from tbl_user_err_threshold.

For more information about using functions, see "Oracle Examples."

Table 4-20 SSP_del_user_err_threshold Parameters

Name	Description	Req'd	(I)nput/ (O)utput
host	The host NE identifier of an NE or SRP.	No	I
asdl_cmd	The ASDL command.	No	I
user_type	The user-defined error type.	No	I

SSP_del_userid

This function deletes a user ID from tbl_uid_pwd.

For more information about using functions, see "Oracle Examples."

Table 4-21 SSP_del_userid Parameters

Name	Description	Req'd	(I)nput/ (O)utput
uid	The user ID.	No	I

SSP_get_async_ne

This function returns the names of all the NEs that have a ASYNC_CONN communication parameter defined with a value of TRUE or FALSE. The existence of a parameter labeled ASYNC_CONN indicates that the NE has an asynchronous interface. The parameter value of TRUE or FALSE indicates whether the NEP server should establish an NE element connection at NEP start-up. This function has no input parameters.

For more information about using functions, see "Oracle Examples."

Table 4-22 SSP_get_async_ne Parameters

Name	Description	Req'd	(I)nput/ (O)utput
host	The host name of the NE having an ASYNCH_CONN communication parameter defined.	No	O
param_value	Value of the ASYNCH_CONN parameter - either TRUE or FALSE. Indicates whether the NEP server should establish an NE element connection at NEP start-up.	No	O

SSP_get_user_routing

This function returns a Host NE (host_clli) that is used to route the ASDL from tbl_user_routing. You must write this function and the associated database table based on the pre-defined interfaces and your own routing algorithm when using a user-defined routing.

For more information about using functions, see "Oracle Examples."

Table 4-23 SSP_get_user_routing Parameters

Name	Description	Req'd	(I)nput/ (O)utput
user_routing	The USER_ROUTING defined as a work order parameter.	No	I
asdl_cmd	The ASDL command.	No	I
host_clli	The host NE identifier to be routed. If it fails to find the host NE associated with the USER_ROUTING, it returns NULL.	No	O

SSP_list_asdl

This function retrieves ASDL-related information from tbl_asdl_config, tbl_asdl_parm, tbl_nep_asdl_prog.

For more information about using functions, see "Oracle Examples."

Table 4-24 SSP_list_asdl Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
RC2	Oracle Database Ref Cursor.	Yes	I/O
RC3	Oracle Database Ref Cursor.	Yes	I/O
asdl	The name of the ASDL.	No	I

The shaded groupings below indicate that multiple result sets are returned: in this case, three sets of data.

For more information about using functions, see "Oracle Examples."

Table 4-25 SSP_list_asdl Results

Name	Datatype	Description
reverse_asdl	TYP_asdl_cmd	reverse ASDL command
ignore_rollback	TYP_yes_no	ignore rollback flag
route_flag	TYP_route	ASDL routing flag
description	varchar(40)	ASDL command description
asdl_lbl	TYP_parm_lbl	ASDL parameter label
csdl_lbl	TYP_parm_lbl	CSDL parameter label
param_typ	TYP_parm_typ	ASDL parameter type
default_vlu	TYP_parm_vlu	ASDL parameter default value
tech	TYP_tech	technology
sftwr_load	TYP_load	software loads
program	TYP_program	ASDL program

SSP_list_asdl_defn

This function lists all or specific ASDL definitions from the SARM database (tbl_asdl_config, tbl_csdl_asdl). You can use wildcards in this procedure. If you do not specify a parameter, all ASDL definitions are returned.

Syntax:

```
var rc refcursor;
var retval number;
exec :retval :=  SSP_list_asdl_defn (:rc[,'asdl_cmd']);
print rc;
```

Example:

```
var rc refcursor;
var retval number;
exec :retval := SSP_list_asdl_defn (:rc,'M-CREATE_SINGLE_LINE_ACCESS');
print rc;
```

The following example lists the information for M-SINGLE_LINE_ACCESS:

```
asdl_cmd                      reverse_asdl                  ignore_rollback route_flag
description
-------- ---------- ----------------------------------------
M-CREATE_SINGLE_LINE_ACCESS  M-DELETE_SINGLE_LINE_ACCESS    N               B
Create a single-line access service.
(1 row affected, return status = 0)
```

For more information about using functions, see "Oracle Examples."

Table 4-26 SSP_list_asdl_defn Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
asdl_cmd	The ASDL command identifier.	No	I

Table 4-27 SSP_list_asdl_defn Results

Name	Datatype	Description
asdl_cmd	TYP_asdl_cmd	ASDL command.
reverse_asdl	TYP_asdl_cmd	Reverse ASDL command.
ignore_rollback	TYP_yes_no	Ignore rollback flag.
route_flag	TYP_route	ASDL routing flag.
description	varchar(40)	ASDL command description.

SSP_list_asdl_map

This function lists ASDL mappings according to various criteria. All parameters take wildcards. This function retrieves asdl mapping information from tbl_nep_asdl_prog.

Syntax:

```
var rc1 refcurson;
var retval number;
exec :retval :=  SSP_list_asdl_map (:RC1) [, 'tech'] [, 'sftwr_load'] [, 'asdl_cmd'] [,
'interpreter_type']
```

For more information about using functions, see "Oracle Examples."

Table 4-28 SSP_list_asdl_map Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
tech	The technology type of the NE or SRP with which the Interpreter is to interact.	No	I
sftwr_load	The version of the software currently running on the NEP or SRP.	No	I
asdl_cmd	The ASDL command. Wildcards are accepted.	No	I
program	The SASDLprogram.	No	I
interpreter_type	A value of **J** indicates a JInterpreter.	No	I

Table 4-29 SSP_list_asdl_map Results

Name	Datatype	Description
tech	TYP_tech	Technology.
sftwr_load	TYP_load	Software loads.
asdl_cmd	TYP_asdl_cmd	ASDL command.
program	TYP_program	ASDL program.

SSP_list_asdl_parm

This function lists ASDL parameters from the SARM database (tbl_asdl_parm) by ASDL command name and/or ASDL parameter label. Wildcards are allowed.

Syntax:

```
var rc1 refcurson;
var retval number;
exec :retval :=  SSP_list_asdl_parm (:RC1) [, 'asdl_cmd'] [, 'asdl_parm_lbl']
```

Example:

```
var rc refcurson;
var retval number;
exec :retval :=  SSP_list_asdl_parm 'M-CREATE_SINGLE_LINE_ACCESS', 'NPA'
```

This example retrieves configuration information for the NPA parameter for the ASDL command **M-CREATE_SINGLE_LINE_ACCESS**.

For more information about using functions, see "Oracle Examples."

Table 4-30 SSP_list_asdl_parm Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
asdl_cmd	The ASDL command.	No	I
asdl_parm_lbl	The ASDL parameter label.	No	I

Table 4-31 SSP_list_asdl_parm Results

Name	Datatype	Description
asdl_cmd	TYP_asdl_cmd	ASDL command.
parm_seq_no	TYP_seq_no	ASDL parameter number.
asdl_lbl	TYP_parm_lbl	ASDL parameter label.
csdl_lbl	TYP_parm_lbl	CSDL parameter label.
default_vlu	TYP_parm_vlu	ASDL parameter default value.
param_typ	TYP_parm_typ	ASDL parameter type.

SSP_list_clli_map

This function lists remote CLLI-to-Host CLLI mapping definitions that are contained in tbl_clli_route. All parameters take wildcards.

For more information about using functions, see "Oracle Examples."

Table 4-32 SSP_list_clli_map Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
mach_clli	The remote NE.	No	I
host_clli	The host NE identifier of an NE or SRP.	No	I

Table 4-33 SSP_list_clli_map Results

Name	Datatype	Description
mach_clli	TYP_clli	Remote NE.
host_clli	TYP_clli	Host NE.
asdl_cmd	TYP_asdl_cmd	ASDL command identifier.

SSP_list_comm_param

This function lists communication parameters based on dev_type, host, device, param_label, or for all of them. This information is retrieved from tbl_comm_param.

For more information about using functions, see "Oracle Examples".

Table 4-34 SSP_list_comm_param Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
dev_type	The device type. Choose from the following types of connections: • D – Serial Port Dialup • F – TCP/IP FTP Connection • G – Generic Terminal Based Connection • H – Serial Port Hardwired • M – Generic Message Based Connection • S – TCP/IP Socket Connection • T – TCP/IP Telnet Connection • W – LDAP Connection • C – CORBA	No	I
host	The host CLLI. It is set to COMMON_HOST_CFG or the host CLLI associated with the command processor. If a common host, the parameter value is the default value, otherwise, it is host-specific.	No	I
device	The physical or logical device (port) name. Set to COMMON_DEVICE_CFG or the device associated with the command processor. If a common device, the parameter value is the default value, otherwise, it is device-specific.	No	I
param_label	The communication parameter label.	No	I

Table 4-35 SSP_list_comm_param Results

Name	Datatype	Description
dev_type	TYP_dev_type	Type of connection. Choose from: • D – Serial Port Dialup • F – TCP/IP FTP Connection • G – Generic Terminal Based Connection • H – Serial Port Hardwired • M – Generic Message Based Connection • S – TCP/IP Socket Connection • T – TCP/IP Telnet Connection • W – LDAP Connection
host	TYP_clli	Host CLLI. It is set to COMMON_HOST_CFG or the host CLLI associated with the command processor. If a common host, the parameter value is the default value, otherwise, it is host-specific.
device	TYP_device	The physical or logical device name. Set to COMMON_DEVICE_CFG or the device associated with the command processor. If a common device, the parameter value is the default value, otherwise, it is device-specific.
param_label	TYP_parm_lbl	Specifies the communication parameter label.
param_value	TYP_perf_parm_vlu	Specifies the communication parameter value.
param_desc	TYP_parm_desc	Specifies the communication parameter description.

SSP_list_csdl

This function retrieves CSDL-related information from tables tbl_csdl_config and tbl_csdl_asdl.

For more information about using functions, see "Oracle Examples."

Table 4-36 SSP_list_csdl Parameters

Name	Description	Req'd	(I)nput/(O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
RC2	Oracle Database Ref Cursor.	Yes	I/O
csdl	The name of the CSDL.	Yes	I

Table 4-37 SSP_list_csdl Results

Name	Datatype	Description
asdl_cmd	TYP_asdl_cmd	ASDL command.
cond_flag	TYP_cond_flag	Condition flag.
label	TYP_parm_lbl	Parameter label to test the condition flag.
value	TYP_parm_vlu	Parameter value associated with the label.
rollback_req	TYP_yes_no	Flag indicating whether rollback is required.
csdl_level	TYP_csdl_level	The level of the CSDL in the SRQ. An integer between 0 and 255 that indicates the sequence level for the CSDL command within the work order. The SARM uses this integer to determine the order in which to provision CSDL commands from an SRP and then provisions CSDL commands that have lower level numbers first. Sequence levels are only relevant for inter-dependent CSDL commands.
fail_event	TYP_code	The system event to be triggered upon CSDL failure.
complete event	TYP_code	The system event to be triggered upon CSDL completion.
option_asdl	TYP_seq_no	Not used.
description	varchar(40)	ASDL command description.

SSP_list_csdl_asdl

This function lists CSDL-to-ASDL mapping definitions contained in tbl_csdl_asdl. Wildcards are allowed.

Syntax:

```
var rc refcursor;
var retval number;
exec :retval :=  SSP_list_csdl_asdl (:rc [,'csdl_cmd'] [, 'base_seq_no'] [,'asdl_cmd']
[,'cond_flag'] [,'parm_lbl'] [,'parm_vlu'] [,'eval_exp'])
```

Example:

```
var rc refcursor;
var retval number;
exec :retval :=  SSP_list_csdl_asdl (:rc,'M-CREATE_BUS_LINE');
```

This example lists all mapping relationships associated with the CSDL command M-CREATE_BUS_LINE as follows:

```
csdl_cmd                  asdl_seq_no asdl_cmd                              cond_flag
label
value


---------- ------------------------------ --------- --------
M-CREATE_BUS_LINE                    1
M-CLEAR_INTERCEPT               A
M-CREATE_BUS_LINE                    2
M-CREATE_SINGLE_LINE_ACCESS     A
M-CREATE_BUS_LINE                    3 ADD_ALWAYS_ON_3WC               D
ALWAYS_ON_AREA
M-CREATE_BUS_LINE                    4 ADD_ALWAYS_ON_CRT               D
ALWAYS_ON_AREA

(4 rows affected, return status = 0)
```

Table 4-38 SSP_list_csdl_asdl Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
csdl_cmd	The CSDL command.	No	I
asdl_cmd	The ASDL command.	No	I
cond_flag	Used to specify conditions that need to be met in order for the SARM to generate the ASDL command for the CSDL command. Type one of the following values: • **A** – Always generates the ASDL command for the CSDL command • **D** – Generates the ASDL command if the CSDL parameter is defined (present) • **N** – Generates the ASDL command if the CSDL parameter is not defined (present) • **E** – Generates the ASDL command if the CSDL parameter is defined and equal to a value. The generation of each ASDL command depends upon the results of the previous ASDL. When the previous command completes successfully, it returns parameters to the SARM. When using 'cond_flag'='E', the following values are required: • 'lbl1' • 'lbl2' • 'val1' • 'val2' When using 'cond_flag'='D' or 'N', the following values are required: • 'lbl1' • 'lbl2'	No	I

Table 4-38 (Cont.) SSP_list_csdl_asdl Parameters

Name	Description	Req'd	(I)nput/ (O)utput
parm_lbl and parm_vlu parameters	Required when you use CSDL parameter-dependent conditions. Set the CSDL command parameter name for 'D', 'N', and 'E' condition flags using **parm_lbl**. The 'E' condition flag checks that the CSDL command parameter is equal to the value specified by **parm_vlu**. For more information about these condition flags, refer to the previous parameter, **cond_flag**.	No	I
eval_exp	Contains combination of parameter names, operators, and values to which the parameters are compared.	No	I

Table 4-39 SSP_list_csdl_asdl Results

Name	Datatype	Description
csdl_cmd	TYP_csdl_cmd	CSDL command name.
asdl_seq_no	TYP_seq_no	ASDL command sequence number.
asdl_cmd	TYP_asdl_cmd	ASDL command.
cond_flag	TYP_cond_flag	Specifies conditions that need to be met for the SARM to generate the ASDL command for the CSDL command. One of the following values: • **A** – Always generates the ASDL command for the CSDL command • **D** – Generates the ASDL command if the CSDL parameter is defined (present) • **N** – Generates the ASDL command if the CSDL parameter is not defined (present) • **E** – Generates the ASDL command if the CSDL parameter is defined and equal to a value. Each ASDL command generation depends upon the results of previous ASDL commands on the work order which returned parameters to the SARM upon successful ASDL command completion. When using 'cond_flag'='E', the following values are required: • 'lbl1' • 'lbl2' • 'val1' • 'val2' When using 'cond_flag'='D' or 'N', the following values are required: • 'lbl1' • 'lbl2'
label	TYP_parm_lbl	Parameter label to test the condition flag.
value	TYP_parm_vlu	Parameter value associated with the label.

SSP_list_csdl_defn

This function lists configuration information for the CSDL command you specify from the SARM database (tbl_csdl_config). This information includes the rollback flag, CSDL command sequence number, fail and completion events, and a description of the command. If you do not specify a CSDL command, the procedure returns information on all CSDL commands currently defined in the SARM.

Syntax:

exec :retval := SSP_list_csdl_defn (:RC1[, ‘csdl_cmd']

Example:

```
var rc refcusror;
var retval number;
exec :retval :=  SSP_list_csdl_defn (:rc,'M-CREATE_BUS_LINE')
```

For more information about using functions, see "Oracle Examples."

Table 4-40 SSP_list_csdl_defn Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
csdl_cmd	The CSDL command identifier.	No	I

Table 4-41 SSP_list_csdl_defn Results

Name	Datatype	Description
csdl_cmd	TYP_csdl_cmd	CSDL command name.
rollback_req	TYP_yes_no	Flag indicating whether rollback is required.
csdl_level	TYP_csdl_level	The level of the CSDL in the SRQ. An integer between 0 and 255 that indicates the sequence level for the CSDL command within the work order. The SARM uses this integer to determine the order in which to provision CSDL commands from an SRP and then provisions CSDL commands that have lower level numbers first. Sequence levels are only relevant for inter-dependent CSDL commands.
fail_event	TYP_code	The system event to be triggered upon CSDL failure.
complete_event	TYP_code	The system event to be triggered upon CSDL completion.
description	varchar(40)	ASDL command description.

SSP_list_dn_map

This function lists directory mappings for ASDL command, directory, exchange number, or for all of them from tbl_nep_rte_asdl_nxx.

For more information about using functions, see "Oracle Examples."

Table 4-42 SSP_list_dn_map Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
asdl_cmd	The ASDL command.	No	I
npa	The Numbering Plan Area code.	No	I
nxx	The Central Office code.	No	I

Table 4-43 SSP_list_dn_map Results

Name	Datatype	Description
asdl_cmd	TYP_asdl_cmd	ASDL command identifier.
npa	TYP_npa	The Numbering Plan Area code.
nxx	TYP_nxx	The Central Office code.
from_line	TYP_line	Beginning LINE of DN range.
to_line	TYP_line	End LINE of DN range.
queue_nm	TYP_clli	Host NE to which the DN routing applies.

SSP_list_host

This function retrieves host-related information from the following tables: tbl_resource_pool, tbl_ne_config, tbl_clli_route.

For more information about using functions, see "Oracle Examples."

Table 4-44 SSP_list_host Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
RC2	Oracle Database Ref Cursor.	Yes	I/O
host	The host CLLI identifier.	Yes	I

Table 4-45 SSP_list_host Results

Name	Datatype	Description
device	TYP_device	The name of the logical device to be used to establish a connection to the NE.
line_type	TYP_dev_type	The communication protocol used by the specified device.
vs_key	TYP_long	Reserved. The shared memory segment identifier for the Virtual Screen buffer.
mach_clli	TYP_clli	The remote NE identifier.
asdl_cmd	TYP_asdl_cmd	The ASDL command.

SSP_list_id_routing

This function lists the host NE and the ID_ROUTING mapping record in tbl_id_routing. You can use this function when routing by ID_ROUTING is used.

For more information about using functions, see "Oracle Examples."

Table 4-46 SSP_list_id_routing Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	No	I/O
host_clli	The host NE identifier.	Yes	I

Table 4-47 SSP_list_id_routing Results

Name	Datatype	Description
host_clli	TYP_clli	The host NE identifier.
asdl_cmd	TYP_asdl_cmd	The ASDL command.
id_routing_from	Varchar	The starting point of a range of ID_ROUTING.
id_routing_to	Varchar	The end point of a range of ID_ROUTING.

SSP_list_intl_msg

This function lists the international message records from tbl_msg_convert. It either lists all messages for a specified language or, if you specify a message identifier, a single record.

A list of core ASAP messages is contained in the *ASAP System Administrator's Guide*.

For more information about using functions, see "Oracle Examples."

Table 4-48 SSP_list_intl_msg Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
lang_cd	The language code.	No	I
msg_id	Unique message identifier.	No	I

Table 4-49 SSP_list_intl_msg Results

Name	Datatype	Description
lang_cd	TYP_lang_cd	The language code.
msg_id	TYP_unid	Message identifier.
msg_type	TYP_msg_typ	Message formatting types: • D – Dynamic • S – Static

Table 4-49 (Cont.) SSP_list_intl_msg Results

Name	Datatype	Description
message	varchar(255)	Message text.
var_description	varchar(40)	Description of the substitute fields.
wo_audit	TYP_wo_audit	Destination for the log message.

SSP_list_ne_host

This function lists host NE definitions from tbl_host_clli. Wildcards are allowed.

For more information about using functions, see "Oracle Examples."

Table 4-50 SSP_list_ne_host Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
host_clli	The host NE identifier of an NE or SRP.	No	I

Table 4-51 SSP_list_ne_host Results

Name	Datatype	Description
host_clli	TYP_clli	The host NE identifier of an NE or SRP.
tech_type	TYP_tech	Technology type.
sftwr_load	TYP_load	Software loads.

SSP_list_nep

This function lists NEP secondary pool definitions, stored in tbl_nep. Wildcards are allowed.

For more information about using functions, see "Oracle Examples."

Table 4-52 SSP_list_nep Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
nep_svr_cd	The NEP managing the secondary pool of devices.	No	I

Table 4-53 SSP_list_nep Results

Name	Datatype	Description
nep_svr_cd	TYP_code	The NEP managing the secondary pool of devices.
dialup_pool	TYP_pool	Secondary pool of devices.

SSP_list_nep_program

This function lists the program to be used within the Interpreter. This information is retrieved from tbl_nep_program.

For more information about using functions, see "Oracle Examples."

Table 4-54 SSP_list_nep_program Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
program	The program name.	No	I

Table 4-55 SSP_list_nep_program Results

Name	Datatype	Description
program	TYP_program	The name of the NEP program.
line_no	int	The line number in the NEP program.
action	TYP_action	The action string used to identify the action performed by the Interpreter in the command processor.
act_string	TYP_action_string	The action string associated with the NEP action.
act_int	int	The action integer which represents the next line number in the NEP program at which the NEP program should continue, or a numeric field specific to the particular action function.

SSP_list_net_elem

This function lists NE definitions based on the host NE and/or NEP server you specify. This function lists NE definitions from tbl_ne_config. Wildcards are allowed.

For more information about using functions, see "Oracle Examples."

Table 4-56 SSP_list_net_elem Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
host_clli	The host NE identifier of an NE or SRP.	No	I
nep_svr_cd	The logical name of the NEP server that connects to this host NE.	No	I

Table 4-57 SSP_list_net_elem Results

Name	Datatype	Description
host_clli	TYP_clli	The host NE.
nep_svr_cd	TYP_code	The logical name of the NEP server that connects to this host NE.

Table 4-57 (Cont.) SSP_list_net_elem Results

Name	Datatype	Description
primary_pool	TYP_pool	The primary resource pool used by the NEP managing this host NE.
max_connections	TYP_short	The maximum number of concurrent connections allowed to this host NE.
drop_timeout	TYP_short	The maximum inactivity (in minutes) before the NEP drops the primary connection to this host NE.
spawn_threshold	TYP_short	The number of ASDL requests in the SARM ASDL Ready Queue at which point the NEP opens a new auxiliary connection to the destination NE.
kill_threshold	TYP_short	The number of ASDL requests in the SARM ASDL Ready Queue. When the number of requests reaches the terminate threshold, the SARM disconnects one or more auxiliary connections.
template_flag	TYP_short	Flag to indicate if this network element entry identifies a static NE (N) or a dynamic network element template (Y).

SSP_list_resource

This function lists NEP resource records from tbl_resource_pool. Wildcards are allowed.

For more information about using functions, see "Oracle Examples."

Table 4-58 SSP_list_resource Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
asap_sys	The ASAP environment (for instance, “TEST", “PROD").	No	I
pool	The pool name.	No	I
device	The physical or logical device name.	No	I

Table 4-59 SSP_list_resource Results

Name	Datatype	Description
asap_sys	TYP_code	The ASAP environment.
pool	TYP_pool	The pool name.
device	TYP_device	The physical or logical device name.
line_type	TYP_dev_type	The device type.
vs_key	TYP_long	Reserved. The shared memory segment identifier for the Virtual Screen buffer.

SSP_list_srp

This function lists SRP definitions from tbl_asap_srp. Wildcards are allowed.

For more information about using functions, see "Oracle Examples."

Table 4-60 SSP_list_srp Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
srp_id	The logical SRP name.	No	I

Table 4-61 SSP_list_srp Results

Name	Datatype	Description
srp_id	TYP_code	The logical SRP name.
srp_desc	varchar(40)	The SRP description.
srp_conn_type	TYP_srp_conn_type	Connection protocol for the SARM to SRP.
srp_host_name	TYP_host_name	Name of the machine that the SRP resides upon.
srp_host_port	TYP_host_port	The port number that the SRP is listening on for socket connections.
wo_estimate_evt	TYP_code	The work order estimate notification event.
wo_failure_evt	TYP_code	The work order failure notification event.
wo_complete_evt	TYP_code	The work order completion notification event.
wo_start_evt	TYP_code	The work order startup notification event.
wo_soft_err_evt	TYP_code	The work order soft error notification event.
wo_blocked_evt	TYP_code	The work order blocked notification event.
wo_rollback_evt	TYP_code	The work order rollback notification event.
wo_timeout_evt	TYP_code	The work order timeout notification event.
wo_accept_evt	TYP_code	The work order acceptance notification event.
ne_unknown_evt	TYP_code	The unknown NE notification event.
ne_avail_evt	TYP_code	The NE available notification event.
ne_unavail_evt	TYP_code	The NE unavailable notification event.
aux_srp_id	TYP_code	The name of the sister SRP.
aux_srp_conn_type	TYP_srp_conn_type	Connection protocol for SARM to auxiliary SRP communications.
aux_srp_host_name	TYP_host_name	Name of the machine that the auxiliary SRP resides upon.
aux_srp_host_port	TYP_host_port	The number of the port that the auxiliary SRP is listening on for socket connections.

SSP_list_stat_text

This function is used to list static text located in tbl_stat_text. This information is retrieved from tbl_stat_text.

For more information about using functions, see "Oracle Examples."

Table 4-62 SSP_list_stat_text Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
stat_id	The logical group of static text messages such as WO_STATE. If this is not specified, all entries in tbl_stat_text are listed.	No	I

Table 4-63 SSP_list_stat_text Results

Name	Datatype	Description
srp_id	TYP_code	The logical SRP name.
status	TYP_stat	Integer key field for grouping.
code	TYP_stat_code	String key field for grouping.
stat_text	TYP_stat_txt	Text describing the label.

SSP_list_user_err_threshold

This function is used to list the user-defined error thresholds for a specific NE, ASDL command, and user error type. This information is retrieved from tbl_user_err_threshold.

For more information about using functions, see "Oracle Examples."

Table 4-64 SSP_list_user_err_threshold Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
host	The host NE identifier of an NE or SRP.	No	I
asdl_cmd	The ASDL command.	No	I
user_type	The user-defined error type.	No	I

Table 4-65 SSP_list_user_err_threshold Results

Name	Datatype	Description
host_clli	TYP_clli	Host NE.
asdl_cmd	TYP_asdl_cmd	ASDL command.
user_type	TYP_code	User-defined type.
minor_threshold	int	Minor event threshold.
minor_event	TYP_code	System event to be triggered when the minor threshold is reached.
major_threshold	int	Major event threshold.
major_event	TYP_code	System event to be triggered when the major threshold is reached.
critical_threshold	int	Critical event notification threshold.

Table 4-65 (Cont.) SSP_list_user_err_threshold Results

Name	Datatype	Description
critical_event	TYP_code	System event to be triggered when the critical threshold is reached.

SSP_list_userid

This function lists user ID definitions from tbl_uid_pwd for the SARM security check logic in the SARM connect handler. You can use this procedure to validate SRP connections to the SARM. Wildcards are allowed.

For more information about using functions, see "Oracle Examples."

Table 4-66 SSP_list_userid Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
uid	The user ID.	No	I

Table 4-67 SSP_list_userid Results

Name	Datatype	Description
uid	TYP_user_id	The user ID.
pwd	TYP_pwd	User password.
status	varchar(40)	User's current status.

SSP_ne_monitor

This is an ASAP function.

SSP_new_asdl_defn

This function defines an ASDL configuration record to tbl_asdl_config.

Syntax:

```
exec :retval :=  SSP_new_asdl_defn ('asdl_cmd','reverse_asdl','ignore_rollback',
'route_flag','[description]')
```

Example:

```
exec :retval :=  SSP_new_asdl_defn ('M-CREATE_SINGLE_LINE_ACCESS','M-
DELETE_SINGLE_LINE_ACCESS','N','B','Create a single-line access service.')
```

For more information about using functions, see "Oracle Examples."

This example defines the ASDL command M-CREATE_SINGLE_LINE_ACCESS and routes it to the NEP. If the ASDL command fails, rollback is initiated and the ASDL command **M-DELETE_SINGLE_LINE_ACCESS** is generated using the parameters from the normal ASDL command.

Table 4-68 SSP_new_asdl_defn Parameters

Name	Description	Req'd	(I)nput/ (O)utput
asdl_cmd	The ASDL command.	Yes	I
reverse_asdl	The reverse ASDL to be invoked should this ASDL require rollback. This ASDL command is only issued if the **ignore_rollback** flag is set to 'N'.	Yes	I
ignore_rollback	Specifies whether or not to rollback the ASDL command. If you want rollback to be performed on the ASDL command, set this flag to 'N', as well as specifying the reverse ASDL command.	Yes	I
route_flag	The routing of the ASDL, where: N – Routes the ASDL to the NEP. This is the only valid value.	Yes	I
description	A brief description of the ASDL command.	No	I

SSP_new_asdl_map

This function adds a new ASDL command based on the technology and software load in tbl_nep_asdl_prog. The ASDL command must be defined in the configuration records in order for this function to add a new mapping.

Syntax:

```
exec :retval :=  SSP_new_asdl_map ('tech','sftwr_load','asdl_cmd','program',
'interpreter_type')
```

Example:

```
exec :retval :=  SSP_new_asdl_map ('DMS','BCS33','M-CREATE_SINGLE_LINE_ACCESS',
'NEW_LINE_SINGLE','S')
```

This example inserts a mapping for the ASDL command **M-CREATE_SINGLE_LINE_ACCESS** for a DMS switch operating with software load BCS33. For these particular conditions, the Interpreter invokes the program NEW_LINE_SINGLE.

For more information about using functions, see "Oracle Examples."

Table 4-69 SSP_new_asdl_map Parameters

Name	Description	Req'd	(I)nput/ (O)utput
tech	The technology type of the NE or SRP with which the Interpreter is to interact.	Yes	I
sftwr_load	The version of the software currently running on the NEP or SRP.	Yes	I
asdl_cmd	The ASDL command.	Yes	I
program	The program name.	Yes	I
interpreter_type	A value of 'J' indicates a JInterpreter.	Yes	I

SSP_new_asdl_parm

This function defines up to nine ASDL parameters for the specified ASDL command starting at base_seq_no in the SARM database parameters.

You cannot add an ASDL command parameter unless the configuration record for the ASDL command has been defined in the SARM database. This ensures data consistency in the static tables.

This function adds new ASDL parameters for a specified ASDL to tbl_asdl_parm.

Syntax:

```
exec :retval :=  SSP_new_asdl_parm ('asdl_cmd','base_seq_no','asdl_parm_lbl',
'csdl_parm_lbl','[def_vlu]','[parm_typ]','asdl_parm_lbl'],'csdl_parm_lbl',
'[def_vlu]','[parm_typ]'...)
```

Example:

In the following example, seven parameters are added to the ASDL command **M-**CREATE_SINGLE_LINE_ACCESS, starting with parameter sequence number 1. All parameters are required, and the **INPA** defaults to 506 if it is not supplied. All other parameters result in a SARM translation error if the parameter is not supplied.

```
exec :retval :=  SSP_new_asdl_parm ('M-CREATE_SINGLE_LINE_ACCESS', 1,

'NPA', 'INPA', '506', 'R',
'NXX', 'INXX', '', 'R',
'LINE', 'ILINE', '', 'R',
'EXT', 'IEXT', '', 'R',
'LEN', 'ILEN', '', 'R',
'MCLI', 'IMCLI', '', 'R',
'TYPE', 'TYPE', '', 'R'
'SUBSCRIPTION[++]', 'SUBSCRIBER', '', 'C')
```

For more information about using functions, see "Oracle Examples."

Table 4-70 SSP_new_asdl_parm Parameters

Name	Description	Req'd	(I)nput/ (O)utput
asdl_cmd	The ASDL command to add parameters to. You can add up to nine parameters to **asdl_cmd**.	Yes	I
base_seq_no	Starting with a base parameter sequence number (). If you add new parameters to an existing ASDL command, enter the next available sequence number for the **base_seq_no** so that existing parameters are not overwritten.	Yes	I

Table 4-70 (Cont.) SSP_new_asdl_parm Parameters

Name	Description	Req'd	(I)nput/ (O)utput
asdl_parm_lbl	ASDL command parameter is transmitted to the NEP with the parameter value. The type of value for this parameter depends on the parameter format you chose: • **Scalar asdl_parm_lbl** – Specifies the parameter label that is sent on the ASDL command. • **Compound asdl_parm_lbl** – Specifies the base name for the parameter that is sent on the ASDL command. • **Indexed asdl_parm_lbl** – Is the base name for the indexed parameter to be sent on the ASDL command. The current index value is specified using the special literal “++".	Yes	I
csdl_parm_lbl	The parameter label either received from the SRP with the parameter value on the CSDL command or the WO parameter label returned from NEP. The label for this parameter depends on the parameter format you choose: • **Scalar csdl_parm_lbl** – Specifies the CSDL command or global WO parameter label. • **Compound csdl_parm_lbl** – Specifies the base name for the compound parameter in the CSDL command or global work order parameter list. • **Indexed csdl_parm_lbl** – Specifies the base name for the indexed parameter in the CSDL command or global work order parameter list. To specify the current index, enter the special literal “++". The initial value is 1 and each ASDL command that is sent causes the value to increment until no more parameters are found at the current index.	Yes	I
def_vlu	The default value for the scalar parameter if it is not specified in the CSDL command or global parameter list. For compound parameters, a default value is not used if a mandatory value can be found.	No	I

Table 4-70 (Cont.) SSP_new_asdl_parm Parameters

Name	Description	Req'd	(I)nput/ (O)utput
parm_typ	Indicates the parameter format by using one of the following values: • R – Required scalar parameter • O – Optional scalar parameter • C – Required compound parameter • N – Optional compound parameter • S – Parameter count. This value gives the Java method the total number of parameters associated with this ASDL command. • I – Optional indexed parameter • M – Mandatory indexed parameter • X – Required XML • Y – Optional XML • P – Required XPATH • Q – Optional XPATH • + – Current index value for the ASDL command parameter. This value is transmitted to the NEP and equals the current index value for the indexed parameter. Note, the previous example makes use of a compound indexed parameter ('SUBSCRIPTION[++]', 'SUBSCRIBER', '', 'C'). For more information on compound indexed parameters, refer to the *ASAP Cartridge Development Guide*.	No	I
al10	Reserved.	No	I

SSP_new_clli_map

This function defines a mapping from a remote CLLI to a host CLLI in tbl_clli_route.

For more information about using functions, see "Oracle Examples."

Table 4-71 SSP_new_clli_map Parameters

Name	Description	Req'd	(I)nput/ (O)utput
mach_clli	The remote NE.	Yes	I
host_clli	The host NE identifier of an NE or SRP.	Yes	I
asdl_cmd	The ASDL command.	No	I

SSP_new_comm_param

This function adds a communication parameter for a specified device type, host, and device into tbl_comm_param. The COMMON_HOST_CFG and/or parameter definition can be used for the following combinations:

- common host and common device
- common host and specific device

- specific host and common device
- specific host and specific device

Because the second and third combinations may overlap, the system warns you when communication parameters are updated.

For more information about using functions, see "Oracle Examples."

Table 4-72 SSP_new_comm_param Parameters

Name	Description	Req'd	(I)nput/ (O)utput
dev_type	The device type. Choose from the following connections: • D – Serial Port Dialup • F – TCP/IP FTP Connection • G – Generic Terminal Based Connection • H – Serial Port Hardwired • M – Generic Message Based Connection • S – TCP/IP Socket Connection • T – TCP/IP Telnet Connection • W – LDAP Connection • C – CORBA	Yes	I
host	The host CLLI. Set to COMMON_HOST_CFG or the host CLLI associated with the command processor. If a common host, the parameter value is the default value, otherwise, it is host-specific.	Yes	I
device	The physical or logical device name. Set to COMMON_DEVICE_CFG, the device associated with the command processor, or the name of a multiplexing device. If a common device, the parameter value is the default value, otherwise, it is device-specific.	Yes	I
param_label	The communication parameter label.	Yes	I
param_value	The communication parameter value.	Yes	I
param_desc	The communication parameter description.	Yes	I

SSP_new_csdl_asdl

This function defines up to nine CSDL-to-ASDL mappings from an CSDL command to an ASDL command with consecutive numbers starting from **base_seq_no** to **tbl_csdl_asdl**.

CSDL and ASDL commands must be defined before adding a CSDL-to-ASDL mapping relationship, otherwise this function rejects the insertion attempt. This is an enforced integrity check that ensures configuration consistency. You can add up to nine mapping relationships with one procedure call.

Syntax:

```
exec :retval :=  SSP_new_csdl_asdl (‘csdl_cmd’,‘base_seq_no’,
‘asdl_cmd’,‘cond_flag’,‘parm_lbl’,‘parm_vlu’,‘eval_exp’,‘prn’,
‘[asdl_cmd]’,‘[cond_flag]’,‘[parm_lbl]’,‘[parm_vlu]’,‘[eval_exp]’,

‘[prn]’, ...)
```

Example:

```
exec :retval :=  SSP_new_csdl_asdl ('M-CREATE_BUS_LINE', 1,
```

```
'M-CLEAR_INTERCEPT', 'A', '', '', '', '0',
'M-CREATE_SINGLE_LINE_ACCESS', 'A', '', '', '', '0',
'ADD_ALWAYS_ON_3WC', 'D', 'ALWAYS_ON_AREA', '', '', '0',
'ADD_ALWAYS_ON_CRT', 'D', 'ALWAYS_ON_AREA', '', '', '0',
'ADD_ALWAYS_ON_CTR', 'A', '', '','' '0')
```

In this example, the CSDL command **M-CREATE_BUS_LINE** is mapped to four ASDL commands that start with the ASDL sequence number 1.

ASDL commands **M-CLEAR_INTERCEPT** and **M-CREATE_SINGLE_LINE_ACCESS** are always generated.

ADD_ALWAYS_ON_3WC and **ADD_ALWAYS_ON_CRT** are generated only if the CSDL command parameter **ALWAYS_ON_AREA** is defined.

The fifth ASDL command, **ADD_ALWAYS_ON_CTR**, is generated only if the CSDL command parameter **TRACE_OPT** is defined and has an ON value.

For more information about using functions, see "Oracle Examples."

Table 4-73 SSP_new_csdl_asdl Parameters

Name	Description	Req'd	(I)nput/ (O)utput
csdl_cmd	The CSDL command identifier.	Yes	I
base_seq_no	The number of the first ASDL mapping to insert. Because each CSDL command may map to several ASDL commands, an index is kept in the static table and used to determine the order of the ASDL commands being sent to the NEP. Up to nine ASDL commands can be mapped with a single call of this procedure. Each call is assigned a sequence number based on the **base_seq_no**. For example, if the CSDL command maps to twelve ASDL commands, the base sequence number should be 1 for the first procedure call (which will create nine mapping relationships, ASDL command sequence numbers 1-9) and 10 for the second procedure call (which will create the remaining three mapping relationships, ASDL command sequence numbers 10-12).	Yes	I
asdl_cmd	The name of the ASDL command that the CSDL command maps to. The ASDL command is generated by the CSDL command based on the value of the condition flag (**cond_flag**).	Yes	I

Table 4-73 (Cont.) SSP_new_csdl_asdl Parameters

Name	Description	Req'd	(I)nput/ (O)utput
cond_flag	Used to specify conditions that need to be met in order for the SARM to generate the ASDL command for the CSDL command. Type one of the following values: • **A** – Always generates the ASDL command for the CSDL command • **D** – Generates the ASDL command if the CSDL parameter is defined (present) • **N** – Generates the ASDL command if the CSDL parameter is not defined (present) • **E** – Generates the ASDL command if the CSDL parameter is defined and equal to a value. The generation of each ASDL command depends upon the results of the previous ASDL. When the previous command completes successfully, it returns parameters to the SARM. When using 'cond_flag'='E', the following values are required: • 'lbl1' • 'lbl2' • 'val1' • 'val2' When using 'cond_flag'='D' or 'N', the following values are required: • 'lbl1' • 'lbl2'	Yes	I
parm_lbl and parm_vlu parameters	Required when you use CSDL parameter-dependent conditions. Set the CSDL command parameter name for 'D', 'N', and 'E' condition flags using **parm_lbl**. The 'E' condition flag checks that the CSDL command parameter is equal to the value specified by **parm_vlu**. For more information about these condition flags, refer to the previous parameter, **cond_flag**.	Yes	I
eval_exp	Contains combination of parameter names, operators, and values to which the parameters are compared.	No	I
pnr	Value of 'point of no return' for rollbacks. Values are: • 0 (default) – This ASDL is not the 'point of no return' for rollback purposes • 1 – This ASDL is the 'point of no return' for partial rollback. If rollback occurs, and the function has run continued beyond this point, roll back to this ASDL but no further. • 2 – 'point of no return' for no rollback. Once past this ASDL, no rollback can occur.	Yes	I

SSP_new_csdl_asdl_idx

This function allows multiple conditions to be inserted into **tbl_csdl_asdl_eval**. Up to nine rules can be inserted with each call. If adding up to nine rules, leave append_rule set to 0. To add more than nine rules to one mapping, call the function again with append_rule set to 1.

Syntax:

```
exec :retval :=  SSP_new_csdl_asdl_idx (‘append_rule',‘csdl_cmd', ‘base_seq_no',
‘asdl_cmd', c‘cond_flag', ‘parm_lbl', ‘parm_vlu',
‘eval_exp', ‘apply_from', ‘apply_to',
‘[cond_flag]', ‘[parm_lbl]', ‘[parm_vlu]', ‘[eval_exp]',‘[apply_from]', ‘[apply_to])
...
```

Examples:

```
exec :retval :=  SSP_new_csdl_asdl_idx (0,'M-CREATE_BUS_LINE',1,'M-CLEAR_INTERCEPT','0',
'A', '', '', '', 1, 1)

exec :retval :=  SSP_new_csdl_asdl_idx (0,'M-CREATE_BUS_LINE',2,
'M-CREATE_SINGLE_LINE_ACCESS', '0',

'A', '', '', '', 1, 1)


exec :retval :=  SSP_new_csdl_asdl_idx (0,'M-CREATE_BUS_LINE',3,'ADD_ALWAYS_ON_3WC','0',

'D', 'ALWAYS_ON_AREA', '', '', 1, 1)


exec :retval :=  SSP_new_csdl_asdl_idx (0,'M-CREATE_BUS_LINE',4,'ADD_ALWAYS_ON_CRT','0',
'D', 'ALWAYS_ON_AREA', '', '', 1, 1)

exec :retval :=  SSP_new_csdl_asdl_idx (0,'M-CREATE_BUS_LINE',5,'ADD_EMAIL_ACCOUT','0',

'E', 'MAX_ADDRESS_EXCEED', 'FALSE', '', 1, 9999,
'E', 'CREATE_ADDRESS_1', 'TRUE', '', 1, 1,
'E', 'CREATE_ADDRESS_2', 'TRUE', '', 2, 2,
'E', 'CREATE_ADDRESS_3', 'TRUE', '', 3, 3,
'E', 'CREATE_ADDRESS_4', 'TRUE', '', 4, 4,
'E', 'CREATE_ADDRESS_5', 'TRUE', '', 5, 5,
'E', 'CREATE_ADDRESS_6', 'TRUE', '', 6, 6,
'E', 'CREATE_ADDRESS_7', 'TRUE', '', 7, 7,
'E', 'CREATE_ADDRESS_8', 'TRUE', '', 8, 8)


exec :retval :=  SSP_new_csdl_asdl_idx (1,'M-CREATE_BUS_LINE',5,'ADD_EMAIL_ACCOUT','0',

'E', 'CREATE_ADDRESS_9', 'TRUE', '', 9, 9,
'E', 'CREATE_ADDRESS_10', 'TRUE', '', 10, 10,
'E', 'CREATE_ADDRESS_11', 'TRUE', '', 11, 11)
```

In this example, the CSDL command **M-CREATE_BUS_LINE** is mapped to five ASDL commands that start with the ASDL sequence number 1.

ASDL commands **M-CLEAR_INTERCEPT** and **M-CREATE_SINGLE_LINE_ACCESS** are always generated.

ADD_ALWAYS_ON_3WC and **ADD_ALWAYS_ON_CRT** are generated only if the CSDL command parameter **ALWAYS_ON_AREA** is defined.

The fifth ASDL command, **ADD_EMAIL_ACCOUNT**, is an indexed ASDL command that contains 12 rules. Each instance of this ASDL command is generated only if the CSDL command parameter **MAX_ADDRESS_EXCEED** is defined and the current index rule has a **FALSE** value and for each index rule is evaluated to true.

For example, if the current index is 5 then this rule ('E', 'CREATE_ADDRESS_5', 'TRUE', '', 5, 5) says the CSDL command parameter CREATE_ADDRESS_5 is defined and the current index rule has a TRUE value.

Note that the last call of the fifth example has **append_rule** set to 1 to indicate that the rules are to be appended to those added by the previous call:

```
exec :retval :=  SSP_new_csdl_asdl_idx (1,'M-CREATE_BUS_LINE',5...)
```

For more information about using functions, see "Oracle Examples."

Table 4-74 SSP_new_csdl_asdl_idx Parameters

Name	Description	Req'd	(I)nput/ (O)utput
append_rule	Indicates whether rules are inserted to a new or existing mapping. The initial call, with this parameter set to 0, can insert up to 9 rules. To add additional rules, call SSP_new_csdl_asdl_idx again, with this parameter set to 1.	Yes	I
csdl_cmd	The CSDL command identifier.	Yes	I
base_seq_no	The number of the first ASDL mapping to insert. Because each CSDL command may map to several ASDL commands, an index is kept in the static table and used to determine the order of the ASDL commands being sent to the NEP. Up to nine ASDL commands can be mapped with a single call of this procedure. Each call is assigned a sequence number based on the **base_seq_no**. For example, if the CSDL command maps to twelve ASDL commands, the base sequence number should be 1 for the first procedure call (which will create nine mapping relationships, ASDL command sequence numbers 1-9) and 10 for the second procedure call (which will create the remaining three mapping relationships, ASDL command sequence numbers 10-12).	Yes	I
asdl_cmd	The name of the ASDL command that the CSDL command maps to. The ASDL command is generated by the CSDL command based on the value of the condition flag (**cond_flag**).	Yes	I
pnr	Value of 'point of no return' for rollbacks. Values are: • 0 (default) – This ASDL is not the 'point of no return' for rollback purposes • 1 – This ASDL is the 'point of no return' for partial rollback. If rollback occurs, and the function has continued beyond this point, roll back to this ASDL but no further. • 2 – 'point of no return' for no rollback. Once past this ASDL, no rollback can occur.	Yes	I

Table 4-74 (Cont.) SSP_new_csdl_asdl_idx Parameters

Name	Description	Req'd	(I)nput/ (O)utput
cond_flag	Used to specify conditions that need to be met in order for the SARM to generate the ASDL command for the CSDL command. Type one of the following values: • **A** – Always generates the ASDL command for the CSDL command • **D** – Generates the ASDL command if the CSDL parameter is defined (present) • **N** – Generates the ASDL command if the CSDL parameter is not defined (present) • **E** – Generates the ASDL command if the CSDL parameter is defined and equal to a value. The generation of each ASDL command depends upon the results of the previous ASDL. When the previous command completes successfully, it returns parameters to the SARM. When using 'cond_flag'='E', the following values are required: • 'lbl1' • 'lbl2' • 'val1' • 'val2' When using 'cond_flag'='D' or 'N', the following values are required: • 'lbl1' • 'lbl2'	Yes	I
parm_lbl parm_vlu	Required when you use CSDL parameter-dependent conditions. Set the CSDL command parameter name for 'D', 'N', and 'E' condition flags using **parm_lbl**. The 'E' condition flag checks that the CSDL command parameter is equal to the value specified by **parm_vlu**. For more information about these condition flags, refer to the previous parameter, **cond_flag**.	Yes	I
eval_exp	Contains combination of parameter names, operators, and values to which the parameters are compared.	No	I
apply_from	The first indexed ASDL that this rule should apply to: Valid range is from 1 to 9999. Must be less than or equal to the value specified in column apply_to. If is not specified, then this rule will be applied to any indexed ASDL up to and including the one specified in column apply_to.	No	I
apply_to	The last indexed ASDL that this rule should apply to: Valid range is from 1 to 9999. Must be greater than or equal to the value specified in column apply_from. If is not specified, then this rule will be applied to any indexed ASDL starting from the one specified in column apply_from.	No	I

SSP_new_csdl_defn

This function adds a new CSDL command into tbl_csdl_config.

Syntax:

```
exec :retval :=  SSP_new_csdl_defn ('csdl_cmd', 'rollback_req', csdl_level,
'[fail_event]', '[complete_event]', '[description]')
```

Example:

```
exec :retval :=  SSP_new_csdl_defn ('M-CREATE_BUS_LINE', 'Y', 82, 'SYS_ERR', 'SYS_INFO',
'Add Business Access Line')
```

In this example, the **M-CREATE_BUS_LINE** CSDL command adds a business access line with a CSDL command level of 82. If the CSDL command fails, the **SYS_ERR** system event is triggered and rollback is performed on the entire order. Upon successful completion, the **SYS_INFO** system event is issued.

For more information about using functions, see "Oracle Examples."

Table 4-75 SSP_new_csdl_defn Parameters

Name	Description	Req'd	(I)nput/ (O)utput
csdl_cmd	The name of the CSDL command to add. It should be a unique CSDL command label in ASAP.	Yes	I
rollback_req	A Yes/No (Y or N) flag that indicates whether rollback is required for this CSDL command. If you set this flag to Y, ASAP automatically rolls back any actions performed by the work order if the work order fails.	Yes	I
csdl_level	An integer between 0 and 255 that indicates the sequence level for the CSDL command within the work order. The SARM uses this integer to determine the order in which to provision CSDL commands from an SRP and then provisions CSDL commands that have lower level numbers first. Sequence levels are only relevant for inter-dependent CSDL commands.	Yes	I
fail_event	ASAP system events that are triggered upon completion or failure of the CSDL command. The events must be first defined in the control database if alarms are to be generated from such events. These are optional parameters.	No	I
complete_event		No	I
description	A CSDL command description. ASAP front-end tools that are monitoring the progress of the work order can use this description. This is an optional parameter.	No	I

SSP_new_dn_map

This function adds new ASDL command routings by directory number to tbl_nep_rte_asdl_nxx.

For more information about using functions, see "Oracle Examples."

Table 4-76 SSP_new_dn_map Parameters

Name	Description	Req'd	(I)nput/ (O)utput
asdl_cmd	The ASDL command.	No	I
npa	The Numbering Plan Area code.	No	I
nxx	The Central Office code.	No	I
from_line	The lowest line number in the range of telephone numbers to provide routing for.	No	I
to_line	The highest line number in the range.	No	I
queue_nm	The host NE to which this ASDL should be routed.	Yes	I

SSP_new_id_routing

This function adds a new host NE and the ID_ROUTING mapping record to tbl_id_routing. You can use this function when routing by ID_ROUTING is used.

For more information about using functions, see "Oracle Examples."

Table 4-77 SSP_new_id_routing Parameters

Name	Description	Req'd	(I)nput/ (O)utput
host_clli	The host NE identifier.	No	I
asdl_cmd	The ASDL command.	Yes	I
id_routing_from	The starting point of a range of ID_ROUTING.	No	I
id_routing_to	The end point of a range of ID_ROUTING.	No	I

SSP_new_intl_msg

This function defines an international message for a particular language in tbl_msg_convert.

For more information about using functions, see "Oracle Examples."

Table 4-78 SSP_new_intl_msg Parameters

Name	Description	Req'd	(I)nput/ (O)utput
lang_cd	The language code.	Yes	I
msg_id	The unique message identifier.	Yes	I
type	The type of message formatting.	Yes	I
message	The message text.	Yes	I
var_description	The description of the substitutable fields, if any, within the message.	No	I
wo_audit	Destination for the log message.	No	I

The following example shows how to add an international message:

```
exec :retval :=  SSP_new_intl_msg ('USA', 1, 'D', 'Work Order %s Timed Out',  'WO Id:
%s')
```

This example adds an international message to the SARM database for American English (USA). International messages use parameters to identify the entity they are associated with. The **var_description** parameter ('WO id: %s') specifies the format and the arguments that are used to generate the actual message.

SSP_new_ne_host

This function defines a host NE with its technology type, software version, and inventory manager in the SARM database table, tbl_host_clli.

For more information about using functions, see "Oracle Examples."

Table 4-79 SSP_new_ne_host Parameters

Name	Description	Req'd	(I)nput/ (O)utput
host_clli	The host NE to which the remote NE is connected.	Yes	I
tech_type	The technology type of the host NE or SRP.	Yes	I
sftwr_load	The version of the software currently running on the NEP or SRP.	Yes	I

SSP_new_nep

This function defines a secondary (dialup) pool of devices or connections for a specified NEP in the SARM database. This function adds a pool of devices or connections to tbl_nep.

For more information about using functions, see "Oracle Examples."

Table 4-80 SSP_new_nep Parameters

Name	Description	Req'd	(I)nput/ (O)utput
nep_svr_cd	The NEP managing the secondary pool of devices.	Yes	I
dialup_pool	The secondary pool of devices.	No	I

SSP_new_nep_program

This function inserts or updates a line of code into tbl_nep_program. If the line exists, it will be updated.

If there is no ASDL mapping relationship, the user is warned that the mapping relationship does not exist. The insertion of the program, however, is not affected.

For more information about using functions, see "Oracle Examples."

Table 4-81 SSP_new_nep_program Parameters

Name	Description	Req'd	(I)nput/ (O)utput
program	The NEP program identifier.	Yes	I
line_no	The line number to delete. If set to NULL, all lines of the program are deleted.	Yes	I
action	The action string identifying a particular action performed by the Interpreter.	Yes	I
act_string	The action string associated with the interpreter.	Yes	I
act_int	The action integer.	Yes	I

SSP_new_net_elem

This function defines a host NE in the SARM database (tbl_ne_config).

For more information about using functions, see "Oracle Examples."

Table 4-82 SSP_new_net_elem Parameters

Name	Description	Req'd	(I)nput/ (O)utput
host_clli	The host NE identifier of an NE or SRP.	Yes	I
nep_svr_cd	The logical name of the NEP that connects to this host NE.	Yes	I
primary_pool	The primary resource pool used by the NEP managing this host NE to determine the devices to use to connect to it.	Yes	I
max_connections	The maximum number of concurrent connections allowed to this host NE.	Yes	I
drop_timeout	The maximum inactivity (in minutes) before NEP drops the primary connection to this host NE.	Yes	I
spawn_threshold	Number of ASDL requests in the SARM ASDL Ready Queue to be exceeded before the NEP opens a new auxiliary connection to that NE.	Yes	I
kill_threshold	Once the SARM has fewer ASDL requests in its ASDL Ready Queue than this number, it disconnects one or more auxiliary connections.	Yes	I
template_flag	Flag to indicate whether this network element entry identifies a static NE (N) or a dynamic network element template (Y).	Y	I

SSP_new_resource

This function defines an NEP resource (“device") to be used for NE access in the SARM database (tbl_resource_pool).

For more information about using functions, see "Oracle Examples."

Table 4-83 SSP_new_resource Parameters

Name	Description	Req'd	(I)nput/ (O)utput
asap_sys	The ASAP environment (TEST, PROD, etc.)	Yes	I
pool	The pool name.	Yes	I
device	The physical or logical device name.	Yes	I
line_type	The type of line for the serial communication.	Yes	I
vs_key	Reserved. The shared memory segment identifier for the Virtual Screen buffer.	No	I

SSP_new_srp

This function adds an SRP to tbl_asap_srp.

For more information about using functions, see "Oracle Examples."

Table 4-84 SSP_new_srp Parameters

Name	Description	Req'd	(I)nput/ (O)utput
srp_id	The logical SRP name.	Yes	I
srp_desc	The SRP description.	Yes	I
aux_srp_id	The name of the sister SRP.	No	I
wo_estimate_evt	The work order estimate notification event.	No	I
wo_failure_evt	The work order failure notification event.	No	I
wo_complete_evt	The work order completion notification event.	No	I
wo_start_evt	The work order startup notification event.	No	I
wo_soft-err_evt	The work order soft error notification event.	No	I
wo_blocked_evt	The work order blocked notification event.	No	I
wo_rollback_evt	The work order rollback notification event.	No	I
wo_timeout_evt	The work order timeout notification event.	No	I
ne_unknown_evt	The unknown NE notification event.	No	I
ne_avail_evt	The NE available notification event.	No	I
ne_unavail_evt	The NE available notification event.	No	I
wo_accept_evt	The system event to be issued.	No	I
srp_conn_type	Connection protocol for SARM to SRP.	No	I
srp_host_name	SRP host machine name.	No	I
srp_host_port	Port number for socket connections.	No	I
aux_srp_conn_type	Connection protocol for SARM communication to the auxiliary SRP.	No	I
aux_srp_host_name	Host machine name of the auxiliary SRP.	No	I
aux_srp_host_port	Port number for socket connections on an auxiliary SRP.	No	I

SSP_new_stat_text

This function adds new static text into tbl_stat_text. If an entry already exists for the static text identifier, the static text is updated with the new information.

For more information about using functions, see "Oracle Examples."

Table 4-85 SSP_new_stat_text Parameters

Name	Description	Req'd	(I)nput/ (O)utput
stat_id	The logical group of static text messages.	Yes	I
status	The integer identifier for member of a logical grouping.	No	I
code	The string identifier for a member of a logical grouping.	No	I
stat_text	The actual text message to use in place of a string/integer identifier.	Yes	I

SSP_new_user_err_threshold

This function creates a new user-defined error threshold in the system for the specified NE, ASDL command, and the user-defined error type in tbl_user_err_threshold.

For more information about using functions, see "Oracle Examples."

Table 4-86 SSP_new_user_err_threshold Parameters

Name	Description	Req'd	(I)nput/ (O)utput
host_clli	The host NE identifier of an NE or SRP.	Yes	I
asdl_cmd	The ASDL command.	Yes	I
user_type	The user-defined error type.	Yes	I
minor_threshold	The threshold for minor system events. This is the number of times the user_type can be returned before the corresponding minor event is generated.	Yes	I
minor_event	The minor system event to be generated when the threshold is exceeded.	Yes	I
major_threshold	The threshold for major system events. This is the number of times the user_type can be returned before the corresponding major event is generated.	Yes	I
major_event	The major system event to be generated when the major threshold is reached.	Yes	I
critical_threshold	The threshold for critical event notifications. This is the number of times the user_type can be returned before the corresponding critical event is generated.	Yes	I
critical_event	The critical system event to be generated when the critical threshold is reached.	Yes	I

SSP_new_userid

This function adds a new user account for the SARM to control access from the SRP in tbl_uid_pwd.

For more information about using functions, see "Oracle Examples."

Table 4-87 SSP_new_userid Parameters

Name	Description	Req'd	(I)nput/ (O)utput
uid	The user ID.	Yes	I
pwd	The password.	Yes	I
status	The user's current status.	No	I

SSP_orphan_purge

This stored procedure scans SARM database tables and deletes fragments of old work orders that do not have an entry in tbl_wrk_ord. Occasionally, the database becomes fragmented and records are left behind in various tables, including:

- tbl_asap_stats
- tbl_info_parm
- tbl_srq
- tbl_srq_csdl
- tbl_srq_log
- tbl_asdl_log
- tbl_srq_parm
- tbl_srq_asdl_parm
- tbl_wo_event_queue

This stored procedure is time-consuming and requires considerable system resources. Therefore, it should not run during peak hours.

For more information, see Database Purging in the *ASAP System Administrator's Guide*.

For more information about using functions, see "Oracle Examples."

Error Management

The management of errors related to provisioning by an NEP provides a detailed error tracking scheme and lets the administrator configure error-processing thresholds using NE and ASDL commands. The thresholds control the release of specific ASDL commands to the NE to prevent an excessive number of errors from occurring.

The following table lists the types or errors that can occur:

- **SUCCEED:** The ASDL provisioning was successful.
- **FAIL:** Fails the current order and stops any subsequent processing.

- **RETRY**: Retries the current ASDL command after a user-configured interval and up to a user-configured number of times before failing the order.
- **MAINTENANCE:** Causes the current ASDL command to wait for the NE to come out of maintenance before processing continues.
- **SOFT_FAIL:** An error has occurred at the NE but order processing can continue.
- **DELAYED_FAIL**: An ASDL had failed during provisioning. The SARM skips any subsequent ASDL in the CSDL, continues provisioning at the next CSDL, and then fails the order.

Refer to the *ASAP Cartridge Development Guide* for more detailed descriptions of these base_types.

User-configured history windows and polling intervals that update the ASAP database are also supported. Information is available in real-time from the SARM server or in a batch from the ASAP database. This batch information can then be used by administrative tools to perform root cause analysis.

SSP_del_err_threshold

This function deletes error thresholds for a specific NE and ASDL command from tbl_err_threshold.

For more information about using functions, see "Oracle Examples."

Table 4-88 SSP_del_err_threshold Parameters

Name	Description	Req'd	(I)nput/ (O)utput
host_clli	The host NE identifier of an NE or SRP.	No	I
asdl_cmd	The ASDL command associated with the threshold. This can be NULL to indicate an NE threshold.	No	I

SSP_del_err_type

This function deletes mappings between base and user exit types. These mappings are defined in tbl_user_err.

For more information about using functions, see "Oracle Examples."

Table 4-89 SSP_del_err_type parameters

Name	Description	Req'd	(I)nput/ (O)utput
user_type	The user-defined error type.	Yes	I
asdl	The ASDL that is running. Error types can be defined for user_type and ASDL combinations.	No	I
csdl	The CSDL that is running. Error types can be defined for user_type and CSDL combinations.	No	I
ne_vendor	The vendor of the network element.	No	I
tech_type	The technology of the network element.	No	I
sftwr_load	Software version of the host network element.	No	I

SSP_err_enable

This function enables the provisioning of an ASDL command that has been disabled because it exceeded an error threshold. If the NE is down, the NE will be enabled by this function. A particular ASDL may also be marked as disabled to an NE, therefore it may be re-enabled to that NE by specifying it along with the NE in the call to SSP_err_enable.

> **Note:**
>
> The action performed by this function is not persistent. That is, if the SARM is taken down after this function has been run, the changes made to the state of the NE and ASDL are lost.

For more information about using functions, see "Oracle Examples."

Table 4-90 SSP_err_enable Parameters

Name	Description	Req'd	(I)nput/ (O)utput
host_clli	The host NE identifier of an NE or SRP.	Yes	I
asdl_cmd	The optional ASDL command to enable a specific type of provisioning.	No	I

SSP_list_err_host

This function lists the NEs and the ASDL commands that have been disabled for provisioning.

For more information about using functions, see "Oracle Examples."

Table 4-91 SSP_list_err_host Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
host_clli	The host NE identifier of an NE or SRP.	No	I
asdl_cmd	The ASDL command.	No	I

Table 4-92 SSP_list_err_host Results

Name	DataType	Description
host_clli	TYP_clli	Host NE.
disable_dts	datetime	Timestamp when the NE was disabled.
asdl_cmd	TYP_asdl_cmd	ASDL command that has been disabled.
order_count	TYP_long	Number of orders waiting for the NE.

SSP_list_err_threshold

This function lists the error thresholds for a specific NE and ASDL command. Error thresholds are stored in tbl_err_threshold.

For more information about using functions, see "Oracle Examples."

Table 4-93 SSP_list_err_threshold Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
host_clli	The host NE identifier of an NE or SRP.	No	I
asdl_cmd	The ASDL command associated with the threshold. This can be NULL to indicate an NE threshold.	No	I

Table 4-94 SSP_list_err_threshold Results

Name	DataType	Description
host_clli	varchar(64)	The host NE identifier.
asdl_cmd	varchar(30)	The ASDL command associated with the threshold. This can be NULL to indicate an NE threshold.
threshold	TYP_long	Error threshold.

SSP_list_err_type

This function lists the mapping between user exit types and base exit types. This mapping is stored in tbl_user_err.

You cannot define both user_type and base_type at the same time.

For more information about using functions, see "Oracle Examples."

Table 4-95 SSP_list_err_type Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
user_type	The user-defined error type.	No	I
base_type	The base error type.	No	I

Table 4-96 SSP_list_err_type Results

Name	DataType	Description
user_type	TYP_code	User-defined type.
base_type	TYP_code	Base error type.
description	varchar(50)	A brief description of the user exit type.

SSP_new_err_threshold

This function adds a new threshold for a specific NE and ASDL command in tbl_err_threshold.

For more information about using functions, see "Oracle Examples."

Table 4-97 SSP_new_err_threshold Parameters

Name	Description	Req'd	(I)nput/ (O)utput
host_clli	The host NE identifier.	Yes	I
asdl_cmd	The ASDL command associated with the threshold. This can be NULL to indicate an NE threshold.	Yes	I
threshold	The error threshold for the time period (consecutive number of errors before connection to an NE should be disabled).	Yes	I

SSP_new_err_type

This function adds a new mapping between user exit types and the base exit types in tbl_user_err.

For more information about using functions, see "Oracle Examples."

Table 4-98 SSP_new_err_type Parameters

Name	Description	Req'd	(I)nput/ (O)utput
user_type	The user-defined error type.	Yes	I
base_type	The base error type.	Yes	I
description	A brief description of the ASDL command.	No	I
asdl	The ASDL that is running. Error types can be defined for user_type and ASDL combinations.	No	I
csdl	The CSDL that is running. Error types can be defined for user_type and CSDL combinations.	No	I
ne_vendor	The vendor of the network element.	No	I
tech_type	The technology of the network element.	No	I
sftwr_load	Software version of the host network element.	No	I
search_pattern	Regular expression pattern that is used to match on network element responses.	No	I

Switch Blackout Processing

If ASAP shared a port to an NE with another system or if regular NE maintenance must be performed, you can define the NE blackout period during which time the NEP will not connect to that NE.

To identify switch blackout periods, ASAP checks a database table to see if the current time is within the user-defined blackout period. You can configure both static (keyed by date and time)

and dynamic (keyed by specific day and time) blackout periods. If a blackout period is detected, the switch is placed into maintenance mode automatically.

SSP_add_blackout

This function configures the static and dynamic blackout periods for a specific NE host. Blackout information is stored in tbl_blackout.

For more information about using functions, see "Oracle Examples."

Table 4-99 SSP_add_blackout Parameters

Name	Description	Req'd	(I)nput/ (O)utput
dayname	The name of the day of the week for a weekly blackout (such as Mondays). Set to NULL to use specific date and time intervals for blackout.	No	I
host_clli	The Host NE identifier of an NE or SRP.	Yes	I
start_tm, end_tm	The start time and end time for the blackout interval. If you have specified a dayname blackout, the blackout is based on the day and the start time and end time. If the dayname parameter is set to NULL, the blackout is based on a specified date and time.	Yes	I
descr	A brief description of the blackout.	No	I

SSP_check_blackout

This function determines whether or not the specified NE is currently blacked out. An NE is blacked out if an entry exists in **tbl_blackout** for the specified NE, where the current time is between the entry's start time and end time.

For more information about using functions, see "Oracle Examples."

Table 4-100 SSP_check_blackout Parameters

Name	Description	Req'd	(I)nput/ (O)utput
curday	The current day.	Yes	I
host_clli	The Host NE identifier.	Yes	I
curr_dt_tm	The current time.	Yes	I
ret	The return status.	Yes	O

SSP_del_blackout

This procedure removes blackout periods for a specific NE host from tbl_blackout.

For more information about using functions, see "Oracle Examples."

Table 4-101 SSP_del_blackout Parameters

Name	Description	Req'd	(I)nput/ (O)utput
dayname	The name of the day of the week for default setup (e.g., Monday). Set to NULL to use specific day and time intervals for the blackout.	No	I
host_clli	The host NE identifier of an NE or SRP.	No	I
start_tm	The start time for the blackout interval. If the dayname parameter is not NULL, then these fields are used as time intervals for the day. If the dayname parameter is NULL, this field must include both date and time.	No	I

SSP_list_blackout

This procedure lists blackout periods for a specific NE host. This information is stored in tbl_blackout.

For more information about using functions, see "Oracle Examples."

Table 4-102 SSP_list_blackout Parameters

Name	Description	Req'd	(I)nput/ (O)utput
RC1	Oracle Database Ref Cursor.	Yes	I/O
RC2	Oracle Database Ref Cursor.	Yes	I/O
RC3	Oracle Database Ref Cursor.	Yes	I/O
dayname	The name of the day of the week for default setup (e.g., Monday). Set to NULL to use specific date and time intervals for the blackout.	No	I
host_clli	The host NE identifier of an NE or SRP.	No	I
start_tm	The start time for the blackout interval. If the dayname parameter is not NULL, then these fields are used as time intervals for the day. If the dayname parameter is NULL, this field must include both date and time.	No	I

Table 4-103 SSP_list_blackout Results

Name	Datatype	Description
dayname	varchar(10)	The name of the day.
host_clli	TYP_clli	The host CLLI.
start_tm	datetime	The start time for the blackout interval.
end_tm	datetime	The end time for the blackout interval.
description	varchar(40)	The description of the blackout period.
dayname	varchar(10)	The name of the day.
host_clli	TYP_clli	The host CLLI.
start_tm	datetime	The start time for the blackout interval.

Table 4-103 (Cont.) SSP_list_blackout Results

Name	Datatype	Description
end_tm	datetime	The end time for the blackout interval.
description	varchar(40)	The description of the blackout period.
dayname	varchar(10)	The name of the day.
host_clli	TYP_clli	The host CLLI.
start_tm	datetime	The start time for the blackout interval.
end_tm	datetime	The end time for the blackout interval.
description	varchar(40)	The description of the blackout period.

Stop Work Order Interface

The Stop Work Order feature is a user-generated event from the OCA client or the JSRP that is received directly by the SARM and applied to a specific work order. The event is received as a function **aims_stop_wo**.

Syntax:

```
CS_RETCODE aims_stop_wo_rpc(SRV_PROC *srvproc)
{
    ASAP_WO_ID wo_id;
    CS_INT rollback;
    CS_RETCODE ret_status;
    CS_CHAR tmp[ASAP_SRQ_EVENT_TEXT_L],evt_text[ASAP_SRQ_EVENT_TEXT_L];
    CS_CHAR audit_flag[10];
    CS_CHAR user_id[32];
```

The **aims_stop_wo** function stops a work order that is in progress. This allows the user to correct any problems that may be occurring before continuing the work order.

The function determines:

- When to stop a work order.
- When to roll back a work order once it has been stopped. Once the function has been received, the SARM applies it as an asynchronous event to the specified work order.
- When to send a return status variable to indicate whether or not the operation was successful.
- A work order for which an aims_stop_wo function was received can go through two states:
  - WO_STOP_WAIT if rollback is required and is in progress. While in this state, the work order can be cancelled.
  - WO_STOPPED if rollback is not required or has finished. While in this state, the work order can be cancelled or its status changed to WO_HELD, WO_INIT, or WO_REVIEW.

A work order is stopped only if it is in the WO_IN PROGRESS state when the **aims_stop_wo** function is received. A request for a work order in any other state is rejected immediately without affecting the work order.

Table 4-104 aims_stop_wo Arguments

Name	Description	Req'd	(I)nput/ (O)utput
wo_id	The work order identifier.	Yes	I
rollback	An integer value that specifies whether or not to roll back the work order before it is stopped. Valid values are: • 1 – Rollback • 0 – Do not rollback Any other value causes the work order to be stopped without rollback.	Yes	I
ret_status	A return parameter that stores the return value of the RPC. Valid values are: • 0 – Request to stop the work order was not accepted. • 1 – Request to stop the work order was accepted.	Yes	O
evt_text	The text of the message associated with the event.	No	I
audit_flag	Indicates which audit log receives the message: • S – SRQ log • W – Work order audit log • B – Both • N – Neither	No	I
userid	Optional user identification for audit log. Set to NULL to disable the audit log. Set to the user ID of the user who runs the procedure to enable the audit log.	No	I

Localizing International Messages

Localization is the process of preparing a product for use with a single language and character set. Localization can include:

- Translating the user interface and documentation
- Adapting time, date, and number formats
- Adding punctuation conventions
- Reconstructing icons and symbols

With the support of the ASAP localization toolkit, you can localize software and non-software components to any language based on the Roman alphabet (English, German, French, Spanish, etc.). Localization usually involves translating the user interface and documentation and adapting time, date, and number formats. In some cases, more significant changes may be required, and sometimes icons, symbols, metaphors, and even concepts must be reconsidered.

The localization toolkit does not let you localize or translate the system messages generated by third-party tools or operating systems.

The stored procedures locate international messages in the SARM database. The default language of American English is provided in the base release for ASAP. You can use it as a guide for defining other languages in ASAP.

Such messages are generated by the SARM and logged in the SARM database. They may be retrieved by the SRP and passed back to the originating system.

Use the following stored procedures to add, remove, and query international messages.

- SSP_new_intl_msg
- SSP_del_intl_msg
- SSP_list_intl_msg

Table 4-105 lists the current messages used by ASAP and distributed as part of the core release.

Table 4-105 ASAP Messages

lang_cd	msg_id	type	message	var_description
USA	1	D	Work Order %s Timed Out	WO Id: %s
USA	2	D	ASDL Command %s Skipped	ASDL: %s
USA	3	D	Cannot Find Mandatory Parameter %s, ASDL %s Fails	Parameter: %s, ASDL: %s
USA	4	D	Soft Error on ASDL %s, WO Processing Continuing	ASDL: %s
USA	5	D	ASDL %s of SRQ %d Completed	ASDL: %s, SRQ Id: %d
USA	6	D	ASDL %s of SRQ %d Failed	ASDL: %s, SRQ Id: %d
USA	7	D	Start of ASDL Provisioning Request for SRQ %d	SRQ Id: %d
USA	8	D	Sent ASDL %s to NE, Awaiting NE Response	ASDL: %s
USA	9	D	Unable to get ASDL Command %s of SRQ %d	ASDL: %s, SRQ Id: %d
USA	10	D	SRQ %d (Last CSDL %s) has Completed	SRQ Id: %d, CSDL: %s
USA	11	D	CSDL %s of SRQ %d has Completed	CSDL: %s, SRQ Id: %d
USA	12	D	Will Retry ASDL Command %s of SRQ %d. Current Retry # is %d	ASDL: %s, SRQ Id: %d, Retry #: %d
USA	13	D	ASDL Command %s of SRQ %d Failed after %d Retries	ASDL: %s, SRQ Id: %d
USA	14	D	NE %s Unavailable while Processing %s	Host Clli: %s, ASDL: %s
USA	15	D	Q Info: Queued: %02d:%02d:%02d, Start: %02d:%02d:%02d, Comp: %02d:%02d:%02d	NEP Queue Information
USA	16	D	ASDL Failure Msg: %s	NE %s ASDL Failed Message
USA	17	D	ASDL Command %s of SRQ %d Failed on Unknown NE Return Status	ASDL: %s, SRQ Id: %d
USA	18	D	NE Command: %s\nASDL Command: %s	NE Command Returned From NE: %s
USA	19	D	Network Element Routing Error, Failed SRQ %d	SRQ Id: %d
USA	20	D	ASDL %s of SRQ %d Rollback Ignored	ASDL: %s, SRQ Id: %d
USA	21	D	NE %s Unavailable while Rolling Back ASDL %s of SRQ %d	Host Clli: %s, ASDL: %s, SRQ Id: %d

Table 4-105 (Cont.) ASAP Messages

lang_cd	msg_id	type	message	var_description
USA	22	D	Roll Back ASDL %s Sent to NE	ASDL: %s
USA	23	D	Roll Back ASDL %s Rejected by NE %s	ASDL: %s
USA	24	D	Will Retry Roll Back of ASDL %s, SRQ %d, Current Retry # %d	ASDL: %s, SRQ Id: %d, Retry #: %d
USA	25	D	Roll Back ASDL %s, SRQ %d Failed After %d Retries	ASDL: %s, SRQ Id: %d, # Retries: %d
USA	26	D	Roll Back ASDL %s, SRQ %d Completed	ASDL: %s, SRQ Id: %d
USA	27	D	Roll Back ASDL %s of SRQ %d Failed\nNEP Message %s	ASDL: %s, SRQ Id: %d, NEP Message: %s
USA	28	D	Roll Back ASDL %s Failed due to Unknown NE Return Status	ASDL: %s
USA	29	D	Syntax Error Processing %s	ASDL: %s
USA	30	D	Error Detected for ASDL %s	ASDL: %s
USA	31	D	Unknown ASDL Error %s	ASDL: %s
USA	32	D	Continue to Process Next ASDL	ASDL: %s
USA	33	D	N.E. Host %s	ASDL: %s Host %s
USA	34	S	Invalid ASDL Parameter Type in Configuration	-
USA	35	D	No ASDL Configuration defined for %s	ASDL: %s
USA	36	D	Network Element %s is in maintenance	Host: %s
USA	37	D	Port Failure on Connection to %s	Host: %s
USA	38	S	SRQ Provisioning Stopped	
USA	39	D	Updated for ASDL %s of SRQ %d	ASDL: %s SRQ %d
USA	40	D	End of Indexed Parameters for ASDL %s	ASDL: %s
USA	41	D	ASDL %s Provisioning Request to %s	ASDL: %s Host: %s
USA	42	D	ASDL %s for %s Route to NE %s	ASDL: %s MCLI/DN: %s Host: %s
USA	43	D	Delay failure threshold exceeded, SRQ Provisioning Stopped	-
USA	44	S	Delay failure with rollback required, SRQ Provisioning Stopped	-

5

Web Services

This chapter describes the Oracle Communications ASAP's Web Services interface. It includes the following topics:

- Web Services Overview
- Architectural Overview of Web Services
- Web Services Interface

Web Services Overview

ASAP provides a web services interface through which external applications can manage service activation activities and operations. Using the web services interface, you can develop distributed platform-and-application-server agnostic in-house solutions.

The interface is defined in the ASAP Web Services Web Service Definition Language (WSDL) file.

ASAP Web Services runs on Oracle WebLogic Application Server. See the discussion on specific version numbers of mandatory and optional third-party software in *ASAP Installation Guide*.

The external transport protocols are HTTP, HTTPS, and JMS and the data service formats are SOAP v1.1 and 1.2.

Access-level security is provided through the implementation of the WebLogic Server WS-Policy specification, enforcing authentication.

ASAP Web Services takes advantage of the existing JSRP functionality to interact with the ASAP core.

Web Services Definition Language (WSDL)

WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of which message formats or network protocols are used in implementation. Type the following URL in your web browser and access the ASAP Web Services WSDL:

`http://server:port/env_id/Oracle/CGBU/Mslv/Asap/Ws?WSDL`

where:

- *server* is the server address.
- *port* is the port number of the Oracle WebLogic Server installation.
- *env_id* is the environment identifier specified when ASAP was installed.

HTTP protocol is used for a handshake with the application server to authenticate and request a web service client stub, which is used as the launch pad to talk to the web service. Then the client can communicate with the ASAP Web Services using one of the HTTP, HTTPS, or JMS protocols.

Architectural Overview of Web Services

ASAP Web Services supports the OSS/J standard to interact with web services clients. The ASAP Web Services WSDL works with values compliant with com.sun.java.product.oss data types. See the TM Forum website at:

https://www.tmforum.org/ossj/

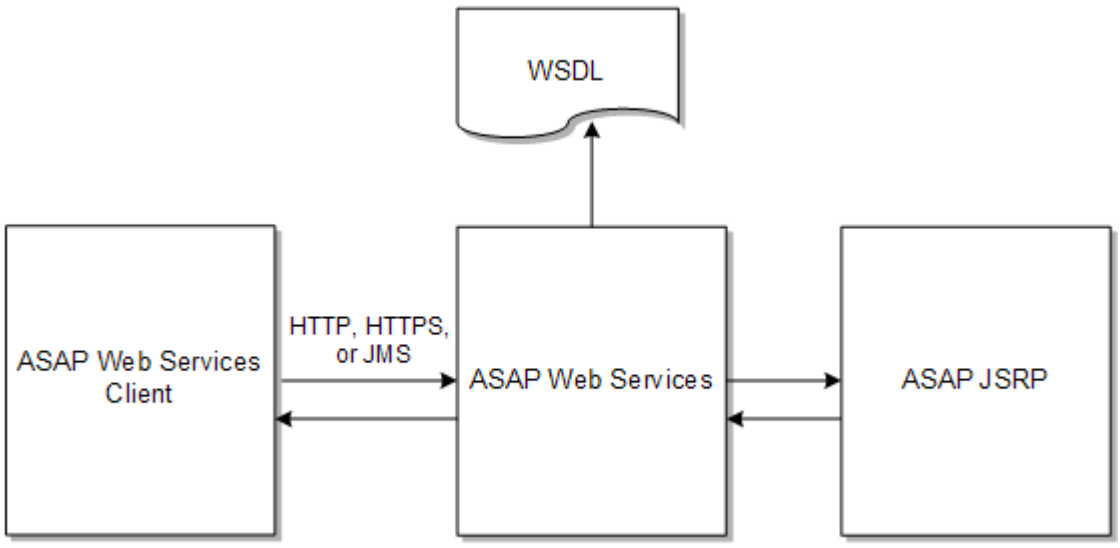
Web services exposes a web interface according to the ASAP Web Service WSDL contract for ASAP Service Activation. Web services clients can take advantage of the web services interface in accordance with ASAP's WSDL contract to exploit service activation functionality. This interface works with OSS/J compliant data types.

The ASAP Web Services message validator will check the incoming messages for compliance with ASAP OSS/J data types (for example, com.sun.java.product.oss.xml.serviceactivation.OrderValue message values actually need to be com.metasolv.serviceactivation.ASAPOrderValue values) and if incoming messages are of the expected types, submits the request to the JSRP. The JSRP forwards the request to SARM, and the results will be returned to the web services client.

Currently, web services does not support asynchronous operations and does not return JSRP event queue messages back to the clients. Web services creates a temporary queue but this queue is only for the purpose of notification of an operation completion.

Figure 5-1 illustrates the flow of messages between ASAP Web Services client and the JSRP.

Figure 5-1 ASAP Web Services with JSRP



Web Services Interface

The ASAP Web Services WSDL exposes most of the functionality that is available through JSRP in ASAP for flow through activation. See "About Web Service Operations" for more information about WSDL operations.

ASAP Web Services must use OSS/J compliant data types. The WSDL file defines message types according to the OSS/J-type XML files with ASAP extensions. You can refer to the following XSD files provided with the ASAP application for extension descriptions:

- XmlCommonSchema.xsd
- XmlServiceActivationSchema.xsd
- ASAPServiceActivation.xsd

See *ASAP Online Reference* for more information about the XSD files. The *ASAP Online Reference* can be extracted from the *ASAP_src***/doc.tar** file, where *ASAP_src* is the location of the ASAP installation files. The annotated XSD files can also be found in the ASAP environment at *ASAP_Home***/xml/xsd**.

The WSDL document declares the OSS/J elements for each web services operation. Following is a snippet for the startOrderByKey operation in a WSDL document:

```
<xs:element name="startOrderByKey">
<xs:complexType>
<xs:sequence>
<xs:element ref="s4:startOrderByKeyRequest"/>
</xs:sequence>
</xs:complexType>
</xs:element>
```

For ASAP, we need to pass a raw XML document in OSS/J plus ASAP extension type.

Security

ASAP Web Services access control security determines the functionality that each user will be able to access. In order to set up access control security, create a security role. Give this role the privilege to invoke ASAP Web Services. When the web services client needs to access the web service, the client will need to authenticate itself to the Oracle WebLogic Server hosting ASAP Web Services. (Refer to the WebLogic Server Administration Guide for details on how to set up access security.)

Note: WebLogic Server access control security only protects WebLogic Server resources and does not cover secure communication with ASAP Web Services. As a result, SOAP messages transmitted between the web service and its invoking clients are in plain text.

Currently, web services only offers access level security. Clients must use a user ID that is a member of group **ASAP_WS_USERS_GROUP** to communicate with ASAP WebServices. The **web.xml** file defines the security role **ASAP_WS_USERS** and **weblogic.xml** file defines the security principal name as **ASAP_WS_USERS_GROUP**. The ASAP installer creates a default user named **asap_ws_user**. This user is a member of the **ASA_WS_USERS_GROUP** group. Due to limitations of the WebLogic Administration Console, information created by the command-line tools such as the role name may not be available in the console.

About Web Service Operations

Table 5-1 lists the supported and unsupported web services OSS/J Common Schema base operations.

Table 5-1 Web Services Common Schema Base Operations

Supported	Unsupported
getManagedEntityTypes	getEventDescriptor
getQueryTypes	getEventTypes
getSupportedOptionalOperations	

Table 5-2 lists the supported and unsupported web services OSS/J Service Activation Schema base operations.

Table 5-2 Web Services Service Activation Schema Base Operations

Supported	Unsupported
abortOrderByKey	getOrdersByTemplates
createOrderByValue	getSupportedOptionalAttributes
getOrderByKey	makeOrderValue
getOrdersByKeys	makeServiceValue
getOrderTypes	orderAttributeValueChangeEvent
getServiceTypes	orderCreateEvent
queryOrders	orderRemoveEvent
removeOrderByKey	orderState
resumeOrderByKey	orderStateChangeEvent
setOrderByValue	priority
startOrderByKey	serviceState
suspendOrderByKey	trySetOrdersByValues
tryAbortOrdersByKeys	
tryCreateOrdersByValues	
tryRemoveOrdersByKeys	
tryStartOrdersByKeys	

Table 5-3 lists the supported and unsupported web services OSS/J ASAP Service Activation Schema base operations.

Table 5-3 Web Service ASAP Service ACtivation Schema Base Operations

Supported	Unsupported
cancelOrderByKey	abortService
lockOrder	addExtendedOrderProperty
stopOrderByKey	addOrderParameter
unlockOrder	addServiceParameter
	addService
	deleteService
	getInitOrderByKey
	orderCompleteEvent
	orderEstimateEvent
	orderFailEvent
	orderNEUnknownEvent
	orderRollbackEvent
	orderSoftErrorEvent
	orderStartupEvent
	orderTimeoutEvent
	orderTimeoutWarningEvent
	removeExtendedOrderProperty
	removeOrderParameter
	removeServiceParameter
	resubmitOrderByKey
	retryService
	setExtendedOrderProperty
	setOrderParameter
	setServiceParameter
	validateOrderOperation
	validateServiceOperation

A

Oracle Examples

The following examples demonstrate the Oracle functions that are run through the SQL*Plus client utility. These samples cover the general methods of function invocation. The Oracle login user requires the appropriate permission on the function invoked; normally, the login user is the owner of the object (for example, the SARM database user).

Example 1

A function is invoked with no input or output arguments.

```
SQL> var retval number;
SQL> exec :retval := SSP_del_csdl_defn
```

Example 2

A function is invoked with several input arguments which are specified positionally. Note that empty strings are denoted as a <space> character if the parameter is required. If the parameter is optional, the null string can be denoted by two consecutive single quotes without a <space> character.

```
SQL> var retval number;
SQL> exec :retval := SSP_new_csdl_defn('M-CREATE_BUS_LINE', 'Y', 82, ' ', ' ', ' ');
```

Example 3

This example illustrates how to add or delete information from a database. The following example clears out and then adds several rows into tbl_csdl_config.

```
SQL> set serveroutput on
SQL> var retval number

SQL> prompt Removing CSDL Definitions from the SARM

SQL> exec :retval := SSP_del_csdl_defn;

SQL> prompt Adding CSDL Definitions to the SARM

SQL> exec :retval := SSP_new_csdl_defn ('C_NEW_FLAT_LINE', 'Y', 60, '', '', 'Add flat-
rate line');

SQL> exec :retval := SSP_new_csdl_defn ('C_ADD_CIDB', 'Y', 65, '', '', 'Add Always-on
CID Block');

SQL> exec :retval := SSP_new_csdl_defn ('C_ADD_DNY_IC', 'Y', 65, '', '', 'Deny incoming
calls');

SQL> exec :retval := SSP_new_csdl_defn ('C_ADD_DNY_TOLL', 'Y', 65, '', '', 'Deny toll
calls');

SQL> exec :retval := SSP_new_csdl_defn ('C_ADD_CWT', 'Y', 80, '', '', 'Add Call
Waiting');
```

```
SQL> exec :retval := SSP_new_csdl_defn ('C_ADD_CID', 'Y', 80, '', '', 'Add Caller ID');

SQL> exec :retval := SSP_new_csdl_defn ('C_ADD_ACB', 'Y', 80, '', '', 'Add Repeat Dial--
*66');

SQL> exec :retval := SSP_new_csdl_defn ('C_ADD_AR', 'Y', 80, '', '', 'Add Return Call--
*69');

SQL> exec :retval := SSP_new_csdl_defn ('C_ADD_SCS', 'Y', 80, '', '', 'Add Speed Call
Short');

SQL> exec :retval := SSP_new_csdl_defn ('C_ADD_3WC', 'Y', 80, '', '', 'Add 3-Way
calling');

SQL> exec :retval := SSP_new_csdl_defn ('C_ADD_CFW', 'Y', 80, '', '', 'Add Call
Forward');
```

Example 4

A function is invoked with input arguments that are bound by parameter name. Optional arguments are not passed. The order of the arguments is not relevant when binding by parameter name.

```
SQL> var retval number;
SQL> exec :retval := SSP_new_csdl_defn(csdl_cmd=>'M-CREATE_BUS_LINE',
csdl_level=>82,rollback_req=>'Y');
```

Example 5

All the previous examples may be run within a PL/SQL block, as shown here:

```
SQL> declare retval number;
2> begin
3> retval := SSP_new_csdl_defn('M-CREATE_BUS_LINE', 'Y', 82, ' ', ' ', ' ');
4> end;
5> /
```

Example 6

A function with a cursor result set is invoked. This case is the most complex since the cursor must be processed before SQL*Plus can view the result set. You require the definition of the return cursor, defined in the database package object (in this example, SarmPkg). The results processing is performed within a PL/SQL block.

```
SQL> set serveroutput on
SQL> declare
  2     retcode integer;
  3     rc1     SarmPkg.SSP_list_csdl_defn_1;
  4     cur_rc1 SarmPkg.SSP_list_csdl_defn_rt1;
  5     csdl    varchar2(25) := '&csdl';
  6  begin
  7     retcode := SSP_list_csdl_defn ( rc1, csdl );
  8     dbms_output.put_line('Return code: ' || retcode);
  9     if rc1%isopen then
 10        loop
 11           fetch rc1 into cur_rc1;
 12           exit when rc1%notfound;
 13           dbms_output.put_line('csdl_cmd = '||cur_rc1.csdl_cmd);
```

```
 14           dbms_output.put_line('rollback_req = '||cur_rc1.rollback_req);
 15           dbms_output.put_line('csdl_level = '||cur_rc1.csdl_level);
 16           dbms_output.put_line('fail_event = '||cur_rc1.fail_event);
 17           dbms_output.put_line('complete_event = '||cur_rc1.complete_event);
 18           dbms_output.put_line('description = '||cur_rc1.description);
 19        end loop;
 20        close rc1;
 21     end if;
 22  end;
 23  /
```

Example 7

In the following example, a Korn shell wrapper enables you to set values before running the script. The following example sets the diag level for all of the servers in one place.

```
# CTRL Tables:  tbl_appl_proc
#  tbl_component
#
#        File:  svr_cfg
#
#     Purpose:  To define the ASAP servers, and allow use of the
#  Class A start scripts.
#
# Stored Procedure Parameter Format:
# CSP_del_appl:
#    *Application Code (Server Name)
# CSP_new_appl:
#     Start Sequence
#     Server Type (M-Master Control, S-Server, C-Client)
#     Application Code (Server Name)
#     Control Server
#     Auto-Start (Y/N)
#     Program (name of executable in $PROGRAMS)
#     Diagnostic level
#     Diagnostic file name
#     Description of server
#    *Server type (ADM, CTRL, MASTER, SARM, OTHER)
# CSP_del_component:
#    *ASAP Territory
#    *ASAP System
#    *Server Name
# CSP_new_component:
#     ASAP Territory
#     ASAP System
#     Server Name
#
# * indicates an optional parameter
#
#

scr=$(whence $0)

# Get database password
CTRL_PASSWORD=$(GetPassword $CTRL_USER 2)

# Define Local Servers
LOC_SRPC=SRPC$ASAP_ENV

# Define default diagnostic level
DIAG_LEVEL=LOW
```

```
sqlplus -s $CTRL_USER/$CTRL_PASSWORD <<HERE | grep -v "successfully completed"
set serveroutput on
var retval number


prompt Removing the ASAP Applications

exec :retval := CSP_del_appl;


prompt Defining the ASAP Applications in Territory $ASAP_TERRITORY, Local
System $ASAP_SYSTEM

exec :retval := CSP_new_appl (1, 'M', '$CTRL', '$CTRL', 'N', 'ctrl_svr', '$DIAG_LEVEL',
'${CTRL}.diag', 'Master Control Server', 'MASTER');
exec :retval := CSP_new_appl (2, 'S', '$SARM', '$CTRL', 'Y', 'sarm', '$DIAG_LEVEL', '$
{SARM}.diag', 'SARM Server','SARM');
exec :retval := CSP_new_appl (3, 'S', '$SRP', '$CTRL',  'Y', 'srp_emul', '$DIAG_LEVEL',
'${SRP}.diag', 'SRP Emulator','SRP');
exec :retval := CSP_new_appl (5, 'S', '$NEP', '$CTRL',  'Y', 'asc_nep', '$DIAG_LEVEL', '$
{NEP}.diag', 'NEP Server','NEP');
exec :retval := CSP_new_appl (10, 'C', '$DAM', '$CTRL',  'N', 'run_asapd',
'$DIAG_LEVEL', '${DAM}.diag', 'ASAP Daemon','daem');
exec :retval := CSP_new_appl (13, 'C', '$JSRP', '$CTRL', 'N', 'srp.ear', '$DIAG_LEVEL',
'${JSRP}.diag', 'Java SRP Server','SRP');
exec :retval := CSP_new_appl (14, 'C', '$LOC_SRPC', '$CTRL', 'Y', 'runSrpClient',
'$DIAG_LEVEL', '${LOC_SRPC}.diag', 'Java SRP Client','SRP');


prompt Removing the ASAP Components

exec :retval := CSP_del_component;


prompt Defining the ASAP Components for Territory $ASAP_TERRITORY, Local
System $ASAP_SYSTEM

exec :retval := CSP_new_component ('$ASAP_TERRITORY', '$ASAP_SYSTEM', '$CTRL');
exec :retval := CSP_new_component ('$ASAP_TERRITORY', '$ASAP_SYSTEM', '$SARM');
exec :retval := CSP_new_component ('$ASAP_TERRITORY', '$ASAP_SYSTEM', '$ADM');
exec :retval := CSP_new_component ('$ASAP_TERRITORY', '$ASAP_SYSTEM', '$NEP');
exec :retval := CSP_new_component ('$ASAP_TERRITORY', '$ASAP_SYSTEM', '$DAM');
exec :retval := CSP_new_component ('$ASAP_TERRITORY', '$ASAP_SYSTEM', '$JSRP');
exec :retval := CSP_new_component ('$ASAP_TERRITORY', '$ASAP_SYSTEM', '$LOC_SRPC');


HERE
```