

Chapter 2

Configuring the SAE

This chapter describes how to configure SAE properties. It contains the following sections:

- Overview on page 51
- Configuring LDAP Access to Directory Data on page 52
- Configuring Router Properties on page 64
- Configuring the License Manager on page 81
- Configuring Interim Accounting on page 84
- Allowing Multiple Logins from the Same IP Address on page 86
- Authenticating Registered Username/Password Pairs on page 86
- Configuring Timers for Session Reactivation on page 87
- Configuring the Number of Threads for Sessions on page 88
- Reducing the Size of Session Objects on page 88
- Modifying the SAE Property File on page 89
- Other Configuration Tasks on page 90

Overview

The SAE property file contains SAE configuration data that is stored in the directory. For example, it contains the configurations of plug-ins, the directory eventing system (DES), router access, LDAP access, the external SAE interface, logging, and the license manager.

You can modify the SAE property file with SDX Configuration Editor, SDX Admin, or a standard text editor. The following sections show how to configure SAE properties with SDX Configuration Editor. Each field description includes a property name, which is used if you modify the properties with SDX Admin or a text editor.

SDX Configuration Editor organizes properties into tabs. This chapter describes how to configure the properties in all of the tabs, except the following:

- Logging—See *Chapter 10, Configuring Logging for SDX Components*
- Plug-ins—See *Chapter 5, Configuring Authorization and Accounting Plug-Ins*
- RADIUS—See *Chapter 5, Configuring Authorization and Accounting Plug-Ins*
- File-Acct Template—See *Chapter 5, Configuring Authorization and Accounting Plug-Ins*
- Ext. Interface—See the documentation for the SAE CORBA remote API, see the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/sdx/api-index.html> or in the SDX software distribution in *SDK/doc/idl/sae/html/index.html*.
- Miscellaneous—Interim accounting and login behavior are covered in this chapter; global UDP plug-in port is covered in *Configuring UDP Ports for RADIUS Plug-Ins* on page 182; assigned IP sessions are covered in *SDX Software Basics Guide, Chapter 18, Providing VoIP Services in the SDX Network*; time-based policies and aggregate services are covered in the *SDX Objects Guide, Chapter 1, Managing Services*.

Configuring LDAP Access to Directory Data

The SDX software stores subscriber (user), service, persistent login, policy, router, and cached subscriber profiles and session data in a directory. The SAE uses LDAP to store and retrieve the data. You can configure the LDAP connections to the directories in which this data is stored. You can also select the filter that the SAE uses to search for subscriptions in the directory and directory eventing parameters for data stored in the directory.

To use SDX Configuration Editor to configure SAE properties for LDAP connections, select a directory configuration object for the SAE that you want to configure, and then select the LDAP tab.

Configuring Access to Subscriber Data

The subscriber data configuration defines the LDAP connection from the SAE to the directory in which subscriber data is stored.

User Data	
Server Address	127.0.0.1 <input type="button" value="Disable"/>
Search Base	o=Users, <base>
Authentication DN	cn=ssp,ou=Components,o=Operators,<base> <input type="button" value="Disable"/>
Password	*** <input type="button" value="Show"/> <input type="button" value="Disable"/>
Enable Directory Eventing	Yes <input type="button" value="v"/>
Directory Polling Interval [s]	30
Secured LDAP protocol	LDAPS <input type="button" value="v"/> <input type="button" value="Disable"/>
Filter for loading subscriptions	Subscription Objectclass Filter <input type="button" value="v"/>

Server Address

- Disables or enables and identifies the directory server that stores subscriber information.
- Value—IP address or hostname; use a space to separate addresses for multiple directory servers: 127.153.27.1 192.168.0.1
- Default—Disabled
- Property name—UserDataSource.repository.ldap.server.address

Search Base

- Subtree in the directory in which subscriber information is stored. When a subscriber logs in to a residential service selection portal, the SAE searches subscriber profiles by mapping the realm of the login name to a retailer object found below the search base.
- Value— < DN >
You can use the special value < base > to refer to the globally configured base distinguished name (DN).
- Guidelines—Sensible values include *o = Users*, *o = umc* for multidomain support and *retailerName = Retailer*, *o = Users*, *o = umc* for single domain support.
- Default—*o = Users*, < base >
- Property name—UserDataSource.repository.ldap.server.base.dir

Authentication DN

- Disables or enables and sets the DN that the SAE uses to authenticate access to the directory server. The specified directory entry must exist and have read access to all attributes. The entry must have write access if subscribers are allowed to customize their subscription profiles.
- Value— <DN >
You can use the special value <base > to refer to the globally configured base DN.
- Default—Disabled, which means that the value configured for the directory is used
- Property name—UserDataSource.repository.ldap.server.authDN

Password

- Disables or enables and sets the password used to authenticate access to the directory server. You must configure the password in the directory to authenticate read-access to the directory.
- Value—Text string or base64 string that matches the value of the userPassword attribute of the authentication DN
- Default—Disabled, which means that the value configured for the directory is used
- Property name—UserDataSource.repository.ldap.server.password

Enable Directory Eventing

- Enables or disables automatic discovery of changes in subscriber profiles.
- Value
 - Yes—Changes in the subscriber profile or subscriptions take effect automatically while the subscriber is logged in.
 - No—Changes in the subscriber profile or subscriptions do not take effect until the next time the subscriber logs in.
- Default—Yes
- Property name—UserDataSource.repository.ldap.server.des.enable_eventing

Directory Polling Interval [s]

- Sets the frequency for checking the directory for updates.
- Value—Number of seconds in the range 15–86400
- Default—30
- Property name—UserDataSource.repository.ldap.server.des.pollinginterval

Secured LDAP protocol

- Enables or disables LDAPS as the secure protocol for connections to the server that stores subscriber data.
- Value—Enable or Disable
- Default—Disable
- Property name—UserDataSource.repository.ldap.server.security.protocol

Filter for loading subscriptions

- Selects the filter that the SAE uses to search for subscriptions in the directory when the SAE loads a subscription.
- Value—Select one of the following values from the drop-down menu:
 - Subscriber reference filter—The SAE runs a search based on the subscriberRef attribute in the umcServiceProfile object, which is the base object class of the service profile hierarchy. The subscriberRef attribute contains a DN that points to the parent of the subscriber object.
 - Subscription Objectclass filter—The SAE performs a one-level search with the directory entry, which represents the subscriber folder as the base DN. The search filter is (objectClass = sspServiceProfile). This method can be slow if you have a large number of subscription entries within the subscriber folder subtree.
- Default—Subscription Objectclass filter
- Property name—UserDataSource.repository.ldap.server.loadSubscriptionFilter

Configuring Access to Service Data

The service data configuration defines the LDAP connection from the SAE to the directory in which service data is stored.

Service Data	
Server Address	127.0.0.1 Enable
Search Base	<base>
Authentication DN	cn=ssp,ou=components,ou=operators,base Enable
Password	*** Show Enable
Enable Directory Eventing	Yes ▼
Directory Polling Interval [s]	30
Secured LDAP protocol	LDAPS ▼ Enable

Server Address

- Disables or enables and identifies the directory server that stores service data.
- Value—IP address or hostname; use a space to separate addresses for multiple directory servers: 127.153.27.1 192.168.0.1
- Default—Disabled, which means that the value configured for the directory is used
- Property name—ServiceDataSource.repository.ldap.server.address

Search Base

- Subtree in the directory in which service information is stored. The SAE loads service definitions on startup and when service reloading is requested.
- Value— < DN >
You can use the special value < base > to refer to the globally configured base DN.
- Default— < base >
- Property name—ServiceDataSource.repository.ldap.server.base.dir

Authentication DN

- Disables or enables and sets the DN that the SAE uses to authenticate access to the directory server. The specified directory entry must exist and have read access to all attributes.
- Value— < DN >
You can use the special value < base > to refer to the globally configured base DN.
- Default—Disabled, which means that the value configured for the directory is used
- Property name—ServiceDataSource.repository.ldap.server.authDN

Password

- Disables or enables and sets the password used to authenticate access to the directory server. You must configure the password in the directory to authenticate read access to the directory.
- Value—Text string or base64 string
- Default—Disabled, which means that the value configured for the directory is used
- Property name—ServiceDataSource.repository.ldap.server.password

Enable Directory Eventing

- Enables or disables automatic discovery of changes in service definitions.
- Value
 - Yes—Changes in service definitions take effect automatically. If a changed service is in use, all service instances are deactivated and then reactivated with the modified settings. Consequently, service may be affected for subscribers who are logged in at the time of the modification.
 - No—Changes in service definitions do not take effect until the SAE is restarted or you reload services using SAE Web Admin.
- Default—Yes
- Property name—ServiceDataSource.repository.ldap.server.des.enable_eventing

Directory Polling Interval [s]

- Sets the frequency for checking the directory for updates.
- Value—Number of seconds in the range 15–86400
- Default—30
- Property name—ServiceDataSource.repository.ldap.server.des.pollinginterval

Secured LDAP protocol

- Enables or disables LDAPS as the secure protocol for connections to the server that stores service data.
- Value—Enable or Disable
- Default—Disable
- Property name—ServiceDataSource.repository.ldap.server.security.protocol

Configuring Access to Policy Data

The policy data configuration defines the LDAP connection from the SAE to the directory in which policy data is stored.

Policy Data	
Policy Search Base	<input type="text" value="o=Policies, <base>"/>
Parameter Search Base	<input type="text" value="o=Parameters, <base>"/>
Enable Directory Eventing	<input type="text" value="Yes"/>
Directory Polling Interval [s]	<input type="text" value="30"/>

Policy Search Base

- Subtree in the directory that stores policy data.
- Value— < DN >
You can use the special value < base > to refer to the globally configured base DN.
- Default—*o = Policies, < base >*
- Property name—PolicyDataSource.repository.ldap.baseDN

Parameter Search Base

- Subtree in the directory that stores policy parameter data.
- Value— < DN >
You can use the special value < base > to refer to the globally configured base DN.
- Default—*o = Parameters, < base >*
- Property name—PolicyDataSource.repository.ldap.parameterBaseDN

Enable Directory Eventing

- Enables or disables automatic discovery of changes in policy definitions and in interface classifiers.
- Value
 - Yes—Changes in policy definitions take effect automatically. If a changed policy is in use, all policy instances are deactivated and then reactivated with the modified settings. Consequently, service may be affected for subscribers who are logged in when the change is made.
 - No—Changes in policy definitions do not take effect until the SAE is restarted or you reload services using SAE Web Admin.
- Default—Yes
- Property name—net.juniper.smgmt.des.enable_eventing

Directory Polling Interval [s]

- Sets the frequency for checking the directory for updates.
- Value—Number of seconds in the range 15–86400
- Default—30
- Property name—net.juniper.smgmt.des.pollinginterval

Configuring Access to the Persistent Login Cache

The persistent login cache configuration defines the LDAP connection from the SAE to the directory where persistent login cache data is stored.

Server Address

- Disables or enables and identifies the directory server that stores persistent login data.
- Value—IP address or hostname; use a space to separate addresses for multiple directory servers: 127.153.27.1 192.168.0.1
- Default—Disabled, which means that the value configured for the directory is used
- Property name—UserCacheDataSource.repository.ldap.server.address

Search Base

- Subtree in the directory that stores persistent login cache data.
- Value— < DN >
You can use the special value < base > to refer to the globally configured base DN.
- Default—ou-authCache, < base >
- Property name—UserCacheDataSource.repository.ldap.server.base.dir

Authentication DN

- Disables or enables and sets the DN that the SAE uses to authenticate access to the directory server. The specified directory entry must exist and have read access to all attributes.
- Value— < DN >
You can use the special value < base > to refer to the globally configured base DN.
- Default—Disabled
- Property name—UserCacheDataSource.repository.ldap.server.authDN

Password

- Disables or enables and sets the password used to authenticate access to the directory server. You must configure the password in the directory to authenticate read access to the directory.
- Value—Text string or base64
- Default—ssp
- Property name—UserCacheDataSource.repository.ldap.server.password

Enable Directory Eventing

- Enables or disables automatic discovery of changes to the persistent login cache.
- Value—Yes or No
- Default—No
- Property name—
UserCacheDataSource.repository.ldap.server.des.enable_eventing

Directory Polling Interval [s]

- Sets the frequency for checking the directory for updates.
- Value—Number of seconds in the range 15–86400
- Default—30
- Property name—
UserCacheDataSource.repository.ldap.server.des.pollinginterval

Secured LDAP protocol

- Enables or disables LDAPS as the secure protocol for connections to the server that stores persistent login cache data.
- Value—Enable or Disable
- Default—Disable
- Property name—UserCacheDataSource.repository.ldap.server.security.protocol

Configuring Directory Eventing

The directory eventing configuration defines directory eventing parameters for data stored in the directory.

For more information about the directory eventing properties, see *Configuring SDX DES Properties* on page 267.

The screenshot shows a configuration panel titled "Directory Eventing". It contains the following fields:

- Timeout: [Text input field]
- Delegate Factory: [Text input field]
- Connection Pool Size: [Text input field]
- Dispatcher Pool Size: [Text input field]
- Fake Delete: [Text input field with a small icon to the right]
- Show Fake Delete: [Text input field with a small icon to the right]

Timeout

- Time that SDX component passes to the directory to specify a time limit for the directory to respond.
- Default—5000
- Property name—DataSource.repository.ldap.server.timeout

Delegate Factory

- Value used by an SDX internal process.
- Value—SDX software sets the value automatically. Do not change this value.
- Default—No value
- DataSource.repository.ldap.server.des.delegate_factory_initial

Connection Pool Size

- Number of directory connections that the DES uses.
- Default—1
- Value—Do not change this value.
- Property name—DataSource.repository.ldap.server.des.connection_pool_size

Dispatcher Pool Size

- Number of events that the SAE can receive from the directory simultaneously.
- Default—1
- Property name—DataSource.repository.ldap.server.des.dispatcher_pool_size

Fake Delete

- Specifies how the DES tracks objects from the directory.
- Value—Yes or No
- Default—Yes
- Property name—DataSource.repository.ldap.server.des.fake_delete

Show Fake Delete

- Specifies whether you can view object that have been deleted from the directory.
- Value—Yes or No
- Default—No
- Property name—DataSource.repository.ldap.server.des.show_fake_delete

Configuring the Location of Router, Persistent Login, and Persistent Session Data

You can also configure the location of router data, persistent login information for DHCP scenarios, and persistent session data.

Network Data Search Base	<input type="text" value="o=Network, <base>"/>
SAE Cache Repository Search Base	<input type="text" value="o=userProfileCache, <base>"/>
Persistent Session Cache repository search base	<input type="text" value="o=PersistentSessions, <base>"/>

Network Data Search Base

- Subtree in the directory that stores router data.
- Value— < DN >
You can use the special value < base > to refer to the globally configured base DN.
- Default—*o = Network, < base >*
- Property name—NetworkDataSource.repository.ldap.baseDN

SAE Cache Repository Search Base

- Base DN for storing and retrieving subscriber profiles. This is the directory subtree in which persistent login information is stored for DHCP scenarios.
- Value— < DN >
You can use the special value < base > to refer to the globally configured base DN.
- Default—*o = userProfileCache, < base >*
- Property name—`UserDataSource.repository.ldap.server.cache.dir`

Persistent Session Cache Repository Search Base

- Base DN for storing and retrieving persistent session information.
- Value— < DN >
You can use the special value < base > to refer to the globally configured base DN.
- Default—*o = PersistentSessions, < base >*
- Property name—`UserDataSource.repository.ldap.server.persistent.session`

Enabling Directory Eventing of Configuration Data

You can enable directory eventing of SAE configuration data.

The image shows a configuration interface with a label 'Enable Configuration Directory Eventing' followed by a dropdown menu. The dropdown menu is currently set to 'Yes' and has a small downward arrow icon on the right side.

Enable Configuration Directory Eventing

- Enables automatic discovery of changes in the SAE configuration data. If this property is enabled, the SAE detects changes in its configuration and reconfigures itself.
- Value—Yes or No
- Default—Yes
- Property name—`Main.auto-reconfigure`

Configuring Router Properties

To set up the SAE to manage Juniper Networks routers, you need to configure router drivers and other properties. This section covers the following tasks that you can use to configure router properties on the SAE:

- Configuring the JUNOS Router Driver on page 64
- Configuring the JUNOSe Router Driver on page 67
- Configuring a Simulated Router Driver on page 71
- Configuring Session Stores on page 72
- Configuring SNMP Communities on page 78
- Identifying a Profile for Unauthenticated Subscribers on page 78
- Configuring the Virtual Portal Address on page 79
- Configuring Router Initialization Scripts on page 79

To use SDX Configuration Editor to configure SAE properties for routers, select a directory configuration object for the SAE that you want to configure, and then select the Router tab.

For information about configuring a driver for PCMM devices, see *SDX Software Basics Guide, Chapter 15, Integrating SDX Software into a PCMM Environment*.

Configuring the JUNOS Router Driver

A JUNOS routing platform interacts with the SAE by using a JUNOS software process called `sdx`. When the `sdx` process establishes a TCP/IP connection to the SAE, the SAE begins to manage the router. The JUNOS router driver configuration defines parameters related to the interactions between the SAE and the `sdx` process. For information about configuring the `sdx` process, see *SDX Integration Guide, Chapter 2, Integrating JUNOS Routing Platforms*.

To configure a session store for a JUNOS router driver, see *Configuring Session Stores on page 72*.

JUNOS Router Driver	
BEEP Server Port	<input type="text" value="3333"/>
Connection Attempts	<input type="text"/>
Keepalive Interval [s]	<input type="text"/>
Message Timeout [ms]	<input type="text"/>
Batch Size	<input type="text"/>
Transaction Batch Time [ms]	<input type="text"/>
SDX Group Name	<input type="text"/>
SDX Session Group Name	<input type="text"/>
Send Commit Check	<input type="text"/>

BEEP Server Port

- Transmission Control Protocol (TCP) port number that is used to communicate with the sdx process on JUNOS routing platforms. This port number must match the port number configured in the sdx process on the router.
- Value—TCP port number
- Default—3333
- Property name—Router.junos.server_port

Connection Attempts

- Number of socket connection attempts that are accepted while the SAE creates sockets before new attempts are dropped.
- Value—Positive value greater than 0; if the value is equal to or less than 0, the default value is used
- Default—50
- Property name—Router.junos.backlog_connections

Keepalive Interval [s]

- Interval between keepalive messages sent from the router. The sdx process on the router monitors the connection to the SAE by sending keepalive messages at one-third the specified interval. If the sdx process does not receive the expected keepalive answer within the specified timeout, it closes the connection. A short interval results in a high load on the BEEP interface. A long interval results in a long time before a connection failure is detected.
- Value—Number of seconds in the range 0–2147483647. A value of 0 means that timeout is disabled.
- Default—45
- Property name—Router.junos.keepalive

Message Timeout [ms]

- Amount of time that the router driver waits for a response from the sdx process. Under a high load the router may not be able to respond fast enough to requests. Change this value only if a high number of timeout events appear in the error log.
- Value—Number of milliseconds in the range 0–2147483647
- Default—30000
- Property name—Router.junos.message_timeout

Batch Size

- Minimum number of service configuration transactions that are committed at the same time. If any of the transactions in a batch fails, all transactions are aborted, and the associated service activations or deactivations fail.
- Value—Integer in the range 0–2147483647
- Default—10
- Property name—Router.junos.batch_size

Transaction Batch Time [ms]

- Maximum time to collect configuration transactions in a batch. The batch is completed if either the batch size or the batch time is reached.
- Value—Number of milliseconds in the range 0–2147483647
- Default—2000
- Property name—Router.junos.batch_time

SDX Group Name

- Name of group on the JUNOS routing platform in which provisioning objects are stored.
- Value—Name configured on the JUNOS routing platform
- Default—sdx
- Property name—Router.junos.group.config

SDX Session Group Name

- Name of group on the JUNOS routing platform in which session objects are stored.
- Value—Name configured on the JUNOS routing platform
- Default—sdx-sessions
- Property name—Router.junos.group.session

Send Commit Check

- Enables or disables commit check. If enabled, a more detailed error message is logged if a batch fails, which allows you to verify individual transactions in a batch.
- Value—True or false
- Guidelines—To maximize service activation performance, commit check should be disabled.
- Default—True
- Property name—Router.junos.send_commit_check

Configuring the JUNOSe Router Driver

Use the following fields to configure the Common Open Policy Service (COPS) connection between the SAE COPS server and the COPS client in the JUNOSe router.

To configure a session store for a JUNOSe router driver, see *Configuring Session Stores* on page 72.

JUNOSe Router Driver	
COPS Server Port	3288
Backlog	50
Keepalive Interval [s]	45
Message Timeout [ms]	120000
COPS Message Maximum Length [bytes]	200000
COPS Message Read Buffer Size [bytes]	30000
COPS Message Write Buffer Size [bytes]	30000
Pending Address Timeout [ms]	5000
Number of COPS Handler Threads	20
Cached driver expiration	600
Drop Unmanaged Interfaces for the JUNOSe XDR Driver	No

COPS Server Port

- Port number of the SAE COPS server.
- Value—Port number that matches the configuration of the SDX client in the JUNOSe router
- Default—3288
- Property name—Router.junose.server_port

Backlog

- Number of connection attempts before connections are dropped.
- Value—Integer
- Default—50
- Property name—Router.junose.backlog_connections

Keepalive Interval [s]

- Interval between keepalive messages sent from the COPS client (the JUNOSE router). The COPS client monitors the COPS connection by sending keepalive messages at random intervals between one-fourth and three-fourths of the specified interval. If the client does not receive the expected keepalive answer within the specified timeout, the client terminates the connection.
- Value—Number of seconds in the range 0–32768. A value of 0 means that timeout is disabled.
- Guidelines—A short interval results in a high load on the COPS interface. A long interval results in a long time before a COPS failure is detected.
- Default—45
- Property name—Router.junose.keepalive

Message Timeout [ms]

- Timeout interval in which the COPS server waits for a response to COPS requests. Under a high load the router may not be able to respond fast enough to COPS requests. Change this value only if a high number of COPS timeout events appear in the error log.
- Value—Number of milliseconds
- Default—60000
- Property name—Router.junose.message_timeout

COPS Message Maximum Length [bytes]

- Maximum length of a COPS message.
- Value—Number of bytes in the range 4 bytes to 2 GB
- Guidelines—We recommend that you use the default setting.
- Default—200000
- Property name—Router.junose.message_max_length

COPS Message Read Buffer Size [bytes]

- Buffer size for receiving COPS messages from the JUNOSE client.
- Value—Number of bytes in the range 4 bytes to 2 GB
- Guidelines—We recommend that you use the default setting unless you are instructed to change it by Juniper Networks engineers.
- Default—30000
- Property name—Router.junose.message_read_buffer_size

COPS Message Write Buffer Size [bytes]

- Buffer size for sending COPS messages to the JUNOSe client.
- Value—Number of bytes in the range 4 bytes to 2 GB
- Guidelines—We recommend that you use the default setting unless you are instructed to change it by Juniper Networks engineers
- Default—30000
- Property name—Router.junos.message_write_buffer_size

Pending Address Timeout [ms]

- Maximum time that an address request remains pending.
- Value—Number of milliseconds
- Guidelines—Realistic values are in the range 1000–15000 (5 seconds to 15 seconds).
- Default—5000
- Property name—Router.junos.pending_address_timeout

Number of COPS Handler Threads

- Size of the thread pool for handling unsolicited messages. These threads are shared among all JUNOSe router drivers.
- Value—Number of threads
- Default—20
- Property name—Router.junos.handler_threads

Cache driver expiration

- Minimum amount of time to keep the state of a router driver after its COPS connection has been closed.
- Value—Number of seconds in the range 0–2147483647
- Default—600
- Property name—Router.junos.cachedDriverExpiration

Drop Unmanaged Interfaces for the JUNOSe XDR Driver

- Specifies whether or not the JUNOSe router driver keeps a record of unmanaged interfaces.
- Value
 - Yes—The router driver keeps a record of unmanaged interfaces.
 - No—The router driver does not keep a record of unmanaged interfaces. With this setting, next interface rules may not work properly.
- Default—No
- Property name—Router.junos.drop_unmanaged_xdr

Configuring the Length of Time MAC Addresses Remain in SAE Cache

When a DHCP subscriber transitions from an authenticated IP address to an unauthenticated IP address or vice-versa, the SAE:

1. Logs out the subscriber associated with the original IP address.
2. Caches the subscriber profile in the in-memory cache, indexed by the DHCP subscriber's MAC address.
3. Waits until the DHCP subscriber with the cached MAC address obtains its new IP address, and then logs in the subscriber and associates it with the new IP address.

The period during which the subscriber profile remains in the in-memory cache can last until the DHCP lease time for the original address. If something happens during this period—for example, the subscriber turns off the client computer—the subscriber profile remains in the SAE's in-memory cache forever. When a new IP address is assigned to the same DHCP client, problems can occur. To avoid such problems, entries in the in-memory cache are removed after a configurable amount of time. To configure this time period in SDX Configuration Editor, select a directory configuration object for the SAE that you want to configure, and then select the Router tab.

MAC Cache Expiration Time	1800
---------------------------	------

maxMacCacheEntryAge

- Amount of time that a subscriber profile remains in the SAE's in-memory cache.
- Value—Number of seconds in the range 0–2147483647
- Guidelines—Configure this parameter to be greater than the time required for a DHCP subscriber to transition from an unauthenticated IP address to an authenticated IP address or vice versa. The time required for a DHCP subscriber to transition from one IP address to another depends on the lease times configured on the JUNOSe router and the instructions given to the subscriber on the Web portal, such as reboot your PC now.
- Default—1800
- Property name—maxMacCacheEntryAge

Configuring a Simulated Router Driver

Simulated router drivers allow you to create subscriber sessions without connecting to a router. You can then use the simulated subscriber sessions to test SAE applications.

To use this feature, you configure a simulated router in the directory in the same way that you configure a real router; make sure that you configure an interface classification script for the simulated router. (See *Chapter 4, Classifying Interfaces and Subscribers*.) You also need to configure the SAE to instantiate a simulated router driver for each simulated router that you create.

You can configure a session store for simulated router drivers. The driver uses the session store to store subscriber sessions, service sessions, and policies. For a description of session store fields, see *Configuring Session Stores* on page 72.

The SDX software has a default simulated router driver instance called `default@simJunos`. To create additional simulated router driver instances:

1. In the Simulated Router Driver area next to the Create a New Instance of button, select Driver, and click Create a New Instance of.

The Create New Instance dialog box appears.

2. Assign a name to the instance, and click OK.

The new instance appears in the Simulated Router Driver area.

Driver Type

- Type of device that the simulated driver simulates.
- Value—JUNOS, JUNOSe, or PCMM
- Default—JUNOS
- Property name—Router.sim. < simulated router name > .type

Router Version

- Version of the router to simulate.
- Value—Software version that is sent by the router; for example, 6.4
- Default—No value
- Property name—Router.sim. < simulated router name > .version

Router Address

- Address of the router that is available for router initialization scripts.
- Value—IP address
- Default—10.0.0.1
- Property name—Router.sim. < simulated router name > .routerIp

Transport Router

- Name of a virtual router that is used to connect to the SAE. This value is passed to the router initialization script. It is not supported on the JUNOS routing platform.
- Value—Name of a virtual router
- Default—No value
- Property name—Router.sim. < simulated router name > .transportRouter

Configuring Session Stores

For overview information about session stores, see *Storing Subscriber and Service Session Data* on page 42.

For the session store feature, you can configure:

- Session store parameters within a router or PCMM device driver configuration. See *Configuring Session Store Parameters for a Device Driver* on page 73.
- Global session store parameters for the session store infrastructure that is shared by all session store instances (active or passive) on the SAE. See *Configuring Global Session Store Parameters* on page 77.

Configuring Session Store Parameters for a Device Driver

Use the fields in this section to configure session store parameters for a router driver or a PCMM device driver.

Session Store	
Maximum Queue Age [ms]	5000
Maximum Queued Operations	50
Maximum Queue Size [bytes]	51050
Maximum File Size [bytes]	25000000
Minimum Disk Space Usage	25
Rotation Batch Size	50
Maximum Session Data Bytes	10000
Disk Load Buffer Size [bytes]	1000000
Network Buffer Size [bytes]	51050
Retry Interval [ms]	300000
Communications Timeout [ms]	60000
Load Timeout [ms]	420000
Session Store Idle Timeout [ms]	3600000
Maximum Backlog Ratio	1.5
Minimum Backlog	5000000

Maximum Queue Age [ms]

- Maximum age that a queue of buffered store operations (such as adding a session to the store or removing a session from the store) can reach before the queue is written to a session store file.
- Value—Number of milliseconds in the range 0–2147483647. A value of –1 indicates that there is no limit. A value of zero causes the session store to write each store operation to a session store file immediately.
- Default—5000
- Property name—Router. < deviceType > .sessionStore.maxQueueAge

Maximum Queued Operations

- Number of buffered store operations that are queued before the queue is written to a session store file.
- Value—Integer in the range 0–2147483647. A value of –1 indicates that there is no limit. A value of zero causes the session store to write each store operation to a session store file immediately.
- Default—50
- Property name—Router. < deviceType > .sessionStore.maxQueueOps

Maximum Queue Size [bytes]

- Maximum size that a queue of buffered store operations can reach before the queue is written to a session store file.
- Value—Number of bytes in the range 0–2147483647
- Default—51050
- Property name—Router. < deviceType > .sessionStore.maxQueueBytes

Maximum File Size [bytes]

- Maximum size of session store files. When a file reaches this size, a new file is created.
- Value—Number of bytes in the range 0–2147483647
- Default—25000000
- Property name—Router. < deviceType > .sessionStore.maxFileBytes

Minimum Disk Space Usage

- Percentage of space in all session store files that is used by live sessions. When the percentage of space in the session store files that is used by live sessions decreases to this percentage, the oldest session store file is compacted and appended to the newest session store file, and then the oldest session store file is deleted.
- Value—Percentage of disk space in the range 1–100
- Guidelines—We recommend a range of 30–50.
- Default—40
- Property name—Router. < deviceType > .sessionStore.minDiskSpaceUsage

Rotation Batch Size

- When the oldest session store file is rotated, specifies the number of sessions that are rotated from the oldest file to the newest file at the same time. While a set of sessions is rotated, no other session store activity can take place.
- Value—Integer in the range 0–2147483647
- Default—50
- Property name—Router. < deviceType > .sessionStore.rotationBatchSize

Maximum Session Data [bytes]

- Maximum size of a single subscriber or service session. Use this parameter to reserve memory for an internal buffer.
- Value—Number of bytes in the range 0–2147483647
- Default—10000
- Property name—Router. < deviceType > .sessionStore.maxSessionDataBytes

Disk Load Buffer Size [bytes]

- Size of the buffer that is used to load all of a session store's files from disk at startup.
- Value—Number of bytes in the range 0–2147483647
- Default—1000000
- Property name—Router. < deviceType > .sessionStore.diskLoadBufferBytes

Network Buffer Size [bytes]

- Size of the buffer that holds messages or message segments that are waiting to be sent to passive session stores.
- Value—Number of bytes in the range $21 + \text{ < size of maximum session size field > } - 2147483647$
- Default—51050
- Property name—Router. < deviceType > .sessionStore.networkBufferBytes

Retry Interval [ms]

- Time interval between attempts by the active session store to connect to missing passive session stores.
- Value—Number of milliseconds in the range 0–2147483647
- Default—300000
- Property name—Router. < deviceType > .sessionStore.retryInterval

Communications Timeout [ms]

- Amount of time that a session store waits before closing when it is blocked from reading or writing a message. This timeout does not apply when a session store is waiting for a remote session store to load its state from disk.
- Value—Number of milliseconds in the range 0–2147483647
- Default—60000
- Property name—Router. < deviceType > .sessionStore.communicationsTimeout

Load Timeout [ms]

- Amount of time that an active session store waits for a passive session store or a passive session store waits for an active session store to load its data from disk before it closes the connection to the session store.
- Value—Number of milliseconds in the range 0–2147483647
- Default—420000
- Property name—Router. < deviceType > .sessionStore.remoteStoreLoadTimeout

Session Store Idle Timeout [ms]

- Amount of time that a passive session store waits for activity from the active session store before it closes the connection to the active session store. This timeout applies after the session store startup and initial update processes are complete.
- Value—Number of milliseconds in the range 0–2147483647
- Default—3600000
- Property name—Router. < deviceType > .sessionStore.idleTimeout

Maximum Backlog Ratio

- Along with the minimum backlog size, specifies when the active session store closes the connection to a passive session store because of a backlog of messages waiting to be sent. After the startup and initial update processes are complete, if the backlog becomes too large, the connection to the passive session store is closed. After the retry interval ends, a new connection is opened.

If the backlog of unsent operations (in bytes) divided by the total size (in bytes) of all live store operations is greater than this number, the connection is closed.

- Value—Floating point number
- Default—1.5
- Property name—Router. < deviceType > .sessionStore.backlogDeathMaxRatio

Minimum Backlog Size [bytes]

- Along with the maximum backlog ratio, specifies when the active session store closes the connection to a passive session store because of a backlog of messages waiting to be sent to the passive session store. After the startup and initial update processes are complete, if the backlog becomes too large, the connection to the passive session store is closed. After the retry interval ends, a new connection is opened.

If the maximum backlog ratio is met, the active session store does not close the connection unless the backlog of messages (in bytes) is greater than this number.

- Value—Number of bytes in the range 0–2147483647
- Default—5000000
- Property name—Router. < deviceType > .sessionStore.backlogDeathMinBehind

Configuring Global Session Store Parameters

Use the fields in this section to configure parameters for the session store infrastructure that is shared by all session store instances (active or passive) on the SAE.

Session Store	
Session Store IP Address	<input type="text"/> <input type="button" value="Disable"/>
Session Store Port	<input type="text"/> <input type="button" value="Disable"/>
Root Directory	<input type="text"/> <input type="button" value="Disable"/>

Session Store IP Address

- IP address or hostname that the session store infrastructure on this SAE uses to listen for incoming TCP connections from active session stores.
- Value—IP address. The address must be an IP address configured for the SAE host. If you do not enter an address or if you disable this field, active session stores cannot create passive session stores on this SAE.
- Guidelines—We recommend that you enter an address that is configured in a list of connected SAEs. See *Configuring Connected SAEs* in *SDX Objects Guide, Chapter 5, Managing Routers and Virtual Routers*.
- Default—No value
- Property name—Router.sessionStore.ServerIp

Session Store Port

- TCP port number on which the session store infrastructure on this SAE listens for incoming connections from active session stores. Note that this field has no effect if you have not configured a session store IP address.
- Value—Port number
- Default—8820
- Property name—Router.sessionStore.ServerPort

Root Directory

- Root directory in which the session store creates files. Note that this field has no effect if you have not configured a session store IP address.
- Value—Directory name
- Default—var/sessionStore
- Property name—Router.sessionStore.rootDir

Configuring SNMP Communities

The SAE can use the Simple Network Management Protocol (SNMP) in its router scripts; for example, to read the IP address pools from the JUNOSe router. You can configure default SNMP community strings that are used for read and write access to the router. These values are used as a global default. We recommend that you configure the value per virtual router. You can do so in the virtual router configuration in SDX Admin (o = umc > Network > Router > virtualRouter).

SNMP configuration interface showing two fields:

- Read-Only Community String: ***** [Show]
- Read-Write Community String: ***** [Show]

Read-Only Community String

- Default SNMP community string used for read access to the router.
- Value—SNMP community string that matches a read-only community string configured on the router
- Default—Public
- Property name—Router.read-only.community.string

Read-Write Community String

- Default SNMP community string used for write access to the router.
- Value—SNMP community string that matches a read-write community string configured on the router
- Default—Private
- Property name—Router.read-write.community.string

Identifying a Profile for Unauthenticated Subscribers

The SAE uses an unauthenticated subscriber profile as a transitional profile for subscribers who are not logged in to the SAE. For example, if a subscriber logs out of the SAE using the API method `Subscriber.logout()`, an unauthenticated subscriber session is created. The unauthenticated subscriber profile must exist and can be subscribed to services available for unauthenticated subscribers. The portal implementation determines whether unauthenticated (anonymous) subscribers can access the portal.

Unauthenticated User DN: uniqueID=unauthenticated,ou=local,retailerName=default,o=Users,<base>

Unauthenticated User DN

- Identifies a subscriber profile for unauthenticated access to the portal.
- Value— < DN >
- Default—
uniqueID = unauthenticated, ou = local, retailerName = default, o = Users, < base >
- Property name—Router.unauthUserDn

Configuring the Virtual Portal Address

The virtual portal IP address allows different subscribers who are connected to different routers and are managed by different SAEs to use the same URL, IP address, or both to reach the portal pages. In a redundant installation, this address is shared by the redundant servers. See *Virtual Portal Address Feature* on page 43.

Virtual Portal Address	<input type="text" value="192.168.254.1"/>	<input type="button" value="Disable"/>
------------------------	--	--

Virtual Portal Address

- IP address of the portal server that is published to subscribers and used in router policies.
- Value—IP address
- Default—192.168.254.1
- Property name—Router.virtual.portal.address

Configuring Router Initialization Scripts

When the SAE establishes a connection with a router, it can run a router initialization script to customize the setup of the connection. Router initialization scripts are run when the connection between a router and the SAE is established and again when the connection is dropped. For more information, see *Router Initialization Scripts* on page 5.

▼ Router Scripts	
Extension Path	<input type="text"/>
General Script	<input type="text"/>
JUNOS Script	<input type="text"/>
JUNOSe Script	<input type="text"/>
JUNOSe Script (XDR)	<input type="text"/>

Extension Path

- Path to router initialization scripts that are not in the default location, */opt/UMC/sae/lib*.
- Value—List of paths separated by semicolons (;)
- Default—No value
- Property name—*Extension.path*

General Script

- Router initialization script that can be used for all types of routers that the SDX software supports. The script is run when the connection between a router and the SAE is established and again when the connection is dropped.
- Value—Name of a script
- Default—No value
- Property name—*Router.script.**

JUNOS Script

- Router initialization script for JUNOS routing platforms. The script is run when the connection between a router and the SAE is established and again when the connection is dropped.
- Value—Name of a script
- Default—*iorPublisher*
- Property name—*Router.script.junos*

JUNOSe Script

- Router initialization script for JUNOSe routers when the JUNOSe driver uses COPS-PR mode when connecting to the SAE. The script is run when the connection between a router and the SAE is established and again when the connection is dropped.
- Value—Name of a script
- Default—No value
- Property name—*Router.script.junose*

JUNOSe Script (XDR)

- Router initialization script for JUNOSe routers when the JUNOSe driver uses XDR mode when connecting to the SAE. The script is run when the connection between a router and the SAE is established and again when the connection is dropped.

In COPS XDR mode, the router does not send the network access server (NAS) IP address to the SAE. If your configuration requires this value, add the following line to a JUNOSe script:

```
import ERXnasip
```

When you add the **import ERXnasip** entry, the script obtains the NAS-IP address from the router through SNMP. This mechanism can affect performance, especially when the SAE manages a large number of virtual routers.

- Value—Name of a script
- Default—Unspecified
- Examples—iorPublisher, poolPublisher
- Property name—Router.script.junose_xdr

Configuring the License Manager

The license manager for an SAE maintains the licenses for the SAE and communicates with the license server to manage licenses needed by the SAE. The SAE license manager properties specify SAE client properties and access to the directory in which SDX license data is stored. The SAE license manager reads the server license to identify the license server to which it connects.

For information about the license server, see *Chapter 7, Using the License Server*.

To use SDX Configuration Editor to configure SAE properties for the license manager, select a directory configuration object for the SAE, and then select the License Manager tab.

Configuring Directory Access

The directory access configuration defines the connection from the SAE to the directory in which SDX license data is stored and directory eventing parameters for the data.

Directory Access	
Server Address	127.0.0.1 Disable
Server Port	389
Search Base	ou=Licenses,o=Management,<base>
Authentication DN	cn=license-operator,o=Operators,<base>
Password	***** Show
Secured LDAP protocol	LDAPS Disable
DES Connection Manager ID	LICENSE_MANAGER
DES Event Base DN	<base> Disable
DES Signature DN	<base> Disable
DES System Management	No Disable

Server Address

- Disables or enables and identifies the directory server that stores licensing data.
- Value—IP address or hostname; use a space to separate addresses for multiple directory servers: 127.153.27.1 192.168.0.1
- Default—Disabled
- Property name—LicenseMgr.repository.ldap.server.address

Server Port

- Port number of the LDAP connection to the directory server that stores licensing data.
- Value—Integer in the range 1–65535
- Default—389
- Property name—LicenseMgr.repository.ldap.server.port

Search Base

- Subtree in the directory where licensing information is stored. The SAE searches for the license key below this path.
- Value— < DN >
You can use the special value < base > to refer to the globally configured base DN.
- Default—*ou = Licenses, o = Management, < base >*
- Property name—LicenseMgr.repository.ldap.server.base.dir

Authentication DN

- DN used by the SAE to authenticate access to the directory server.
- Value— < DN >
You can use the special value < base > to refer to the globally configured base DN.
- Default—*cn = license-operator, o = Operators, < base >*
- Property name—LicenseMgr.repository.ldap.server.authDN

Password

- Password used to authenticate access to the directory.
- Value—Text string or Base64 string
- Default—License
- Property name—LicenseMgr.repository.ldap.server.password

Secured LDAP protocol

- Enables or disables LDAPS as the secure protocol for connections to the directory server that stores license data.
- Value—Enable or Disable
- Default—Disable
- Property name—LicenseMgr.repository.ldap.server.security.protocol

DES Connection Manager ID

- DES connection manager within the Java Naming and Directory Interface (JNDI) framework.
- Value—Text string
- Default—LICENSE_MANAGER
- Property name—LicenseMgr.repository.ldap.server.des.connection_manager_id

DES Event Base DN

- Disables or enables and sets the base DN for the license manager data.
- Value— < DN >
You can use the special value < base > to refer to the globally configured base DN.
- Default— < base >
- Property name—LicenseMgr.repository.ldap.server.des.event_baseDN

DES Signature DN

- Disables or enables and sets the DN of the entry that specifies the LDAP schema attribute usedDirectory. This attribute identifies the type of directory, such as openLDAP or DirX, on which the license data is stored. See *Configuring the Type of Directory* on page 271.
- Value— < DN >
You can use the special value < base > to refer to the globally configured base DN.
- Default—Disabled
- Property name—LicenseMgr.repository.ldap.server.des.signatureDN

DES System Management

- Specifies whether the SDX SNMP agent exports MIBs for this directory connection.
- Value—Yes or No
- Default—No
- Property name—LicenseMgr.repository.ldap.server.des.enable_sysman

Configuring Client Properties

The Client configuration sets the SAE client properties.

The screenshot shows a configuration window titled "Client". It contains two input fields: "Client Type" with the value "SDX" and "Client Cache" with the value "var/run/lic_cache".

Client Type

- Type of the license client.
- Value—SDX is currently the only valid value.
- Default—SDX
- Property name—LicenseMgr.license.client.type

Client Cache

- Path to a cache file.
- Value—Valid path
- Default—*var/run/lic_cache*
- Property name—LicenseMgr.license.client.cache

Configuring Interim Accounting

You can enable and disable interim accounting and set intervals between interim accounting messages for services and subscribers. These settings apply to all subscriber sessions and service sessions. You can override these settings for specific services by configuring an accounting interim interval in the value-added service configuration.

To configure interim accounting, fill in the fields in the Miscellaneous tab in SDX Configuration Editor.

The screenshot shows a configuration window titled "Interim Accounting". It contains four input fields: "Service Interim Accounting" set to "Yes", "Service Interim Interval [s]" set to "900", "User Interim Accounting" set to "Yes", and "User Interim Interval [s]" set to "900".

Service Interim Accounting

- Enables or disables service interim accounting. If enabled, the SAE continually generates Interim-Update accounting requests for all active services at the interval specified in the Service Interim Interval field.
- Value—Yes or No
- Default—Yes
- Property name—AccountingMgr.interim.accounting.running

Service Interim Interval [s]

- Interval between service interim accounting messages. A short interval causes the SAE to send many messages to the router and to the RADIUS servers. A long interval can result in a large loss of accounting information in the event of a system failure.
- Value—Number of seconds in the range 900–86400
- Default—900
- Property name—AccountingMgr.interim.accounting.polling.interval

User Interim Accounting

- Enables or disables interim accounting for subscribers. If enabled, the SAE continually generates Interim-Update accounting requests for all active subscribers at the interval specified in the User Interim Interval field.
- Value—Yes or No
- Default—Yes
- Property name—AccountingMgr.user.interim.accounting.running

User Interim Interval [s]

- Interval between subscriber interim accounting messages. A short interval causes the SAE to send many messages to any configured accounting servers. A long interval can result in a large loss of accounting information in the event of a system failure.
- Value—Number of seconds in the range 900–86400
- Default—900
- Property name—AccountingMgr.user.interim.accounting.polling.interval

Allowing Multiple Logins from the Same IP Address

You can specify whether or not the SAE allows multiple logins from the same IP address. To do so, fill in the Login Behavior: Allow Same IP Login field in the Miscellaneous tab in SDX Configuration Editor.

Subscriber Sessions

Assigned IP Sessions Idle Timeout [s]

Login Behavior: Allow Same Ip Login

Login Behavior: Allow Same IP Login

- Specifies whether the SAE allows a login from the same IP address without requiring that the previous session logs out first.
- Value
 - Yes—SAE logs in the new subscriber session and automatically logs out the previous session.
 - No —SAE denies login requests if a subscriber session for an IP address is active.
- Property name—UserManager.sameIpLogin

Authenticating Registered Username/Password Pairs

You can specify whether or not registered username/password pairs are authenticated. To do so, fill in the Registration authentication field in the Miscellaneous tab in SDX Configuration Editor.

Login Registration

Registration authentication

Registration authentication

- Specifies whether the application programming interface (API) method registerLoginCredentials authenticates the registered username/password or creates the registration without authentication.
- Value—Yes or No
- Guidelines—Set to Yes if your authentication server does not allow authentication while a session for the authenticated username is active.
- Property name—RegisterLoginCredentials.authenticateRegistration

Configuring Timers for Session Reactivation

If a service session fails unexpectedly, the SAE tries to start the session again in the background. You can change how many times the SAE tries to activate the session and the interval between these attempts. In most instances, the default values do not need to be changed.

Background Service Activation

Activation time [s]

Limit of retries

Activation time [s]

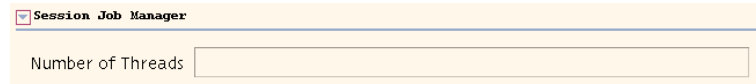
- Time between attempts to activate a service session if activation fails or to deactivate a service session if deactivation fails. This process takes place in the background.
- Value—Number of seconds in the range -1–9223372036854775807
-1 indicates no limit
- Default—60
- Property name—Service.background.retry_time

Limit of retries

- Number of times the SAE tries to activate a service session if activation fails or to deactivate a service session if deactivation fails. This process takes place in the background.
- Value—Integer in the range -1–2147483647
-1 indicates no limit
- Default— -1
- Property name—Service.background.retry_limit

Configuring the Number of Threads for Sessions

You can configure the number of threads used to handle session-related activity.



The screenshot shows a configuration panel titled "Session Job Manager". Below the title is a text input field labeled "Number of Threads".

Number of Threads

- Sets the number of threads used for session-related activity; for example, interim accounting, subscriber and service session timeout, idle timeouts, aggregate service keepalives, and remote session monitoring.
- Value—Integer in the range 1–50
- Default—10
- Property name—SessionJobManager.numThreads

Reducing the Size of Session Objects

You can use serialized data compression to reduce the size of sessions objects that the SAE sends across the network for the session store feature. Enabling this property reduces the size of objects, but increases the CPU load on the SAE.



The screenshot shows a configuration panel titled "Compress Serialized Data". Below the title is a text input field.

Compress Serialized Data

- Enables or disables reducing the size of session objects (subscriber and service sessions) that the SAE sends across the network for the session store feature.
- Value—Yes or No
- Guidelines—We recommend that you do not enable this option because of the increase to the CPU load.
- Default—No
- Property name—Main.compressSerializedData

Modifying the SAE Property File

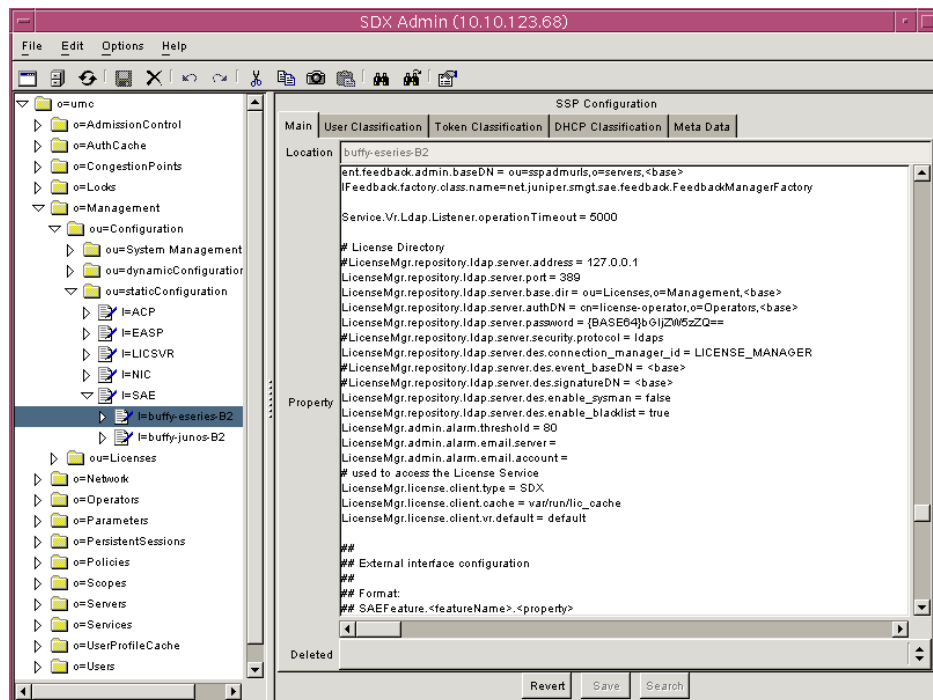
This section shows how to edit the property file with SDX Admin or a standard text editor. Use the property names that are included in field descriptions for properties in SDX Configuration Editor.

Editing Properties with SDX Admin

To edit the properties with SDX Admin:

1. Start SDX Admin.
2. In the SDX Admin navigation pane, access the object *I = SAE*, *ou = staticConfiguration*, *ou = configuration*, *o = management*, *o = umc*.
3. In this folder, click on the *I = POP-ID* object associated with this SAE.

The SAE configuration appears in the Main tab in the SSP Configuration pane.



4. Scroll to the text you want to edit, or click Search to find an item in the configuration file.
5. Add or modify the relevant information, and click Save.

Editing Properties with a Text Editor

To edit the properties with a text editor:

1. Open a shell in the directory in which you installed the SAE.

The default installation directory for SAE is */opt/UMC/sae*.

2. Download the properties to a file with the SAE configuration utility.

etc/config -g <filename>

A file called < filename > , which contains the SAE properties, appears in the *sae* subdirectory.

3. Edit the file with a text editor, such as VI or EMACS.
4. Update the object's file properties with the SAE configuration utility.

etc/config -p <filename>

Other Configuration Tasks

There are two configuration tasks that you can perform only by modifying the property file:

- Loading subscriptions based on RADIUS authorization
- Accepting Login Names with Different Formats

Loading Subscriptions Based on RADIUS Authorization

You can set up the SAE to load subscriptions based on values that it receives in RADIUS authorization response packets. For this method of loading subscriptions to work, the subscriber must be subscribed to the service.

To use this feature, you set up the RADIUS authorization plug-in to return the `setLoadServices` attribute, and you specify a regular expression in the SAE property file. When the plug-in returns the `setLoadServices` attribute, the SAE applies the regular expression to the string in the `setLoadServices` attribute.

There are two SAE properties that you can use to set the expression:

- `SubscriptionParser.regex`—Specifies the regular expression that is used to match a single service name.
- `SubscriptionParser.auto`—Specifies the number of a group of services that corresponds to activate-on-login services. That is, if a subscription is matched by this group, it is activated.

For example:

```
SubscriptionParser.regex = ([^;!]*);|([^;!]*)!
SubscriptionParser.auto = 2
```

A group match corresponds to a regular expression that is enclosed in (). In this example, the regular expression in the subscription parser contains two groups:

1. A string of characters other than “;” and “!”, followed by “;”
2. A string of characters other than “;” and “!”, followed by “!”

The value of 2 in the SubscriptionParser.auto property causes the second group of services—services followed by !—to be activated on login. For example, if the setLoadServices string is video-gold;audio-gold!, it is parsed to video-gold and audio-gold. The audio-gold subscription is activated provided that the subscriber is subscribed to audio-gold services.



NOTE: Persistent service sessions are not parsed. That is, if a subscriber has activated persistent service sessions, then these sessions are activated independent of the RADIUS authorization responses.

Another way to load subscriber services based on RADIUS authorization is to use the serviceBundle vendor-specific attribute (VSA) as input to the subscriber classification script and load different subscriber profiles based on the RADIUS response. Different subscriber profiles subscribe to different services. This approach gives wholesalers a basic tool to outsource service subscriptions to a retailer. The wholesaler and retailer must agree on a RADIUS attribute (for example, serviceBundle) that is provided by the retailer and interpreted by the SAE (that is, the wholesaler).

The subscription parser properties are located in */opt/UMC/sae/etc/dir.template*. (See *Modifying the SAE Property File* on page 89.)

SubscriptionParser.regex

- Regular expression that is applied to the setLoadServices attribute in RADIUS authorization response packets. The regular expression matches a single service name and is applied repeatedly until no match is found.
- Value—Regular expression; you can group matches by enclosing them in parenthesis ().
- Default—”([^;!]*);|([^;!]*)!”
- Property name—SubscriptionParser.regex

SubscriptionParser.auto

- Expression that identifies the number of a group of services that are to be activated automatically. If a subscription is matched by this group, it is automatically activated.
- Value—Expression
- Default—2
- Example—The default regular expression corresponds to a string of service names that are separated by “,” or “!”. If a service name is followed by “!”, it is activated automatically.
- Property name—SubscriptionParser.auto

Accepting Login Names with Different Formats

You can configure the SAE to accept login names of different formats. For example, the format `subscriberName@domainName` is a common format for the login name of subscribers who connect through PPP; however, other subscribers may use other formats, such as `domainName/userName`.

To configure the SAE to accept these different formats, you specify a set of properties that parse the login name to obtain the `userName` and `domainName` objects for the subscriber. Each property contains a regular expression that includes one or two subexpressions—*independent expressions in the complete regular expression—each of which is enclosed in parentheses.*

The property for login name parsing has the form:

```
LoginName.parser.<number>.<userGroup>[.<domainGroup>] = \
<regular expression>
```

number

- Number that specifies the order in which the SAE should apply the property when it parses the `loginName`. The SAE applies the properties in the specified order from lowest to highest.

userGroup

- Number of the backreference that extracts the username.
- In the following example, the `userGroup` backreference is set to 1. This means that the first backreference in the expression `([^\@]*)` identifies the username:
`LoginName.parser.1.1.2 = ([^\@]*)@(.*)`

domainGroup

- Optional number of the backreference that extracts the domain name.
- In the following example, the `domainGroup` backreference is set to 1. Therefore, the first backreference in the expression `([^\/*]*)` identifies the domain name:
`LoginName.parser.2.2.1 = ([^\/*]*)/(.*)`

regular expression

- Regular expression that includes one or two subexpressions—*independent expressions in the complete regular expression—each of which is enclosed in parentheses.*
- When you define regular expressions for a domain name parser, you must include four backslashes (`\\`) to effect a single backslash. For example, suppose you define the following parser:
`LoginName.parser.1.2.1 = (.*)[\\](.*)`

This example parses the login name `isp1\jane` as:

domain name: `isp1`
username: `jane`

- For more information about using regular expressions for this feature, see: <http://jakarta.apache.org/regexp/apidocs/org/apache/regexp/RE.html>

Default Login Parser Properties

Table 6 shows default properties that the SAE uses to parse login names. Table 7 shows some examples of subscriber and domain names obtained through the default parsing properties.

Table 6: Default SAE Properties That Parse Login Names

Property	Function	Values
LoginName.parser.1.1.2 = <code>([^\@]*)@(.*)</code>	Parses login names of the format <code>userName@domainName</code>	LoginName.parser.1.1.2—First parser applied by the SAE to login names; first backreference identifies the username, and second backreference identifies the domain name. <code>([^\@]*)</code> —First backreference: username is a string of characters other than the at-sign (<code>@</code>). <code>@</code> —An at-sign precedes the domain name. <code>(.*)</code> —Second backreference: domain name is a string of characters.
LoginName.parser.2.2.1 = <code>([^\/]*)/(.*)</code>	Parses login names of the format <code>domainName/userName</code>	LoginName.parser.2.2.1—Second parser applied by the SAE to login names; second backreference identifies the subscriber name, and first backreference identifies the domain name. <code>([^\/]*)</code> —First backreference: domain name is a string of characters other than the forward slash (<code>/</code>). <code>/</code> —A forward slash precedes the username. <code>(.*)</code> —Second backreference: username is a string of characters.
LoginName.parser.3.1 = <code>(.*)</code>	Parses login names that contain no domain name	LoginName.parser.3.1—Third parser applied by the SAE to login names; first backreference identifies the username, no domain name. <code>(.*)</code> —First backreference: username is a string of characters.

Table 7: Examples of Subscriber and Domain Names Obtained from Default Properties

Login Name	Output from Default Parsing Properties
<code>joeUser@isp1.com</code>	<ul style="list-style-type: none"> ■ The username is <code>joeUser</code>. ■ The domain name is <code>isp1.com</code>.
<code>isp1/joe</code>	<ul style="list-style-type: none"> ■ The username is <code>joe</code>. ■ The domain name is <code>isp1</code>.
<code>isp1/joe@isp2</code>	<ul style="list-style-type: none"> ■ The username is <code>isp1/joe</code>. ■ The domain name is <code>isp2</code>.

