Oracle® Communications ASAP

SUN_LDAP_5_2_SUB_1_0 Cartridge Guide

First Edition

March 2012

ORACLE®

Oracle Communications SUN_LDAP_5_2_SUB_1_0 Cartridge Guide, First Edition

Contents

4 Configuring ASAP to Support Additional NE Instances

1

Cartridge Overview

This guide provides a detailed description of the SUN_LDAP_5_2_SUB_1_0 Cartridge cartridge. It contains overview and technical information to assist with extending and integrating the cartridge into a customer environment.

The scope of this guide includes Oracle Communications ASAP (ASAP) as it pertains to the use of this cartridge. It is not intended to be a complete ASAP reference guide. For additional information when using this cartridge, refer to the ASAP documentation.

The SUN_LDAP_5_2_SUB_1_0 Cartridge cartridge provides the ASAP service configuration and network element (NE) interface to activate subscriber services on SUNW-LDAP_5-2_HOST NEs.

Hardware and Software Requirements

The following sections contain the high-level software and hardware environment requirements for provisioning subscriber services on authentication center:

- NE Interface
- ASAP Version

NE Interface

The following database tables in Service Activation Request Manager (SARM) are configured to support the NE configuration:

- tbl_host_clli
- tbl_clli_route
- tbl_comm_param
- tbl_resource_pool
- tbl_ne_config

ASAP Version

This cartridge was developed and tested using ASAP version 7.2.0.

For more information on the operating environment of this ASAP version, refer to the ASAP version 7.2.0 Release Notes.

Connecting to the NE

The cartridge uses LDAP protocol.

Services, Features, and Options

This cartridge supports the following services:

Table 1–1 Supported Services

Service	Description
C_SUNW-LDAP_5-2_MODIFY_DN	Modify DN of an entry in the directory.
C_SUNW-LDAP_5-2_QUERY_DN	Query LDAP directory for a given DN(Distinguished name). Equivalent to reading the entry with a given DN.
C_SUNW-LDAP_5-2_ADD_ENTRY	Add an entry to the directory given its DN, object class and attributes.
C_SUNW-LDAP_5-2_DELETE_ENTRY	Delete an entry from the directory given its DN.
C_SUNW-LDAP_5-2_MODIFY_ENTRY	Modify an entry(add or delete attribute, attribute values, replace an attribute value etc) in the directory.
C_SUNW-LDAP_5-2_QUERY_SCHEMA	Query the LDAP schema for all object classes, attribute types and matching rules.
C_SUNW-LDAP_5-2_QUERY_WITH-SEARCH-FILTER	Query the LDAP directory for all matching entries of a search filter.

Communication Parameters

The following is the list of parameters for the sample NE configuration XML used by Service Activation Configuration Tool (SACT).

Table 1–2 Communication Parameters

Parameter Label	Parameter Value	Description
HOST_IPADDR	213.46.231.219	The host name or IP Address of the remote NE.
PORT	5005	Port number to connect on remote NE host.
OPEN_TIMEOUT	20	Connection timeout in seconds.
READ_TIMEOUT	30	Read timeout in seconds.
HOST_USERID	mail=metasolv@lab.chello.com,ou=administration,dc=lab,dc=chello,dc=com	Bind DN
HOST_PASSWORD	MeTaSoLv	Bind Password
LDAP_VERSION	3	LDAP protocol version.
SIZELIMIT	500	Maximum no of search results to be returned.
RESPONSELOG	TRUE	Enable/Disable response log
DELIMITER	|	Delimiter to be used between values for an attribute

Related Documentation

This cartridge is developed according to the following NE provisioning specifications:

- DirectoryServer-TechnicalOverview.pdf

2

Atomic Service Description Layer (ASDL) Commands

ASDL commands represent a set of atomic actions that ASAP can perform on a network element (NE). ASAP can combine ASDLs to create meaningful services (CSDLs) within a cartridge.

This chapter presents detailed information on the ASDL parameters that we provide with this cartridge. The following table lists and describes the type of parameter information that is included.

Table 2–1 ASDL Parameter Information

Item	Description
Parameter Name	Identifies the parameter that is configured for the stated service.
Description	Describes the parameter.
Range	Describes or lists the range of values that can be used to satisfy this parameter.
Default Value	Configures a default value for the parameter so that it is not mandatory for the upstream system to provide a value.
Type	Indicates one of the following parameter types: ■ S - Scalar, specifies the parameter label transmitted on the ASDL command. Scalar parameters are conventional name-value pair parameters. ■ C - Compound, specifies the base name of the compound parameter transmitted on the ASDL command. A compound parameter contains structures or arrays of information that are represented by a particular structure name or compound parameter name. Each compound parameter can contain a large number of elements. If you use compound parameters, you only require a single entry in the ASAP translation tables to call the compound parameter and all its associated parameter elements. ■ I - Indexed, identifies a parameter that contains a sequential numerical index value to tell the SARM that it should execute the same operation (for example, an ASDL command) for all occurrences of that index. Consequently, if there are several options on a particular CSDL command (OPT1, OPT2, OPT3, etc.), you can specify the OPT parameter as an indexed parameter. When you specify the OPT parameter as an indexed parameter, the SARM generates several occurrences of that same ASDL command and each command has a different value for the option being transmitted to the NEP. For more information on parameter types, refer to the *ASAP Developer's Guide*.

Table 2–1 (Cont.) ASDL Parameter Information

Item	Description
Class	Indicates one of the following parameter classifications: ■ R - Required scalar parameter ■ O - Optional scalar parameter ■ C - Required compound parameter ■ N - Optional compound parameter ■ M - Mandatory indexed parameter ■ I - Optional indexed parameter ■ S - Parameter count

For a detailed description of the Required and Optional parameter classifications, refer to the *ASAP System Administrator's Guide.*

ASDL Commands

This cartridge provides the following ASDL commands:

- A_SUNW-LDAP_5-2_ADD_ENTRY
- A_SUNW-LDAP_5-2_ADD_ENTRY-RB
- A_SUNW-LDAP_5-2_DELETE_ENTRY
- A_SUNW-LDAP_5-2_MODIFY_DN
- A_SUNW-LDAP_5-2_MODIFY_DN-RB
- A_SUNW-LDAP_5-2_MODIFY_ENTRY
- A_SUNW-LDAP_5-2_MODIFY_ENTRY-RB
- A_SUNW-LDAP_5-2_QUERY_DN
- A_SUNW-LDAP_5-2_QUERY_SCHEMA
- A_SUNW-LDAP_5-2_QUERY_WITH-SEARCH-FILTER

A_SUNW-LDAP_5-2_ADD_ENTRY

Add an entry to the directory given its DN, object class and attributes. It is implemented by the Java method **com.metasolv.cartridge.oss.sunw_ldap_5_2.prov.LDAPProvisioning.addEntry.**

Table 2–2 A_SUNW-LDAP_5-2_ADD_ENTRY

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	NE Logical ID	N/A	N/A	S	R
DN	DN of the entry to be added	N/A	N/A	S	R
ATTRIBUTE	This compound parameter contains the attribute types and values of the attributes of the entry to be added. For example ATTRIBUTE[1].TYPE subtoken contains the attribute type and ATTRIBUTE[1].VALUE subtoken contains the attribute values.	N/A	N/A	C	R

MML commands

MML Syntax :

```
add(netscape.ldap.LDAPEntry entry)

<entry> would have the below values:
[dn = <DN>
attributes :
<ATTRIBUTE[1].TYPE>=<ATTRIBUTE[1].VALUE>
<ATTRIBUTE[2].TYPE>=<ATTRIBUTE[2].VALUE>
<ATTRIBUTE[3].TYPE>=<ATTRIBUTE[3].VALUE>
..
<ATTRIBUTE[n].TYPE>=<ATTRIBUTE[n].VALUE>]
```

Output Parameters

None.

A_SUNW-LDAP_5-2_ADD_ENTRY-RB

Rollback the delete of an entry to the directory given its DN, object class and attributes. It is implemented by the Java method **com.metasolv.cartridge.oss.sunw_ldap_5_2.prov.LDAPProvisioning.addEntryRB.**

Table 2–3 A_SUNW-LDAP_5-2_ADD_ENTRY-RB

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	NE Logical ID	N/A	N/A	S	R
OLD_DN	DN of the entry to be added	N/A	N/A	S	R
OLD_ATTRIBUTE	This compound parameter contains the attribute types and values of the attributes of the entry to be added. For example ATTRIBUTE[1].TYPE subtoken contains the attribute type and ATTRIBUTE[1].VALUE subtoken contains the attribute values.	N/A	N/A	C	R

MML commands

MML Syntax :

```
add(netscape.ldap.LDAPEntry entry)

<entry> would have the below values:
[dn = <OLD_DN>
attributes :
<OLD_ATTRIBUTE[1].TYPE>=<OLD_ATTRIBUTE[1].VALUE>
<OLD_ATTRIBUTE[2].TYPE>=<OLD_ATTRIBUTE[2].VALUE>
<OLD_ATTRIBUTE[3].TYPE>=<OLD_ATTRIBUTE[3].VALUE>
..
<OLD_ATTRIBUTE[n].TYPE>=<OLD_ATTRIBUTE[n].VALUE>]
```

Output Parameters

None.

A_SUNW-LDAP_5-2_DELETE_ENTRY

Delete an entry from the directory given its DN. It is implemented by the Java method **com.metasolv.cartridge.oss.sunw_ldap_5_2.prov.LDAPProvisioning.deleteEntry.**

Table 2–4 A_SUNW-LDAP_5-2_DELETE_ENTRY

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	NE Logical ID	N/A	N/A	S	R
DN	DN of the entry to be deleted	N/A	N/A	S	R

MML commands

MML Syntax :

```
delete(java.lang.String dn)
[dn: <DN>]
```

Output Parameters

None.

A_SUNW-LDAP_5-2_MODIFY_DN

Modify DN of an entry in the directory. It is implemented by the Java method **com.metasolv.cartridge.oss.sunw_ldap_5_2.prov.LDAPProvisioning.modifyDN.**

Table 2–5 A_SUNW-LDAP_5-2_MODIFY_DN

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	NE Logical ID	N/A	N/A	S	R
DN	DN of the entry to be modified	N/A	N/A	S	R
NEW_RDN	New RDN of the entry	N/A	N/A	S	R
NEW_PARENT_DN	DN of the new parent under which the entry to be relocated	N/A	N/A	S	O
DELETE_OLD_RDN	If true, the old RDN value is deleted. Possible values : true or false	N/A	N/A	S	R

MML commands

MML Syntax :

```
rename(java.lang.String dn, java.lang.String newRDN, java.lang.String newParentDN,
boolean deleteOldRDN)
```

Output Parameters

None.

A_SUNW-LDAP_5-2_MODIFY_DN-RB

Rollbacks modification to DN of an entry. It is implemented by the Java method **com.metasolv.cartridge.oss.sunw_ldap_5_2.prov.LDAPProvisioning.modifyDNRB.**

Table 2–6 A_SUNW-LDAP_5-2_MODIFY_DN-RB

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	NE Logical ID	N/A	N/A	S	R
DN	DN of the entry to be modified	N/A	N/A	S	R
NEW_RDN	New RDN of the entry	N/A	N/A	S	R
NEW_PARENT_DN	DN of the new parent under which the entry to be relocated	N/A	N/A	S	O

MML commands

MML Syntax :

```
rename(java.lang.String dn, java.lang.String newRDN, java.lang.String newParentDN,
boolean deleteOldRDN)
```

Output Parameters

None.

A_SUNW-LDAP_5-2_MODIFY_ENTRY

Modify an entry(add or delete attribute, attribute values, replace an attribute value etc) in the directory. It is implemented by the Java method **com.metasolv.cartridge.oss.sunw_ldap_5_2.prov.LDAPProvisioning.modifyEntry.**

Table 2–7 A_SUNW-LDAP_5-2_MODIFY_ENTRY

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	NE Logical ID	N/A	N/A	S	R
DN	DN of the entry to be modified	N/A	N/A	S	R
ATTRIBUTE	This compound parameter contains for each attribute to be modified, the modification type, attribute type and values. For example ATTRIBUTE[1].MODIFY_ACTION contains the modification type i.e ADD, DELETE or REPLACE , ATTRIBUTE[1].TYPE contains the attribute type and ATTRIBUTE[1].VALUE contains the attribute values.	N/A	N/A	C	R

MML commands

MML Syntax :

```
modify(java.lang.String dn, netscape.ldap.LDAPModification[] mods)
```

```
<dn> would have the below value:
dn = <DN>
<mods> would have the below values:
attributes :
<ATTRIBUTE[1].MODIFY_ACTION> <ATTRIBUTE[1].TYPE>=<ATTRIBUTE[1].VALUE>
<ATTRIBUTE[2].MODIFY_ACTION> <ATTRIBUTE[2].TYPE>=<ATTRIBUTE[2].VALUE>
<ATTRIBUTE[3].MODIFY_ACTION> <ATTRIBUTE[3].TYPE>=<ATTRIBUTE[3].VALUE>
..
<ATTRIBUTE[n].MODIFY_ACTION> <ATTRIBUTE[n].TYPE>=<ATTRIBUTE[n].VALUE>
```

Output Parameters

None.

A_SUNW-LDAP_5-2_MODIFY_ENTRY-RB

Modify an entry(add or delete attribute, attribute values, replace an attribute value etc) in the directory. It is implemented by the Java method **com.metasolv.cartridge.oss.sunw_ldap_5_2.prov.LDAPProvisioning.modifyEntryRB.**

Table 2–8 A_SUNW-LDAP_5-2_MODIFY_ENTRY-RB

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	NE Logical ID	N/A	N/A	S	R
OLD_DN	DN of the entry to be modified	N/A	N/A	S	R
OLD_ATTRIBUTE	This compound parameter contains for each attribute to be modified, the modification type, attribute type and values. For example OLD_ATTRIBUTE[1].MODIFY_ACTION contains the modification type i.e ADD, DELETE or REPLACE , OLD_ATTRIBUTE[1].TYPE contains the attribute type and OLD_ATTRIBUTE[1].VALUE contains the attribute values.	N/A	N/A	C	R

MML commands

MML Syntax :

```
modify(java.lang.String dn, netscape.ldap.LDAPModification[] mods)

<dn> would have the below value:
dn = <OLD_DN>
<mods> would have the below values:
attributes :
<OLD_ATTRIBUTE[1].MODIFY_ACTION> <OLD_ATTRIBUTE[1].TYPE>=<OLD_ATTRIBUTE[1].VALUE>
<OLD_ATTRIBUTE[2].MODIFY_ACTION> <OLD_ATTRIBUTE[2].TYPE>=<OLD_ATTRIBUTE[2].VALUE>
<OLD_ATTRIBUTE[3].MODIFY_ACTION> <OLD_ATTRIBUTE[3].TYPE>=<OLD_ATTRIBUTE[3].VALUE>
..
<OLD_ATTRIBUTE[n].MODIFY_ACTION> <OLD_ATTRIBUTE[n].TYPE>=<OLD_ATTRIBUTE[n].VALUE>
```

Output Parameters

None.

A_SUNW-LDAP_5-2_QUERY_DN

Query LDAP directory for a given DN(Distinguished name). Equivalent to reading the entry with a given DN. It is implemented by the Java method **com.metasolv.cartridge.oss.sunw_ldap_5_2.prov.LDAPProvisioning.queryDN.**

Table 2–9 A_SUNW-LDAP_5-2_QUERY_DN

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	NE Logical ID	N/A	N/A	S	R
DN	Distinguished name of the entry to be queried.	N/A	N/A	S	R

MML commands

MML Syntax :

```
search(<DN>, LDAPv2.SCOPE_BASE, "(objectclass = *)", null, false);
```

Output Parameters

Return the below as INFO Param and CSDL Param.

DN[n] Distinguished name of the entry

DN[n].ATTRIBUTE[m].TYPE Attribute type

DN[n].ATTRIBUTE[m].VALUES Delimiter separated string of attribute values

A_SUNW-LDAP_5-2_QUERY_SCHEMA

Query the LDAP schema for all object classes, attribute types and matching rules. It is implemented by the Java method **com.metasolv.cartridge.oss.sunw_ldap_5_2.prov.LDAPProvisioning.querySchema.**

Table 2–10 A_SUNW-LDAP_5-2_QUERY_SCHEMA

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	NE Logical ID	N/A	N/A	S	R

MML commands

MML Syntax :

```
fetchSchema(LDAPConnection ld);
```

Output Parameters

Return the below as INFO Param and CSDL Param.

OBJECTCLASS[n].OID Object identifier of the object class

OBJECTCLASS[n].NAME Name of the object class

OBJECTCLASS[n].DESCRIPTION Description of the object class

OBJECTCLASS[n].PARENT Name of the parent object class

OBJECTCLASS[n].REQ_ATTRIBUTES Required attributes name

OBJECTCLASS[n].OPT_ATTRIBUTES Optional attributes name.

ATTRIBUTE[m].OID Object identifier of the attribute.

ATTRIBUTE[m].NAME Name of the attribute.

ATTRIBUTE[m].DESCRIPTION Description of the attribute.

ATTRIBUTE[m].SYNTAX Attribute syntax

ATTRIBUTE[m].IS_SINGLE_VALUED Flag indicating whether the attribute is single valued or multi valued. Possible values: True, False

MATCHING_RULE[k].OID Object identifier of the matching rule.

MATCHING_RULE[k].NAME Name of the matching rule.

MATCHING_RULE[k].DESCRIPTION Description of the matching rule.

MATCHING_RULE[k].SYNTAX Syntax of the matching rule.

A_SUNW-LDAP_5-2_QUERY_WITH-SEARCH-FILTER

Query the LDAP directory for all matching entries of a search filter. It is implemented by the Java method **com.metasolv.cartridge.oss.sunw_ldap_5_2.prov.LDAPProvisioning.queryWithSearchFilter.**

Table 2–11 A_SUNW-LDAP_5-2_QUERY_WITH-SEARCH-FILTER

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	NE Logical ID	N/A	N/A	S	R
BASE_DN	The Base DN from where the search should start.	N/A	N/A	S	R
SEARCH_SCOPE	Possible values are:SCOPE_BASE,SCOPE_ONE,SCOPE_SUB	N/A	N/A	S	R
SEARCH_FILTER	Search filter string. For example (cn=Jensen).	N/A	N/A	S	R

MML commands

MML Syntax :

```
search(java.lang.String base,int scope, java.lang.String filter,java.lang.String[]
attrs, boolean attrsOnly)
```

Output Parameters

Return the below as INFO Param and CSDL Param.

DN[n] Distinguished name of the entry

DN[n].ATTRIBUTE[m].TYPE Attribute type

DN[n].ATTRIBUTE[m].VALUES Delimiter separated string of attribute values

User Exit Types

User exit types allow cartridge developers and systems administrators to map ASDL exit codes to one of the predefined base exit types. Base exit types determine the product behavior. Cartridges map return codes and status values from a network element to a user defined exit type.

Regular expressions (regex) are used to perform pattern searches on responses from network elements. The pattern is stored in "tbl_user_err" in the SARM database. The user exit type contains a regex pattern that is applied at runtime.

Regular expressions enable users to associate a series of responses to a specific base type. For example, a regular expression "6." can identify a pattern where any response with the character "6" followed by any number of characters will translate to base type of FAIL.

Regular expressions can also allow very specific searches within a response from a network element. Regular expressions are typically compiled before being executed. Compilation produces a binary version of the expression and ensures that the syntax of the regular expression is correct. This compilation occurs using SACT\SADT when user exit types are deployed into ASAP. If the syntax is deemed to be incorrect during compilation, SADT displays an error message and the deployment of the user exit type will fail.

For more information on pattern matching, refer to the *ASAP Developer's Guide* and the *ASAP System Administrator's Guide*.

Understanding User Exit Type XML Files

```
…
<userDefinedExitType>
<neDescriptor>
<softwareLoad>DYNAMIC_SL</softwareLoad>
<technology>DYNAMIC_VENDOR-DYNAMIC_TECH</technology>
</neDescriptor>
<searchPattern>SUCCESS.</searchPattern>...1
<userType>U_SUCCEED</userType>...2
<baseType>SUCCEED<</baseType>...3
<description>The ASDL provisioning was successful</description>
</userDefinedExitType>
<userDefinedExitType>
<searchPattern>90.</searchPattern>
<userType>U_FAIL</userType>
<baseType>FAIL</baseType>
<description>The ASDL failed - fail the current order and stop
processing.</description>
</userDefinedExitType>
<userDefinedExitType>
<searchPattern>101-110[201-215]</searchPattern>...4
<userType>U_SOFT_FAIL</userType>
<baseType>SOFT_FAIL</baseType>
<description>The ASDL has encountered a soft failure. Processing will
continue.</description>
</userDefinedExitType>
<userDefinedExitType>
<searchPattern>801-850</searchPattern>...5
<userType>U_MINOR_ERROR</userType>
<baseType>SOFT_FAIL</baseType>
<description>The ASDL has encountered a soft failure. Processing will
continue.</description>
```

```
</userDefinedExitType>
<userDefinedExitType>
<searchPattern>251-275&&[^261-265]</searchPattern>...6
<userType>U_DELAYED_FAIL</userType>
<baseType>DELAYED_FAIL</baseType>
<description>The ASDL has failed during provisioning.</description>
</userDefinedExitType>
<userDefinedExitType>
<neDescriptor>
<softwareLoad>BCS36</softwareLoad>
<technology>NORTEL_DMS</technology>
<neVendor>Nortel</neVendor>
</neDescriptor>
<searchPattern>*.</searchPattern>
<userType>U_MAINTAIN</userType>
<baseType>MAINTENANCE</baseType>
<description>The ASDL will Wait until the NE comes out of Maintenance
Mode</description>
</userDefinedExitType>
```

The numbered elements highlighted in bold in the previous code sample are explained as follows:

1. Pattern searches accommodate situations in which responses from the device contain small variants that represent the same meaning. The user type contains an associated search pattern that is applied at runtime. Using regular expressions, you can default a series of responses. For example a regular expression "90." can specify a pattern where any response with the character "90" followed by any character will translate to base type of FAIL. If the regular expression is defined as "90*", then any response with the character "90" followed by any number of characters will translate to base type of FAIL.
2. The user type that the search pattern maps to.
3. The base type that maps to the user type.
4. 101 to 110 and 201 to 215 will translate to a base type of SOFT_FAIL
5. 801-850 will translate to a base type of SOFT_FAIL. Note that the user type differs from the previous range.
6. 251 to 275 but not 261 to 265 will translate to a base type of DELAYED_FAILURE.

The previous code sample shows some typical search pattern examples. Some additional examples follow:

- ^.*\b(one|two|three)\b.*$ = matches a complete line of text that contains any of the words "one", "two" or "three"
- ^(?=.*?\bone\b)(?=.*?\btwo\b)(?=.*?\bthree\b).*$ matches a complete line of text that contains all of the words "one", "two" and "three"
- "[^"\r\n]*" matches a single-line string that does not allow the quote character to appear inside the string.
- \b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b matches any IP address.

For more information on search patterns, refer to http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html.

For more information on user exit types, refer to the *ASAP Developer's Guide*.

User Defined ASDL Exit Types

The following table lists the user defined ASDL exit types.

Table 2–12 User Defined ASDL Exit Types

Search Pattern	User Type	Base Type	Description
0	SLDAP_SUCCESS	SUCCEED	Success
1	SLDAP_OPR_ERROR	FAIL	Operation error
2	SLDAP_PROTOCOL_ERR	FAIL	Protocol error
3	SLDAP_TIME_EXCEED	FAIL	Time limit exceeded
4	SLDAP_SIZE_EXCEED	FAIL	Size limit exceeded
5	SLDAP_CMP_FALSE	FAIL	Compare false
6	SLDAP_CMP_TRUE	FAIL	Compare true:e j
7	SLDAP_AUTH_NOSUP	FAIL	Authentication method not supported
8	SLDAP_AUTH_REQ	FAIL	Strong authentication required
9	SLDAP_PARTIAL_RSLT	FAIL	LDAP partial results
10	SLDAP_REFERRAL	FAIL	Referral
11	SLDAP_ADMLT_EXCEED	FAIL	Admin limit exceeded
12	SLDAP_NO_CRT_EXT	FAIL	Unavailable critical extension
13	SLDAP_CONF_REQ	FAIL	Confidentiality required
14	SLDAP_SASL_INPRG	FAIL	SASL bind in progress
16	SLDAP_NO_ATTR	FAIL	NO such attribute
17	SLDAP_UNDEF_ATTR	FAIL	Undefined attribute type
18	SLDAP_NO_MATCH	FAIL	Inappropriate matching
19	SLDAP_CONS_VIOL	FAIL	Constraint violation
20	SLDAP_ATTRVAL_EXIST	FAIL	Attribute or value exists
21	SLDAP_INVLD_ATTR	FAIL	Invalid attribute syntax
32	SLDAP_NO_OBJ	FAIL	No such object
33	SLDAP_ALIAS_PROB	FAIL	Alias problem
34	SLDAP_INVLD_DN	FAIL	Invalid DN syntax
35	SLDAP_IS_LEAF	FAIL	Is leaf
36	SLDAP_DEREF_PROB	FAIL	Alias dereferencing problem
48	SLDAP_INAPP_AUTH	FAIL	Inappropriate authentication
49	SLDAP_INVLD_CRED	FAIL	Invalid credentials
50	SLDAP_INSUF_RIGHTS	FAIL	Insufficient access rights
51	SLDAP_BUSY	FAIL	Busy
52	SLDAP_UNAVAIL	FAIL	Unavailable
53	SLDAP_UNWIL_PERF	FAIL	Unwilling to perform
54	SLDAP_LOOP_DET	FAIL	Loop detect

Table 2–12 (Cont.) User Defined ASDL Exit Types

Search Pattern	User Type	Base Type	Description
64	SLDAP_NAM_VIOL	FAIL	Naming violation
65	SLDAP_OBJCLASS_VIOL	FAIL	Object class Violation
66	SLDAP_NOALLOW_ NOLEAF	FAIL	Not allowed on non leaf
67	SLDAP_NOALLOW_RDN	FAIL	Not allowed on RDN
68	SLDAP_ENTRY_EXISTS	FAIL	Entry already exist
69	SLDAP_MOD_PROHIBIT	FAIL	Object class modifications prohibited
71	SLDAP_AFFECTS_DSAS	FAIL	Affects multiple DSAs
80	SLDAP_OTHER	FAIL	Other
81	SLDAP_SVR_DOWN	FAIL	Server down
85	SLDAP_TIMEOUT	FAIL	LDAP Timeout
89	SLDAP_PARM_ERR	FAIL	Param error
91	SLDAP_CONN_ERR	FAIL	Connect error
92	SLDAP_NOT_SUPP	FAIL	LDAP not supported
93	SLDAP_CTRL_NOFND	FAIL	Control not found
94	SLDAP_NO_RESULTS	FAIL	No results returned
95	SLDAP_MORE_RSLTS	FAIL	More results to return
96	SLDAP_CLI_LOOP	FAIL	Client loop
97	SLDAP_REFLT_EXCEED	FAIL	Referral limit exceeded
99992	SLDAP_PROV_EXCEPTION	FAIL	Exception due to cartridge validation errors
99991	SLDAP_NF_EXCEPTION	FAIL	Number format exception
99990	SLDAP_IO_EXCEPTION	FAIL	IO exception
99995	SLDAP_EXCEPTION	FAIL	General exception
NO_UDET_MATCH	SLDAP_NO_UDET_MATCH	FAIL	No user exit type matched

UserExitType.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceModel xmlns="http://www.metasolv.com/ServiceActivation/2003/ServiceModel">
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>5-2<//softwareLoad>
               <technology>LDAP<//technology>
               <neVendor>SUNW<//neVendor>
          <//neDescriptor>
          <searchPattern>0<//searchPattern>
          <userType>SLDAP_SUCCESS<//userType>
          <baseType>SUCCEED<//baseType>
          <description>Success<//description>
     <//userDefinedExitType>
     ..................
     <//serviceModel >
```

3

Service Definition

The SUN_LDAP_5_2_SUB_1_0 Cartridge cartridge contains a set of CSDLs that map to one or more ASDL commands. You can also create additional CSDLs that map to existing and newly-created ASDLs. An upstream system can assemble any of these CSDL commands onto a work order for provisioning.

This chapter presents detailed information about the CSDL parameters in this cartridge. The following table lists and describes the type of parameter information that is included.

Table 3–1 ASDL Parameter Information

Item	Description
Parameter Name	Identifies the parameter that is configured for the stated service.
Description	Describes the parameter.
Range	Describes or lists the range of values that can be used to satisfy this parameter.
Default Value	Configures a default value for the parameter so that it is not mandatory for the upstream system to provide a value.
Type	Indicates one of the following parameter types: ■ S - Scalar, specifies the parameter label transmitted on the ASDL command. Scalar parameters are conventional name-value pair parameters. ■ C - Compound, specifies the base name of the compound parameter transmitted on the ASDL command. A compound parameter contains structures or arrays of information that are represented by a particular structure name or compound parameter name. Each compound parameter can contain a large number of elements. If you use compound parameters, you only require a single entry in the ASAP translation tables to call the compound parameter and all its associated parameter elements. ■ I - Indexed, identifies a parameter that contains a sequential numerical index value to tell the SARM that it should execute the same operation (for example, an ASDL command) for all occurrences of that index. Consequently, if there are several options on a particular CSDL command (OPT1, OPT2, OPT3, etc.), you can specify the OPT parameter as an indexed parameter. When you specify the OPT parameter as an indexed parameter, the SARM generates several occurrences of that same ASDL command and each command has a different value for the option being transmitted to the NEP. For more information on parameter types, refer to the *ASAP Developer's Guide*.

Table 3–1 (Cont.) ASDL Parameter Information

Item	Description
Class	Indicates one of the following parameter classifications: ■ R - Required scalar parameter ■ O - Optional scalar parameter ■ C - Required compound parameter ■ N - Optional compound parameter ■ M - Mandatory indexed parameter ■ I - Optional indexed parameter ■ S - Parameter count

For a detailed description of the Required and Optional parameter classifications, refer to the *ASAP System Administrator's Guide.*

CSDL Commands

This cartridge provides the following CSDL commands:

- C_SUNW-LDAP_5-2_ADD_ENTRY
- C_SUNW-LDAP_5-2_DELETE_ENTRY
- C_SUNW-LDAP_5-2_MODIFY_DN
- C_SUNW-LDAP_5-2_MODIFY_ENTRY
- C_SUNW-LDAP_5-2_QUERY_DN
- C_SUNW-LDAP_5-2_QUERY_SCHEMA
- C_SUNW-LDAP_5-2_QUERY_WITH-SEARCH-FILTER

C_SUNW-LDAP_5-2_ADD_ENTRY

Add an entry to the directory given its DN, object class and attributes.

Table 3–2 C_SUNW-LDAP_5-2_ADD_ENTRY

Parameter Name	Description	Range	Default Value	Type	Class
ATTRIBUTE	This compound parameter contains the attribute types and values of the attributes of the entry to be added. For example ATTRIBUTE[1].TYPE subtoken contains the attribute type and ATTRIBUTE[1].VALUE subtoken contains the attribute values.	N/A	N/A	C	R
DN	DN of the entry to be added	N/A	N/A	S	R
NE_ID_SUNW-LDAP_5-2	NE Logical ID	N/A	N/A	S	R

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 3–3 CSDL to ASDL Mapping

CSDL	ASDL
C_SUNW-LDAP_5-2_ADD_ENTRY	A_SUNW-LDAP_5-2_ADD_ENTRY

C_SUNW-LDAP_5-2_DELETE_ENTRY

Delete an entry from the directory given its DN.

Table 3–4 C_SUNW-LDAP_5-2_DELETE_ENTRY

Parameter Name	Description	Range	Default Value	Type	Class
DN	DN of the entry to be deleted	N/A	N/A	S	R
NE_ID_SUNW-LDAP_5-2	NE Logical ID	N/A	N/A	S	R

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 3–5 CSDL to ASDL Mapping

CSDL	ASDL
C_SUNW-LDAP_5-2_DELETE_ENTRY	A_SUNW-LDAP_5-2_DELETE_ENTRY

C_SUNW-LDAP_5-2_MODIFY_DN

Modify DN of an entry in the directory.

Table 3–6 C_SUNW-LDAP_5-2_MODIFY_DN

Parameter Name	Description	Range	Default Value	Type	Class
DELETE_OLD_RDN	If true, the old RDN value is deleted. Possible values : true or false	N/A	N/A	S	R
DN	DN of the entry to be modified	N/A	N/A	S	R
NEW_PARENT_DN	DN of the new parent under which the entry to be relocated	N/A	N/A	S	O
NEW_RDN	New RDN of the entry	N/A	N/A	S	R
NE_ID_SUNW-LDAP_5-2	NE Logical ID	N/A	N/A	S	R

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 3–7 CSDL to ASDL Mapping

CSDL	ASDL
C_SUNW-LDAP_5-2_MODIFY_DN	A_SUNW-LDAP_5-2_MODIFY_DN

C_SUNW-LDAP_5-2_MODIFY_ENTRY

Modify an entry(add or delete attribute, attribute values, replace an attribute value etc) in the directory.

Table 3–8 C_SUNW-LDAP_5-2_MODIFY_ENTRY

Parameter Name	Description	Range	Default Value	Type	Class
ATTRIBUTE	This compound parameter contains for each attribute to be modified, the modification type, attribute type and values. For example ATTRIBUTE[1].MODIFY_ACTION contains the modification type i.e ADD, DELETE or REPLACE , ATTRIBUTE[1].TYPE contains the attribute type and ATTRIBUTE[1].VALUE contains the attribute values.	N/A	N/A	C	R
DN	DN of the entry to be modified	N/A	N/A	S	R
NE_ID_SUNW-LDAP_5-2	NE Logical ID	N/A	N/A	S	R

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 3–9 CSDL to ASDL Mapping

CSDL	ASDL
C_SUNW-LDAP_5-2_MODIFY_ENTRY	A_SUNW-LDAP_5-2_MODIFY_ENTRY

C_SUNW-LDAP_5-2_QUERY_DN

Query LDAP directory for a given DN(Distinguished name). Equivalent to reading the entry with a given DN.

Table 3–10 C_SUNW-LDAP_5-2_QUERY_DN

Parameter Name	Description	Range	Default Value	Type	Class
DN	Distinguished name of the entry to be queried.	N/A	N/A	S	R
NE_ID_SUNW-LDAP_5-2	NE Logical ID	N/A	N/A	S	R

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 3–11 CSDL to ASDL Mapping

CSDL	ASDL
C_SUNW-LDAP_5-2_QUERY_DN	A_SUNW-LDAP_5-2_QUERY_DN

C_SUNW-LDAP_5-2_QUERY_SCHEMA

Query the LDAP schema for all object classes, attribute types and matching rules.

Table 3–12 C_SUNW-LDAP_5-2_QUERY_SCHEMA

Parameter Name	Description	Range	Default Value	Type	Class
NE_ID_SUNW-LDAP_5-2	NE Logical ID	N/A	N/A	S	R

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 3–13 CSDL to ASDL Mapping

CSDL	ASDL
C_SUNW-LDAP_5-2_QUERY_SCHEMA	A_SUNW-LDAP_5-2_QUERY_SCHEMA

C_SUNW-LDAP_5-2_QUERY_WITH-SEARCH-FILTER

Query the LDAP directory for all matching entries of a search filter.

Table 3–14 C_SUNW-LDAP_5-2_QUERY_WITH-SEARCH-FILTER

Parameter Name	Description	Range	Default Value	Type	Class
BASE_DN	The Base DN from where the search should start.	N/A	N/A	S	R
NE_ID_SUNW-LDAP_5-2	NE Logical ID	N/A	N/A	S	R
SEARCH_FILTER	Search filter string. For example (cn=Jensen).	N/A	N/A	S	R
SEARCH_SCOPE	Possible values are:SCOPE_BASE,SCOPE_ONE,SCOPE_SUB	N/A	N/A	S	R

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 3–15 CSDL to ASDL Mapping

CSDL	ASDL
C_SUNW-LDAP_5-2_QUERY_WITH-SEARCH-FILTER	A_SUNW-LDAP_5-2_QUERY_WITH-SEARCH-FILTER

4

Configuring ASAP to Support Additional NE Instances

You can configure Oracle Communications ASAP (ASAP) to support the SUNW-LDAP_5-2_HOST - NEP configuration using the Service Activation Configuration Tool (SACT). Refer to the *ASAP System Administrator's Guide* for more information.

Extracting Source Files

Before you can access an XML file to modify it, you must extract it from the .sar file. Use the following procedure to extract source files from the sar file.

To extract source files:

1. If necessary, create a repository directory, copy the .sar file to the new directory and un-jar the sar file.
2. After you un-jar the sar file, you can access the XML files.

Loading a New XML File

When you finish modifying an XML file, you must create a new sar file, then restart the cartridge using the new file.

Configuration XML File

Below is an example of the Activation.Configuration.XML file for the SUN_LDAP_5_2_SUB_1_0 Cartridge cartridge.

```
<?xml version="1.0" encoding="UTF-8"?>
<activationConfig
xmlns="http://www.metasolv.com/ServiceActivation/2003/ActivationConfig"
xmlns:cfg="http://www.mslv.com/studio/activation/model/config"
xmlns:route="http://www.mslv.com/studio/activation/model/routing"
xmlns:sm="http://www.metasolv.com/ServiceActivation/2003/ServiceModel"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <connectionPool name="SLDAPPOL">
          <device name="sun_ldap_5-2_ldap_dev1">
               <environment>DEVELOPMENT</environment>
               <lineType>LDAP_CONNECTION</lineType>
          </device>
     </connectionPool>
     <element name="SUNW-LDAP_5-2_HOST">
          <vendor>SUNW</vendor>
```

```
          <technology>LDAP</technology>
          <softwareLoad>5-2</softwareLoad>
          <nepServerName>$NEP</nepServerName>
          <primaryPool>SLDAPPOL</primaryPool>
          <maximumConnections>1</maximumConnections>
          <dropTimeout>2</dropTimeout>
          <spawnThreshold>10</spawnThreshold>
          <killThreshold>8</killThreshold>
          <routingElement name="SUNW-LDAP_5-2_HOST"/>
          <communicationParameter>
               <label>HOST_IPADDR</label>
               <value>
                    <value>213.46.231.219</value>
               </value>
               <description>The host name or IP Address of theremote
NE.</description>
               <lineType>LDAP_CONNECTION</lineType>
          </communicationParameter>
          <communicationParameter>
               <label>PORT</label>
               <value>
                    <value>5005</value>
               </value>
               <description>Port number to connect on remote  NE
host.</description>
               <lineType>LDAP_CONNECTION</lineType>
          </communicationParameter>
          <communicationParameter>
               <label>OPEN_TIMEOUT</label>
               <value>
                    <value>20</value>
               </value>
               <description>Connection timeout in seconds.</description>
               <lineType>LDAP_CONNECTION</lineType>
          </communicationParameter>
          <communicationParameter>
               <label>READ_TIMEOUT</label>
               <value>
                    <value>30</value>
               </value>
               <description>Read timeout in seconds.</description>
               <lineType>LDAP_CONNECTION</lineType>
          </communicationParameter>
          <communicationParameter>
               <label>HOST_USERID</label>
               <value>

<value>mail=metasolv@lab.chello.com,ou=administration,dc=lab,dc=chello,dc=com</val
ue>
               </value>
               <description>Bind DN</description>
               <lineType>LDAP_CONNECTION</lineType>
          </communicationParameter>
          <communicationParameter>
               <label>HOST_PASSWORD</label>
               <value>
                    <value>MeTaSoLv</value>
               </value>
               <description>Bind Password</description>
               <lineType>LDAP_CONNECTION</lineType>
```

```
          </communicationParameter>
          <communicationParameter>
               <label>LDAP_VERSION</label>
               <value>
                    <value>3</value>
               </value>
               <description>LDAP protocol version.</description>
               <lineType>LDAP_CONNECTION</lineType>
          </communicationParameter>
          <communicationParameter>
               <label>SIZELIMIT</label>
               <value>
                    <value>500</value>
               </value>
               <description>Maximum no of search results to be
returned.</description>
               <lineType>LDAP_CONNECTION</lineType>
          </communicationParameter>
          <communicationParameter>
               <label>RESPONSELOG</label>
               <value>
                    <value>TRUE</value>
               </value>
               <description>Enable/Disable response log</description>
               <lineType>LDAP_CONNECTION</lineType>
          </communicationParameter>
          <communicationParameter>
               <label>DELIMITER</label>
               <value>
                    <value>|</value>
               </value>
               <description>Delimiter to be used between values for an
attribute</description>
               <lineType>LDAP_CONNECTION</lineType>
          </communicationParameter>
     </element>
</activationConfig>
```