

Configuring Tracking Plug-Ins

You can perform the following tasks to configure tracking plug-ins:

- Configuring Flat File Accounting Plug-Ins on page 1
- Configuring Headers for Flat File Accounting Plug-Ins on page 2
- Configuring Basic RADIUS Accounting Plug-Ins on page 4
- Configuring Flexible RADIUS Accounting Plug-Ins on page 6
- Configuring Custom RADIUS Accounting-Plug-Ins on page 8

Configuring Flat File Accounting Plug-Ins

Flat file accounting plug-ins write information to a file in a comma-separated format. The SRC software has a default flat file accounting plug-in instance called fileAcct. The fileAcct instance logs all possible attributes for 24-hour periods in the file *var/acct/log*.

Another item that you can configure for flat files is the names of the headers that appear in the file.

Use the following configuration statements to create flat-file accounting plug-in instances:

```
shared sae configuration plug-ins name name file-accounting {
    filename filename ;
    template template ;
    interval interval ;
    fields [(status | nas-id | host | router-name | interface-name | interface-alias |
            interface-descr | port-id | user-ip-address | login-name | accounting-id | auth-user-id
            | if-radius-class | if-session-id | service-name | radius-class | event-time | session-id
            | terminate-cause | session-time | in-octets | out-octets | in-packets | out-packets
            | nas-ip | user-mac-address | service-session-name | service-session-tag | user-type
            | user-radius-class | user-session-id | primary-user-name | subscription-name |
            login-id | if-index | event-time-millisecond | nas-port | operational | user-inet-address
            | nas-inet-address | router-type | interface-speed)...];
}
```

To create flat-file accounting plug-ins:

1. From configuration mode, access the basic RADIUS accounting plug-in configuration. In this sample procedure, the plug-in called fileAcct is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration plug-ins name  
          fileAcct file-accounting
```

2. Configure the name and location of the file to which the SAE writes accounting information.

```
[edit shared sae group west-region configuration plug-ins name fileAcct  
file-accounting]  
user@host# set filename filename
```

3. Configure the name of the template that defines header names for attributes listed in accounting files.

```
[edit shared sae group west-region configuration plug-ins name fileAcct
file-accounting]
user@host# set template template
```

4. Configure the number of hours of information stored in each accounting file.

```
[edit shared sae group west-region configuration plug-ins name fileAcct
file-accounting]
user@host# set interval interval
```

5. Configure the fields that you want to record in the accounting file.

```
[edit shared sae group west-region configuration plug-ins name fileAcct
file-accounting]
user@host# set fields [(status | nas-id | host | router-name | interface-name |
interface-alias | interface-descr | port-id | user-ip-address | login-name |
accounting-id | auth-user-id | if-radius-class | if-session-id | service-name |
radius-class | event-time | session-id | terminate-cause | session-time | in-octets
| out-octets | in-packets | out-packets | nas-ip | user-mac-address |
service-session-name | service-session-tag | user-type | user-radius-class |
user-session-id | primary-user-name | subscription-name | login-id | if-index |
event-time-millisecond | nas-port | operational | user-inet-address |
nas-inet-address | router-type | interface-speed)...]
```

6. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name fileAcct
file-accounting]
user@host# show
filename var/acct/log;
template FileAccounting.std;
interval 24;
fields [ status nas-id host router-name interface-name interface-alias
interface-descr port-id user-inet-address login-name accounting-id
auth-user-id if-session-id service-name event-time session-id
terminate-cause session-time in-octets out-octets in-packets out-packets
nas-inet-address user-mac-address service-session-name service-session-tag
user-type user-session-id ];
```

Configuring Headers for Flat File Accounting Plug-Ins

When the SAE writes data to a flat file, it writes into the first line the headers that identify the attributes in the file. For example, in the following accounting file, the first line lists headers for all attribute fields in the file, and the following lines list the actual data in each field:

```
Accounting Status,NAS ID,SSP Host,Router Name,Interface Name,Interface
Alias,Interface Description,NAS port ID,User IP Address,User ID,User Accounting
ID,User Authentication ID,INTF Radius Class,INTF,SessionId, Service Name,Radius
Class,Timestamp,SessionId, Terminate Cause,Session Time,Input Octets,Output
```

```
Octets,Input Packets,Output Packets,NAS IP,User Mac address,Service Session
Name,Service Session Tag,User Session Type,User Session Radius Class,User
Session ID
start,SSP.uelmo,uelmo,default@erx7_ssp57,FastEthernet1/1.1,,IP1/1.1,default@erx7_ssp57
FastEthernet1/1:65535, 10.10.10.20,pebbles@virneo.net,,,erx fastEthernet
1/1:0001048619,Video-Gold,Video-Gold,Fri Jan 30 14:23:29 EDT 2004,
VideoGold:null:1064946209182, 0,0,0,0,0,0, 10.10.7.17,,,PPP,,
pebbles:1064946144841
```

You can assign your own names to the headers that appear in the file. To do so, define the header names in a template, and then set up file accounting plug-in instances to use the template. The default template, FileAccounting.std, defines header names for all possible attributes. You can use the default template or create your own templates.

Use the following configuration statements to create a file accounting template:

```
shared sae configuration file-accounting-template name ...
shared sae configuration file-accounting-template name attributes (status | nas-id |
host | router-name | interface-name | interface-alias | interface-descr | port-id |
user-ip-address | login-name | accounting-id | auth-user-id | if-radius-class | if-session-id
| service-name | radius-class | event-time | session-id | terminate-cause | session-time
| in-octets | out-octets | in-packets | out-packets | nas-ip | user-mac-address |
service-session-name | service-session-tag | user-type | user-radius-class |
user-session-id | primary-user-name | subscription-name | login-id | if-index |
event-time-millisecond | nas-port | operational | user-inet-address | nas-inet-address
| router-type | interface-speed | service-bundle | user-dn | uid | domain | retailer-dn |
password | service-scope | session-timeout | downstream-bandwidth |
upstream-bandwidth | dhcp-packet | aggr-session-id | aggr-login-name | aggr-user-dn
| aggr-user-inet-address | aggr-accounting-id | aggr-auth-user-id) {
value ;
}
```

To set up a file accounting template:

1. From configuration mode, access the file accounting template configuration. In this sample procedure, the template called std is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration
file-accounting-template std
```

2. Define header names.

```
[edit shared sae group west-region configuration file-accounting-template std]
user@host# set attributes attribute value
```

For example:

```
[edit shared sae group west-region configuration file-accounting-template std]
user@host# set attributes terminate-cause "RADIUS Termination Cause"
```

3. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration file-accounting-template
std]
user@host# show
attributes {
    terminate-cause "RADIUS Termination Cause";
    service-session-name "Service Session Name";
}
```

Configuring Basic RADIUS Accounting Plug-Ins

You can use basic RADIUS accounting plug-ins to send accounting information to an external RADIUS accounting server or to a group of redundant servers. To communicate with nonredundant servers, you need to create multiple instances of the plug-in.

Use the following configuration statements to configure RADIUS accounting plug-ins:

```
shared sae configuration plug-ins name name radius-accounting {
    load-balancing-mode (failover | roundRobin);
    failback-timer failback-timer ;
    nas-ip (Ssplp | Erxlp);
    retry-interval retry-interval ;
    maximum-queue-length maximum-queue-length ;
    bind-address bind-address ;
    udp-port udp-port ;
    username (login-name | accounting-id | auth-user-name | manager-id);
    calling-station-id (mac | no);
    default-peer default-peer ;
}
```

To set up basic RADIUS accounting plug-ins:

1. From configuration mode, access the basic RADIUS accounting plug-in configuration. In this sample procedure, the plug-in called basicRadius is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration plug-ins name
basicRadius radius-accounting
```

2. Configure the mode for load-balancing RADIUS servers.

```
[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set load-balancing-mode (failover | roundRobin)
```

3. Specify if and when the SAE attempts to fail back to the default peer.

```
[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set failback-timer failback-timer
```

4. (Optional) Configure the value of the NAS-IP attribute.

```
[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set nas-ip (Ssplp | Erxlp)
```

5. Configure the time the SAE waits for a response from a RADIUS server before it resends the RADIUS packet.

```
[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set retry-interval retry-interval
```

6. Configure the maximum number of unacknowledged RADIUS messages that the plug-in receives from the RADIUS server before it discards new messages.

```
[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set maximum-queue-length maximum-queue-length
```

7. (Optional) Configure the source IP address that the plug-in uses to communicate with the RADIUS server. If you do not specify an address, the global default address is used.

```
[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set bind-address bind-address
```

8. (Optional) Configure the source UDP port or a range of source UDP ports used for communication with the RADIUS server. If you do not specify a UDP port, the global UDP port is used.

```
[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set udp-port udp-port
```

9. Configure the value of the User-Name attribute (RADIUS attribute [1]).

```
[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set username (login-name | accounting-id | auth-user-name |
manager-id)
```

10. Specify whether the SAE sends the MAC address of the subscriber in the Calling-Station-Id attribute.

```
[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set calling-station-id (mac | no)
```

11. Configure the default peer, which is the RADIUS server to which the SAE sends packets for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name basicRadius
radius-accounting]
user@host# set default-peer default-peer
```

12. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name basicRadius
 radius-accounting]
user@host# show
load-balancing-mode failover;
failback-timer -1;
retry-interval 3000;
maximum-queue-length 10000;
username login-name;
calling-station-id no;
default-peer peer1;
```

Configuring Flexible RADIUS Accounting Plug-Ins

Flexible RADIUS accounting plug-ins provide the same features as basic RADIUS accounting plug-ins. In addition, they allow you to customize RADIUS accounting packets that the SAE sends to RADIUS servers. You can specify which fields are included in the RADIUS accounting packets and what information is contained in the fields.

Use the following configuration statements to configure flexible RADIUS accounting plug-ins:

```
shared sae configuration plug-ins name name flex-radius-accounting {
  load-balancing-mode (failover | roundRobin);
  failback-timer failback-timer ;
  timeout timeout ;
  retry-interval retry-interval ;
  maximum-queue-length maximum-queue-length ;
  bind-address bind-address ;
  udp-port udp-port ;
  error-handling (0 | 1);
  default-peer default-peer ;
  template template ;
}
```

To set up flexible RADIUS accounting plug-ins:

1. From configuration mode, access the flexible RADIUS accounting plug-in configuration. In this sample procedure, the plug-in called flexRadiusAct is configured in the west-region SAE group.

```
user@host# edit shared sae group west-region configuration plug-ins name
flexRadiusAct flex-radius-accounting
```

2. Configure the mode for load-balancing RADIUS servers.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct
 flex-radius-accounting]
user@host# set load-balancing-mode (failover | roundRobin)
```

3. Specify if and when the SAE attempts to fail back to the default peer.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct
 flex-radius-accounting]
```

```
user@host# set fallback-timer fallback-timer
```

4. (Optional) Configure the maximum time the SAE waits for a response from a RADIUS server.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct  
flex-radius-accounting]  
user@host# set timeout timeout
```

5. Configure the time the SAE waits for a response from a RADIUS server before it resends the RADIUS packet.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct  
flex-radius-accounting]  
user@host# set retry-interval retry-interval
```

6. Configure the maximum number of unacknowledged RADIUS messages that the plug-in receives from the RADIUS server before it discards new messages.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct  
flex-radius-accounting]  
user@host# set maximum-queue-length maximum-queue-length
```

7. (Optional) Configure the source IP address that the plug-in uses to communicate with the RADIUS server. If you do not specify an address, the global default address is used.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct  
flex-radius-accounting]  
user@host# set bind-address bind-address
```

8. (Optional) Configure the source UDP port or a range of source UDP ports used for communication with the RADIUS server. If you do not specify a UDP port, the global UDP port is used.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct  
flex-radius-accounting]  
user@host# set udp-port udp-port
```

9. Configure the way the SAE handles errors.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct  
flex-radius-accounting]  
user@host# set error-handling (0 | 1)
```

10. Configure the name of the RADIUS server to which the SAE sends packets for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct  
flex-radius-accounting]  
user@host# set default-peer default-peer
```

11. Configure the name of the RADIUS packet template that defines attributes for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name flexRadiusAct
flex-radius-accounting]
user@host# set template template
```

12. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name
flexRadiusAct flex-radius-accounting]
user@host# show
load-balancing-mode failover;
failback-timer -1;
timeout 15000;
retry-interval 3000;
maximum-queue-length 10000;
error-handling 0;
default-peer peer2;
template stdAcct;
peer-group peer2 {
  server-address 10.10.1.1;
  server-port 1818;
  secret *****;
}
```

Configuring Custom RADIUS Accounting-Plug-Ins

The custom RADIUS accounting plug-ins provide the same functions as the flexible RADIUS accounting plug-ins, but are designed to deliver better system performance. To use a custom plug-in, you must provide a Java class that implements the service provider interface (SPI) defined in the RADIUS client library. Use this SPI to specify which fields and field values to include in RADIUS accounting packets. The RADIUS client library is part of the SAE core application programming interface (API).

See the documentation for the RADIUS client library in the SAE core API documentation on the Juniper Networks Web site at

<http://www.juniper.net/techpubs/software/management/src/api-index.html>

For a sample implementation, see the `SDK+AppSupport+Demos+Samples.tar.gz` file on the Juniper Networks Web site at: <https://www.juniper.net/support/csc/swdist-ex/src.html>. The application is located the following directory:

`SDK/plugin/java/src/net/juniper/smg/sample/radiuslib/RadiusPacketHandlerImpl.java`.

Use the following configuration statements to set up custom RADIUS accounting plug-ins:

```
shared sae configuration plug-ins name name custom-radius-accounting {
  java-class-radius-packet-handler java-class-radius-packet-handler ;
  class-path-radius-packet-handler class-path-radius-packet-handler ;
  append-acct-status-type-attribute;
  require-mandatory-attributes;
  load-balancing-mode (failover | roundRobin);
  failback-timer failback-timer ;
```

```

timeout timeout ;
retry-interval retry-interval ;
maximum-queue-length maximum-queue-length ;
bind-address bind-address ;
udp-port udp-port ;
default-peer default-peer ;
}

```

To set up custom RADIUS accounting plug-ins:

1. From configuration mode, access the custom RADIUS accounting plug-in configuration. In this sample procedure, the plug-in called customRadiusAct is configured in the west-region SAE group.

```

user@host# edit shared sae group west-region configuration plug-ins name
customRadiusAct custom-radius-accounting

```

2. Configure the name of the Java class that implements the RadiusPacketHandler interface in the RADIUS client library.

```

[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set java-class-radius-packet-handler java-class-radius-packet-handler

```

3. Configure the URLs that identify a location from which Java classes are loaded when the plug-in is initialized.

```

[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set class-path-radius-packet-handler class-path-radius-packet-handler

```

4. (Optional) Enable the plug-in to include the Acct-Status-Type attribute in a RADIUS accounting request packet.

```

[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set append-acct-status-type-attribute

```

5. (Optional) Specify that a RADIUS authentication or accounting request must contain all mandatory RADIUS attributes before sending the request packet.

```

[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set require-mandatory-attributes

```

6. Configure the mode for load-balancing RADIUS servers.

```

[edit shared sae group west-region configuration plug-ins name customRadiusAct
 custom-radius-accounting]
user@host# set load-balancing-mode (failover | roundRobin)

```

7. Specify if and when the SAE attempts to fail back to the default peer.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
custom-radius-accounting]
user@host# set fallback-timer fallback-timer
```

8. (Optional) Configure the maximum time the SAE waits for a response from a RADIUS server.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
custom-radius-accounting]
user@host# set timeout timeout
```

9. Configure the time the SAE waits for a response from a RADIUS server before it resends the RADIUS packet.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
custom-radius-accounting]
user@host# set retry-interval retry-interval
```

10. Configure the maximum number of unacknowledged RADIUS messages that the plug-in receives from the RADIUS server before it discards new messages.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
custom-radius-accounting]
user@host# set maximum-queue-length maximum-queue-length
```

11. (Optional) Configure the source IP address that the plug-in uses to communicate with the RADIUS server. If you do not specify an address, the global default address is used.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
custom-radius-accounting]
user@host# set bind-address bind-address
```

12. (Optional) Configure the source UDP port or a range of source UDP ports used for communication with the RADIUS server. If you do not specify a UDP port, the global UDP port is used.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
custom-radius-accounting]
user@host# set udp-port udp-port
```

13. Configure the name of the RADIUS server to which the SAE sends packets for this plug-in.

```
[edit shared sae group west-region configuration plug-ins name customRadiusAct
custom-radius-accounting]
user@host# set default-peer default-peer
```

14. (Optional) From operational mode, verify your configuration.

```
[edit shared sae group west-region configuration plug-ins name
customRadiusAct custom-radius-accounting]
user@host# show
```

```
java-class-radius-packet-handler  
net.juniper.smgmt.radius.RadiusPacketHandlerImpl;  
append-acct-status-type-attribute;  
load-balancing-mode failover;  
failback-timer -1;  
timeout 15000;  
retry-interval 3000;  
maximum-queue-length 10000;  
default-peer peer3;
```

- Related Topics**
- Activating and Tracking Service Sessions
 - Types of Tracking Plug-Ins

