

Chapter 14

Distributing Directory Changes to SDX Components

This chapter provides information about the directory eventing system (DES). It contains the following sections:

- Overview of the Directory Eventing System on page 163
- Configuring JNDI Properties for the Directory Eventing System on page 164
- Extending the Directory Eventing System for SDX Components on page 165
- Identifying the Type of Directory on page 169
- Managing Directory Problems on page 170
- Directory Eventing System with Large Directories on page 171

Overview of the Directory Eventing System

You can configure a DES for an SDX component that connects to one or more directories. DES provides two functions:

- Automatic notification of changes in the directory

DES polls the directory periodically to determine changes that affect the configuration or operation of a particular component. If DES finds relevant changes, it automatically provides the changes to the component. However, if DES does not find relevant changes, it does not provide any information.

- Redundancy

You must define a primary directory for SDX components that require access to a directory. You can also define a list of secondary (backup) directories.

DES detects when a connection to the primary directory fails, and:

1. Connects to the first available secondary directory in the specified list.
2. Reverts to the primary directory when it becomes available.

If a connection to a secondary directory fails, DES:

1. Connects to the primary directory if it is available.
2. If the primary directory is unavailable, connects to the first available directory in the specified list.

DES is not a central service for all SDX components; rather, you configure a DES for an individual SDX component. The SDX component determines the format of the DES properties; each DES property begins with a variable `<connectionPrefix>`. This variable is a property prefix that depends on the SDX component and the directory to which it connects. See the documentation for a specific SDX component for information about this variable.

Some components have connections to multiple directories; consequently you must configure DES properties for each connection. For example, the SAE may use different directories for service, configuration, and subscriber information.

Configuring JNDI Properties for the Directory Eventing System

DES is a Java Naming and Directory Interface (JNDI)-compliant service and accepts standard JNDI properties. For more information about JNDI, see <http://java.sun.com/products/jndi/>.

Standard DES properties have the format:

`<connectionPrefix> . <standardJNDISuffix>`

The variable `<connectionPrefix>` is a property prefix that depends on the SDX component and the directory to which it connects. The variable `<standardJNDISuffix>` is a standard JNDI property.

For example, the property `net.juniper.smgmt.des.retry_interval` is a standard JNDI property that specifies the how often the DES for the NIC agent polls the directory.

If you do not specify values for the standard DES properties, DES accepts the default values. The following list shows the `<standardJNDISuffix>` variables for the most common standard JNDI properties that you may want to customize for an SDX component.

.java.naming.provider.url

- URL of the primary directory.
- Value—URL in the format `ldap:// <host> :389`
 - `<host>` —IP address or name of directory host
- Example—`ldap://127.0.0.1:389/`

.java.naming.security.principal

- Distinguished name (DN) of the directory entry that defines the username with which the SDX component accesses the directory.
- Value— < DN >
- Example—*cn = nic, ou = Components, o = Operators, < base >*

.java.naming.security.credentials

- Password with which the SDX component accesses the directory.
- Value— < password >
- Guidelines—The password can be encoded in base64 and not visible in plain text. To use an encoded value, use the format {BASE64} < encoded-value > .
- Example—admin

.java.naming.security.protocol

- Security protocol (SSL) for the connection.
- Value—ssl

.java.naming.factory.initial

- Name of the Java factory class from which the SDX software creates the LDAP initial context.
- Value—Path to Java factory
- Example—*net.juniper.smgmt.lib.des.DESInitialContextFactory*

Extending the Directory Eventing System for SDX Components

The SDX software defines a number of DES properties that extend the standard set. These DES properties have the format:

< connectionPrefix > .des. < propertySuffix >

The variable < connectionPrefix > is a property prefix that depends on the SDX component and the directory to which it connects. The variable < propertySuffix > depends on the DES property.

For example, the property *net.juniper.smgmt.des.enable_eventing* is a property that specifies whether the DES for the NIC agent polls the directory periodically.

The following list describes the < propertySuffix > variables for the DES properties that you can configure for SDX components.

enable_eventing

- Specifies whether the SDX component polls the directory for changes.
- Value
 - True—SDX component polls the directory for changes.
 - False—SDX component does not poll the directory for changes.

pollinginterval

- Time interval at which the SDX component polls the directory.
- Value—Number of seconds in the range 15–2147483647

event_baseDN

- DN of an entry superior to the data associated with this SDX component in the directory.
- Value—*o = < DN > , < base >*
 - *< DN >*—DN of superior entry
- Guidelines—If you are storing non-SDX data in the directory, and that data changes frequently whereas the SDX data does not, you may need to adjust the default value to improve performance. For optimal performance, set the value to the DN of an entry superior to both the SDX data and the changing non-SDX data.
- Default—*o = umc, < base >*

delegate_factory_initial

- Value used by an SDX internal process.
- Value—SDX software sets the value automatically



CAUTION: Do not change this value.

connection_pool_size

- Number of directory connections that DES uses.
- Value—1



CAUTION: Do not change this value.

dispatcher_pool_size

- Number of events that the SDX component can receive from the directory simultaneously.
- Value—Integer in the range 1–2147483647



CAUTION: Some SDX components require a specific value for this property. See the documentation for the component to determine whether you can change this value.

connection_manager_id

- DES connection manager within the JNDI framework.
- Value—Text string
- Example—DIRAGENT_POOL_VR

fake_delete

- Specifies how DES tracks objects deleted from the directory.
- Value—SDX software sets the value automatically



CAUTION: Do not change this value.

show_fake_delete

- Specifies whether you can view the objects deleted from the directory.
- Value
 - True—Deleted objects are visible.
 - False—Deleted objects are not visible.
- Default—False



CAUTION: Do not change this value.

share_connection

- Specifies whether other SDX components running in the same process as this SDX component share a connection to the directory with this SDX component.
- Value—
 - True—SDX components share the connection.
 - False—SDX components do not share the connection.



CAUTION: Do not change this value.

backup_provider

- List of redundant directories.
- Value—List of URLs separated by semicolons; URLs have the format `ldap:// <host> :389`
 - <host> —IP address or name of the directory host
- Example—`ldap://127.0.0.1:389/; ldap://127.0.0.2:389/`

enable_sysman

- Specifies whether the SDX SNMP agent exports MIBs for this directory connection.
- Value
 - True—SNMP agent exports MIBs.
 - False—SNMP agent does not export MIBs.

connect.timeout

- Maximum time that DES waits for the directory to respond.
- Value—Number of seconds in the range 1–2147483647

retry_interval

- Time interval at which DES attempts to connect to the directory.
- Value—Number of seconds in the range 10–2147483647

connectcheck_interval

- Time interval at which DES verifies its connection to the directory.
- Value—Number of seconds in the range 15–2147483647

signatureDN

- DN of the directory entry that specifies the usedDirectory attribute. The usedDirectory attribute identifies the type of directory, such as openLDAP or DirX, to which the SDX software is connected. For information about this attribute, see the LDAP schema files in the SDX software distribution in the directory *SDK/doc/ldap* or on the Juniper Networks Web site at

<http://www.juniper.net/techpubs/software/management/sdx>

For information about setting this property, see *Identifying the Type of Directory* on page 169.

If the value of signatureDN is not the DN of a directory entry or is the DN of an entry that does not have a usedDirectory attribute, the SDX software logs an error and proceeds as it would for directory types other than DirX. If the value of the usedDirectory attribute does not correspond to a type of directory that the SDX software supports, the SDX software logs an error and proceeds as it would for directory types other than DirX.

- Value— < DN >
- Default—GlobalUserDatabase.server.signatureDN = o = umc
- Example—GlobalUserDatabase.server.signatureDN = o = SDX, o = Juniper, o = Applications

Example

```
java.naming.security.principal = cn=nic,ou=Components,o=Operators,<base>
java.naming.security.credentials = {BASE64}bmlj
java.naming.provider.url = ldap://127.0.0.1:389/
java.naming.factory.initial=net.juniper.smgmt.lib.des.DESInitialContextFactory
net.juniper.smgmt.des.enable_eventing = true
net.juniper.smgmt.des.delegate_factory_initial = com.sun.jndi.ldap.LdapCtxFactory
net.juniper.smgmt.des.connection_pool_size = 1
net.juniper.smgmt.des.connection_manager_id = DIRAGENT_POOL_VR
net.juniper.smgmt.des.dispatcher_pool_size = 1
net.juniper.smgmt.des.fake_delete = true
net.juniper.smgmt.des.show_fake_delete = false
net.juniper.smgmt.des.directory_init_delta = 2592000
net.juniper.smgmt.des.polling_interval = 30
```

```

net.juniper.smgd.des.share_connection=true
net.juniper.smgd.des.event_baseDN = <base>
net.juniper.smgd.des.enable_sysman = false
net.juniper.smgd.des.connect.timeout = 10
net.juniper.smgd.des.retry_interval = 30
net.juniper.smgd.des.connectioncheck_interval = 60
net.juniper.smgd.des.signatureDN = o=umc

```

Identifying the Type of Directory

The SDX software includes a DES property called `signatureDN` that identifies the DN of the entry that specifies the LDAP schema attribute `usedDirectory`. This attribute identifies the type of directory, such as `openLDAP` or `DirX`, to which the SDX software connects. For information about this attribute, see the LDAP schema files in the SDX software distribution in the directory *SDK/doc/ldap* or on the Juniper Networks Web site at

<http://www.juniper.net/techpubs/software/management/sdx>

Identifying the type of directory allows the SDX software to accommodate the different ways that different directories process DES queries, and enables more efficient retrieval of information. In particular, this feature offers benefits for the following tasks:

- Checking whether an object in the directory has not been deleted
- Finding new entries in the directory

If you load the LDAP schema from the SDX software distribution, the SDX software automatically sets the `usedDirectory` attribute for the type of directory to which it connects. If you use this LDAP schema as the structure for your directory, you can use the default value (`o = umc`) for the `signatureDN` property, and you do not need to configure the type of directory.

However, if you use a customized LDAP schema rather than the provided LDAP schema, use the following procedure to allow the SDX software to determine the type of directory:

1. Choose the entry that specifies the `usedDirectory` attribute.
2. Specify a value for the `usedDirectory` attribute.
3. In the property file of the SDX component that connects to this directory, set the `signatureDN` property to the DN of the entry with the `usedDirectory` attribute for the `signatureDN` property.

For example, use SDX Configuration Editor or SDX Admin to configure DES properties for the SAE.

4. Repeat Steps 1 to 3 for each DES connection.

Managing Directory Problems

When an SDX component communicates with the directory, that component may pass a time (known as a server timeout) to the directory to specify a time limit for the directory to respond. If the directory is not working correctly, however, it may not respond during this time, and will cause the SDX component to stop operating.

DES recovers if the directory is not working correctly. In addition, you can configure DES to prohibit communications with a directory if that directory repeatedly fails to respond. If you do so, DES starts the following procedure for all communication with the directory:

1. Assigns a client timeout to the communication.
 The client timeout exceeds the server timeout.
2. If the directory does not respond during this time, DES closes the connection to the directory.
3. DES tries to reconnect to the directory and proceeds as follows:
 - If DES cannot connect to the directory, it connects to the next available directory specified by the DES redundancy properties.
 - If DES can connect to the directory, it contacts the directory again and repeats Steps 1 to 3.
4. If a directory fails to respond 10 times, DES prevents further communication with the directory.

To enable DES to prevent connection to a directory that repeatedly fails to respond, configure the `enable_blacklist` property.

enable_blacklist

- Specifies whether DES prevents connection to a directory if the directory fails to respond during 10 polls.
- Value
 - True—DES prevents connection to the directory.
 - False—DES does not prevent connection to the directory.
- Default—False

If DES prevents connection to a directory, do the following to reestablish the connection to the directory.

1. Fix the problem with the directory.
2. Restart the SDX component that communicates with this directory.

Directory Eventing System with Large Directories



CAUTION: Do not use directory eventing for an OpenLDAP directory with more than 1,000 subtrees.

OpenLDAP does not support some of the features required for DES to operate with large directories. If you enable directory eventing for large OpenLDAP directories, performance will be poor.

