

## Chapter 5

# Configuring Subscriber-Related Properties on the SAE on a Solaris Platform

This chapter describes how to configure subscriber-related properties on the SAE with SDX Configuration Editor or by modifying a property file. You can use SDX Configuration Editor and property files on a Solaris platform.

You can also use the SRC CLI to configure an SAE on the C-series platform or on a Solaris platform. See *Chapter 4, Configuring Subscriber-Related Properties on the SAE with the SRC CLI*.

Topics in this chapter include:

- Overview on page 50
- Configuring the Length of Time MAC Addresses Remain in SAE Cache on page 50
- Identifying a Profile for Unauthenticated Subscribers on page 51
- Configuring Interim Accounting for Services and Subscribers on page 52
- Avoiding Overcharges for Sessions That Time Out on page 53
- Allowing Multiple Logins from the Same IP Address on page 54
- Authenticating Registered Username/Password Pairs on page 55
- Configuring Timers for Session Reactivation on page 56
- Modifying the SAE Property File on page 57
- Loading Subscriptions Based on RADIUS Authorization on page 58
- Accepting Login Names with Different Formats on page 60

## Overview

The SAE property file contains SAE configuration data that is stored in the directory.

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.

## 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:

1. In the navigation pane, select a directory configuration object for the SAE that you want to configure.
2. Select the **Router** tab.

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

3. Fill in the field as described in *Max Cache Expiration Time Field* on page 50.

### Max Cache Expiration Time Field

Use this field to configure the length of time that a subscriber profile remains in the SAE's in-memory cache.

#### MAC Cache Expiration Time

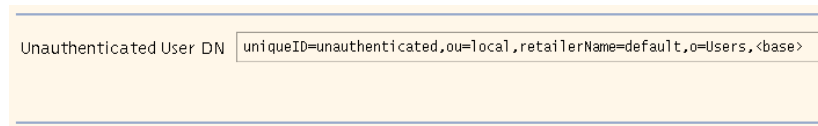
- Amount of time that a subscriber profile remains in the SAE's in-memory cache.
- Value—Number of seconds in the range 0–2147483647

- 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 JUNOS router and the instructions given to the subscriber on the Web portal, such as reboot your PC now.
- Default—1800
- Property name—maxMacCacheEntryAge

## 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. To specify an unauthenticated subscriber profile with SDX Configuration Editor:

1. In the navigation pane, select a directory configuration object for the SAE that you want to configure.
2. Select the **Router** tab.



Unauthenticated User DN

3. Fill in the field as described in *Unauthenticated User DN Field* on page 51.

### Unauthenticated User DN Field

Use the field in this section to specify the unauthenticated user DN.

#### 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 Interim Accounting for Services and Subscribers

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 with SDX Configuration Editor:

1. In the navigation pane, select a directory configuration object for the SAE that you want to configure.
2. Select the **Miscellaneous** tab, and expand the **Interim Accounting** section.

| Interim Accounting           |     |
|------------------------------|-----|
| Service Interim Accounting   | Yes |
| Service Interim Interval [s] | 900 |
| User Interim Accounting      | Yes |
| User Interim Interval [s]    | 900 |

3. Fill in the fields as described in *Interim Accounting Fields* on page 52.

### Interim Accounting Fields

Use the fields in this section to configure interim accounting.

#### 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

**Avoiding Overcharges for Sessions That Time Out**

When an idle timeout terminates a session, you can set up the SAE to reduce the session time reported in the accounting stop message by the idle time. This way the session time is accurately reported to avoid overcharges for the session. To adjust the session time with SDX Configuration Editor:

1. In the navigation pane, select a directory configuration object for the SAE that you want to configure.
2. Select the **Miscellaneous** tab, and expand the **Idle Timeout** section.



The screenshot shows a configuration window with a tab labeled 'Idle Timeout'. Below the tab, there is a field labeled 'Adjust Session Time' with a dropdown menu currently showing 'Yes'. To the right of the dropdown is a small icon of a document with a checkmark.

3. Fill in the field as described in *Idle Timeout Field* on page 54.

### Idle Timeout Field

Use the field in this section to specify whether or not the session time reported in accounting stop messages are adjusted.

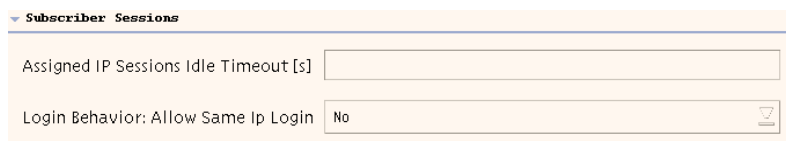
### Adjust Session Time

- Specifies whether, when an idle timeout terminates a session, the session time reported in the accounting stop message is reduced by the idle time. This way the session time is accurately reported to avoid overcharges for the session.
- Value
  - True—Reduces the session time by the amount of time specified by the idle timeout
  - False—Does not reduce the session time by the amount of time specified by the idle timeout
- Default—True
- Property name—AccountingMgr.adjustSessionTime

## 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 with SDX Configuration Editor:

1. In the navigation pane, select a directory configuration object for the SAE that you want to configure.
2. Select the **Miscellaneous** tab, and expand the **Subscriber Sessions** section.



▼ **Subscriber Sessions**

Assigned IP Sessions Idle Timeout [s]

Login Behavior: Allow Same Ip Login

3. Fill in the Login Behavior: Allow Same IP Login field as described in *Allow Same IP Login Field* on page 54.

### Allow Same IP Login Field

Use the field in this section to specify whether or not the SAE allows multiple logins from the same IP address.

**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 with SDX Configuration Editor:

1. In the navigation pane, select a directory configuration object for the SAE that you want to configure.
2. Select the **Miscellaneous** tab, and expand the **Login Registration** section.



The screenshot shows a configuration window with a yellow header bar. Below the header, the 'Login Registration' section is expanded, showing a table with one row. The table has two columns: 'Registration authentication' and a value field. The value field contains the text 'Yes' and has a small icon to its right.

| Login Registration          |     |
|-----------------------------|-----|
| Registration authentication | Yes |

3. Fill in the field as described in *Login Registration Field* on page 55.

**Login Registration Field**

Use the field in this section to specify whether or not registered username/password pairs are authenticated.

**Registration authentication**

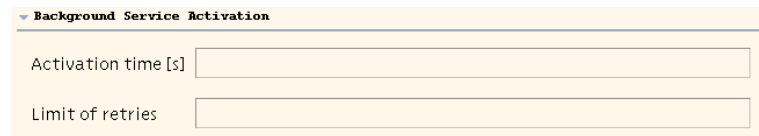
- Specifies whether the application programming interface (API) method `registerLoginCredentials` authenticates the registered username/password or creates the registration without authentication.
- Value—Yes or No
- Guidelines—Set to Yes if your authentication server does not allow authentication while 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.

To configure session reactivation behavior with SDX Configuration Editor:

1. In the navigation pane, select a directory configuration object for the SAE that you want to configure.
2. Select the **Miscellaneous** tab, and expand the **Background Service Activation** section.



The screenshot shows a configuration window with a yellow header bar labeled 'Background Service Activation'. Below the header, there are two input fields. The first field is labeled 'Activation time [s]' and the second field is labeled 'Limit of retries'. Both fields are currently empty.

3. Fill in the fields as described in *Background Service Activation Fields* on page 56.

### Background Service Activation Fields

Use the fields in this section to configure service reactivation behavior for the SAE.

#### 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



## Modifying the SAE Property File

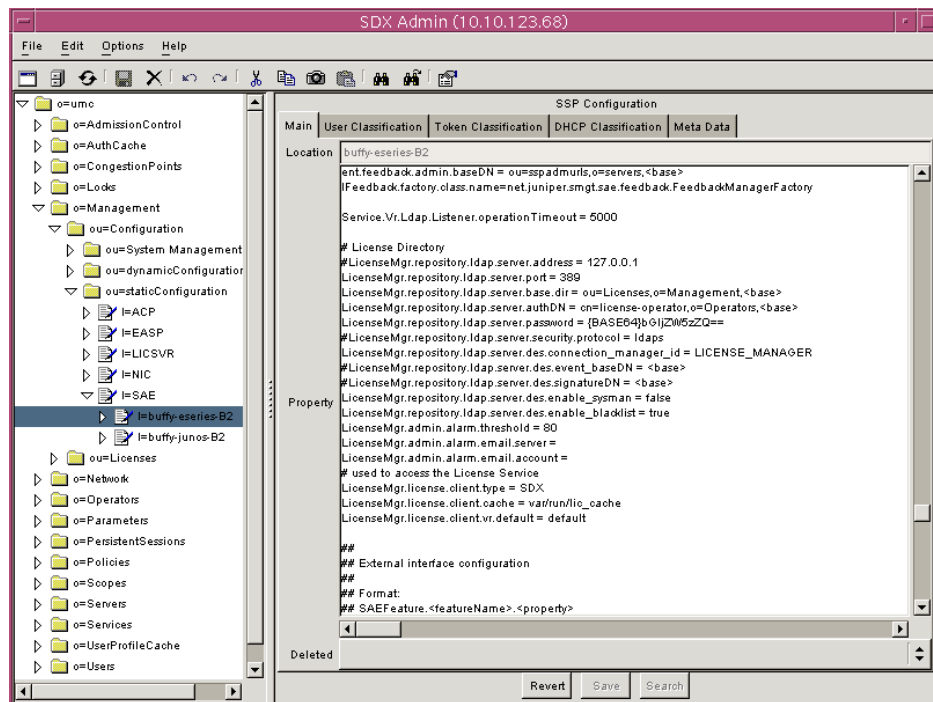
This section shows how to edit the property file with SDX Admin or a standard text editor. Use the property names that are included in field descriptions for properties in SDX Configuration Editor.

### Editing Properties with SDX Admin

To edit the properties with SDX Admin:

1. In the SDX Admin navigation pane, access the object *I = SAE*, *ou = staticConfiguration*, *ou = configuration*, *o = management*, *o = umc*.
2. 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.



3. Scroll to the text you want to edit, or click **Search** to find an item in the configuration file.
4. 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>**

## 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 57.)

#### **`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 10 shows default properties that the SAE uses to parse login names. Table 11 shows some examples of subscriber and domain names obtained through the default parsing properties.

**Table 10: Default SAE Properties That Parse Login Names**

| Property                                 | Function  | Values  |
|--|---|---|
| LoginName.parser.1.1.2<br>= ([^@]*)@(.*) | Parses login names of the format<br>userName@domainName | 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.<br><br>([^\@]*)—First backreference: username is a string of characters other than the at-sign (@).<br><br>@—An at-sign precedes the domain name.<br><br>(.*)—Second backreference: domain name is a string of characters.                 |
| LoginName.parser.2.2.1<br>= ([^/]*)/(.*) | Parses login names of the format<br>domainName/username | 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.<br><br>([^\/]*)—First backreference: domain name is a string of characters other than the forward slash (/).<br><br>/—A forward slash precedes the username.<br><br>(.*)—Second backreference: username is a string of characters. |
| LoginName.parser.3.1 = (.*)              | 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.<br><br>(.*)—First backreference: username is a string of characters.  |

**Table 11: Examples of Subscriber and Domain Names Obtained from Default Properties**

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

