

# Event and Accounting Logs

---

## In This Chapter

This chapter provides information about configuring event and accounting logs in the system.

Topics in this chapter include:

- [Logging Overview on page 380](#)
- [Log Destinations on page 382](#)
- [Event Logs on page 387](#)
  - [Event Sources on page 388](#)
  - [Event Control on page 389](#)
  - [Log Manager and Event Logs on page 391](#)
  - [Event Filter Policies on page 392](#)
  - [Event Log Entries on page 393](#)
  - [Simple Logger Event Throttling on page 395](#)
  - [Default System Log on page 396](#)
  - [Event Handling System on page 396](#)
- [Accounting Logs on page 398](#)
  - [Accounting Records on page 398](#)
  - [Accounting Files on page 415](#)
  - [Design Considerations on page 415](#)
- [Configuration Notes on page 421](#)

## Logging Overview

The two primary types of logging supported in the OS are event logging and accounting logs.

Event logging controls the generation, dissemination and recording of system events for monitoring status and troubleshooting faults within the system. The OS groups events into three major categories or event sources:

- Security events — Events that pertain to attempts to breach system security.
- Change events — Events that pertain to the configuration and operation of the node.
- Main events — Events that pertain to applications that are not assigned to other event categories/sources.
- Debug events — Events that pertain to trace or other debugging information.

The following are events within the OS and have the following characteristics:

- A time stamp in UTC or local time.
- The generating application.
- A unique event ID within the application.
- The VRF-ID.
- A subject identifying the affected object.
- A short text description.

Event control assigns the severity for each application event and whether the event should be generated or suppressed. The severity numbers and severity names supported in the OS conform to ITU standards M.3100 X.733 & X.21 and are listed in [Table 39](#).

**Table 39: Event Severity Levels**

Severity Number	Severity Name
1	cleared
2	indeterminate (info)
3	critical
4	major
5	minor
6	warning

Events that are suppressed by event control will not generate any event log entries. Event control maintains a count of the number of events generated (logged) and dropped (suppressed) for each application event. The severity of an application event can be configured in event control.

An event log within the OS associates the event sources with logging destinations. Examples of logging destinations include, the console session, a specific telnet or SSH session, memory logs, file destinations, SNMP trap groups and syslog destinations. A log filter policy can be associated with the event log to control which events will be logged in the event log based on combinations of application, severity, event ID range, VRF ID, and the subject of the event.

The OS accounting logs collect comprehensive accounting statistics to support a variety of billing models. The routers collect accounting data on services and network ports on a per-service class basis. In addition to gathering information critical for service billing, accounting records can be analyzed to provide insight about customer service trends for potential service revenue opportunities. Accounting statistics on network ports can be used to track link utilization and network traffic pattern trends. This information is valuable for traffic engineering and capacity planning within the network core.

Accounting statistics are collected according to the parameters defined within the context of an accounting policy. Accounting policies are applied to customer Service Access Points (SAPs) and network ports. Accounting statistics are collected by counters for individual service queues defined on the customer's SAP or by the counters within forwarding class (FC) queues defined on the network ports.

The type of record defined within the accounting policy determines where a policy is applied, what statistics are collected and time interval at which to collect statistics.

The only supported destination for an accounting log is a compact flash system device (cf1 or cf2). Accounting data is stored within a standard directory structure on the device in compressed XML format.

## Log Destinations

Both event logs and accounting logs use a common mechanism for referencing a log destination. routers support the following log destinations:

- [Console on page 382](#)
- [Session on page 382](#)
- [Memory Logs on page 382](#)
- [Log Files on page 383](#)
- [SNMP Trap Group on page 385](#)
- [Syslog on page 385](#)

Only a single log destination can be associated with an event log or with an accounting log. An event log can be associated with multiple event sources, but it can only have a single log destination.

A file destination is the only type of log destination that can be configured for an accounting log.

---

### Console

Sending events to a console destination means the message will be sent to the system console. The console device can be used as an event log destination.

---

### Session

A session destination is a temporary log destination which directs entries to the active telnet or SSH session for the duration of the session. When the session is terminated, for example, when the user logs out, the “to session” configuration is removed. Event logs configured with a session destination are stored in the configuration file but the “to session” part is not stored. Event logs can direct log entries to the session destination.

---

### Memory Logs

A memory log is a circular buffer. When the log is full, the oldest entry in the log is replaced with the new entry. When a memory log is created, the specific number of entries it can hold can be specified, otherwise it will assume a default size. An event log can send entries to a memory log destination.

## Log Files

Log files can be used by both event logs and accounting logs and are stored on the compact flash devices (specifically cf1: or cf2:) in the file system. It is recommended that event and accounting logs not be configured on the cf3: device that is used for software images and bootup configuration.

A log file is identified with a single log file ID, but a log file will generally be composed of a number individual files in the file system. A log file is configured with a rollover parameter, expressed in minutes, which represents the length of time an individual log file should be written to before a new file is created for the relevant log file ID. The rollover time is checked only when an update to the log is performed. Thus, complying to this rule is subject to the incoming rate of the data being logged. For example, if the rate is very low, the actual rollover time may be longer than the configured value.

The retention time for a log file specifies the amount of time the file should be retained on the system based on the creation date and time of the file.

When a log file is created, only the compact flash device for the log file is specified. Log files are created in specific subdirectories with standardized names depending on the type of information stored in the log file.

Event log files are always created in the **log** directory on the specified compact flash device. The naming convention for event log files is:

```
log eeff-timestamp
```

where:

*ee* is the event log ID

*ff* is the log file destination ID

*timestamp* is the timestamp when the file is created in the form of *yyyymmdd-hhmmss*  
where:

*yyyy* is the four-digit year (for example, 2007)

*mm* is the two digit number representing the month (for example, 12 for December)

*dd* is the two digit number representing the day of the month (for example, 03 for the 3rd of the month)

*hh* is the two digit hour in a 24-hour clock (for example, 04 for 4 a.m.)

*mm* is the two digit minute (for example, 30 for 30 minutes past the hour)

*ss* is the two digit second (for example, 14 for 14)

Accounting log files are created in the `\act-collect` directory on a compact flash device (specifically *cf1* or *cf2*). The naming convention for accounting log files is nearly the same as for log files except the prefix **act** is used instead of the prefix **log**. The naming convention for accounting logs is:

```
act aaff-timestamp.xml.gz
```

where:

*aa* is the accounting policy ID

*ff* is the log file destination ID

*timestamp* is the timestamp when the file is created in the form of *yyyymmdd-hhmmss*

where:

*yyyy* is the four-digit year (for example, 2007)

*mm* is the two digit number representing the month (for example, 12 for December)

*dd* is the two digit number representing the day of the month (for example, 03 for the 3rd of the month)

*hh* is the two digit hour in a 24-hour clock (for example, 04 for 4 a.m.)

*mm* is the two digit minute (for example, 30 for 30 minutes past the hour)

*ss* is the two digit second (for example, 14 for 14 seconds)

Accounting logs are `.xml` files created in a compressed format and have a `.gz` extension.

The `\act-collect` directory is where active accounting logs are written. When an accounting log is rolled over, the active file is closed and archived in the `\act` directory before a new active accounting log file created in `\act-collect`.

## SNMP Trap Group

An event log can be configured to send events to SNMP trap receivers by specifying an SNMP trap group destination.

An SNMP trap group can have multiple trap targets. Each trap target can have different operational parameters.

A trap destination has the following properties:

- The IP address of the trap receiver.
- The UDP port used to send the SNMP trap.
- SNMP version (v1, v2c, or v3) used to format the SNMP notification.
- SNMP community name for SNMPv1 and SNMPv2c receivers.
- Security name and level for SNMPv3 trap receivers.

For SNMP traps that will be sent out-of-band through the Management Ethernet port on the SF/CPM, the source IP address of the trap is the IP interface address defined on the Management Ethernet port. For SNMP traps that will be sent in-band, the source IP address of the trap is the system IP address of the router.

Each trap target destination of a trap group receives the identical sequence of events as defined by the log ID and the associated sources and log filter applied.

---

## Syslog

An event log can be configured to send events to one syslog destination. Syslog destinations have the following properties:

- Syslog server IP address.
- The UDP port used to send the syslog message.
- The Syslog Facility Code (0 - 23) (default 23 - local 7).
- The Syslog Severity Threshold (0 - 7) - events exceeding the configured level will be sent.

Because syslog uses eight severity levels whereas the router uses six internal severity levels, the severity levels are mapped to syslog severities. Table 40 displays the severity level mappings to syslog severities.

**Table 40: Router to Syslog Severity Level Mappings**

SR OS Event Severity	Syslog Severity Numerical Code	Syslog Severity name	Syslog Severity Definition
--	0	emergency	System is unusable
critical (3)	1	alert	Action must be taken immediately
major (4)	2	critical	Critical conditions
minor (5)	3	error	Error conditions
warning (6)	4	warning	Warning conditions
--	5	notice	Normal but significant condition
cleared (1) indeterminate (2)	6	info	Informational messages
--	7	debug	Debug-level messages

The general format of an SR OS syslog message is as follows (see RFC3164). Note that the ‘<’ and ‘>’ are informational delimiters to make reading and understanding the format easier and they do not appear in the actual syslog message except as part of the ‘PRI’:

```
<HEADER> <log-prefix>: <PRI> <MSG>
```

where:

*HEADER* is MMM DD HH:MM:SS <source IP addr>

*log-prefix* is an optional 32 characters of text as configured in the log-prefix command. A ‘.’ will not appear at this point in the message if no log-prefix is configured.

<*PRI*> (the ‘<’ and ‘>’ are included in the syslog message) is the configured facility\*8+severity (as described in the System Management Guide and RFC3164)

<*MSG*> is <router-name> <application>-<severity>-<Event Name>-<Event ID> [<subject>]: <description>

where:

*router-name* is vprn1, vprn2, ... | Base | management | vpls-management

*subject* may be empty resulting in []:

Example:

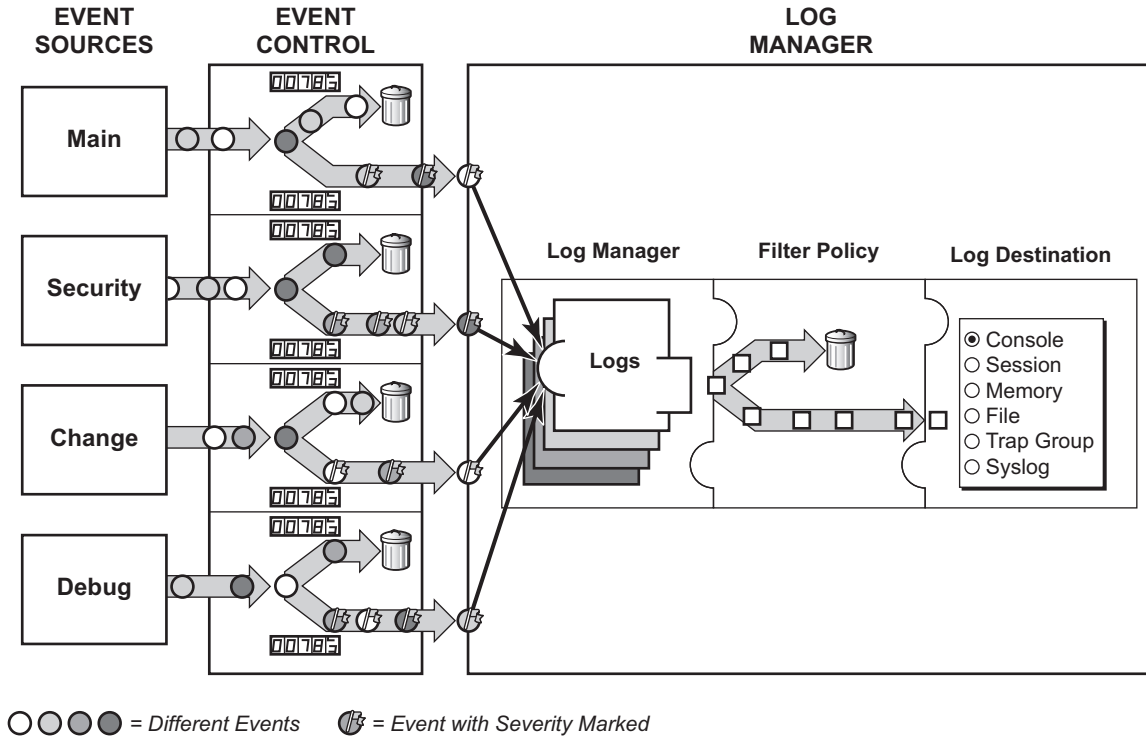
```
Jun  4 07:05:07 10.252.30.36 133 vprn110 RADIUS-MINOR-
tmnxRadSrvPlcySrvOperStateCh-2001 []: The operational state of RADIUS server 1
(address=10.63.211.22) in RADIUS server policy "rad-server-pol-H3Gaaa" changed
to out-of-service.
```



# Event Logs

Event logs are the means of recording system generated events for later analysis. Events are messages generated by the system by applications or processes within the router.

Figure 12 depicts a function block diagram of event logging.



CLI0001B

Figure 12: Event Logging Block Diagram

## Event Sources

In [Figure 12](#), the event sources are the main categories of events that feed the log manager.

- **Security** — The security event source is all events that affect attempts to breach system security such as failed login attempts, attempts to access MIB tables to which the user is not granted access or attempts to enter a branch of the CLI to which access has not been granted. Security events are generated by the SECURITY application and the authenticationFailure event in the SNMP application.
- **Change** — The change activity event source is all events that directly affect the configuration or operation of the node. Change events are generated by the USER application. The Change event stream also includes the tmnxConfigModify (#2006), tmnxConfigCreate (#2007), tmnxConfigDelete (#2008) and tmnxStateChange (#2009) change events from the SYSTEM application.
- **Debug** — The debug event source is the debugging configuration that has been enabled on the system. Debug events are generated by the DEBUG application.
- **Main** — The main event source receives events from all other applications within the router.

Examples of applications within the system include IP, MPLS, OSPF, CLI, services, etc. The following example displays a partial sample of the **show log applications** command output which displays all applications.

```
*A:ALA-48# show log applications
=====
Log Event Application Names
=====
Application Name
-----
...
BGP
CCAG
CFLOWD
CHASSIS
...
MPLS
MSDP
NTP
...
TOD
USER
VRRP
VRTR
=====
*A:ALA-48#
```

## Event Control

Event control pre-processes the events generated by applications before the event is passed into the main event stream. Event control assigns a severity to application events and can either forward the event to the main event source or suppress the event. Suppressed events are counted in event control, but these events will not generate log entries as it never reaches the log manager.

Simple event throttling is another method of event control and is configured similarly to the generation and suppression options. See [Simple Logger Event Throttling on page 395](#).

Events are assigned a default severity level in the system, but the application event severities can be changed by the user.

Application events contain an event number and description that explains why the event is generated. The event number is unique within an application, but the number can be duplicated in other applications.

The following example, generated by querying event control for application generated events, displays a partial list of event numbers and names.

```
router# show log event-control
=====
Log Events
=====
Application
ID#      Event Name                               P  g/s    Logged    Dropped
-----
BGP:
  2001  bgpEstablished                          MI  gen     1         0
  2002  bgpBackwardTransition                   WA  gen     7         0
  2003  tBgpMaxPrefix90                         WA  gen     0         0
...
CCAG:
CFLOWD:
  2001  cflowdCreated                           MI  gen     1         0
  2002  cflowdCreateFailure                     MA  gen     0         0
  2003  cflowdDeleted                           MI  gen     0         0
...
CHASSIS:
  2001  cardFailure                             MA  gen     0         0
  2002  cardInserted                            MI  gen     4         0
  2003  cardRemoved                             MI  gen     0         0
...
'''
DEBUG:
L 2001  traceEvent                              MI  gen     0         0
DOT1X:
FILTER:
  2001  filterPBRPacketsDropped                 MI  gen     0         0
IGMP:
  2001  vRtrIcmpIfRxQueryVerMismatch            WA  gen     0         0
  2002  vRtrIcmpIfCModeRxQueryMismatch          WA  gen     0         0
IGMP_SNOOPING:
```

## Event Control

```
IP:
L 2001 clearRTMError          MI gen      0      0
L 2002 ipEtherBroadcast      MI gen      0      0
L 2003 ipDuplicateAddress    MI gen      0      0
...
ISIS:
  2001 vRtrIisisDatabaseOverload WA gen      0      0
```

## Log Manager and Event Logs

Events that are forwarded by event control are sent to the log manager. The log manager manages the event logs in the system and the relationships between the log sources, event logs and log destinations, and log filter policies.

An event log has the following properties:

- A unique log ID  
The log ID is a short, numeric identifier for the event log. A maximum of ten logs can be configured at a time.
- One or more log sources  
The source stream or streams to be sent to log destinations can be specified. The source must be identified before the destination can be specified. The events can be from the main event stream, events in the security event stream, or events in the user activity stream.
- One event log destination  
A log can only have a single destination. The destination for the log ID destination can be one of console, session, syslog, snmp-trap-group, memory, or a file on the local file system.
- An optional event filter policy  
An event filter policy defines whether to forward or drop an event or trap-based on match criteria.

## Event Filter Policies

The log manager uses event filter policies to allow fine control over which events are forwarded or dropped based on various criteria. Like other policies with the 7750 SR, filter policies have a default action. The default actions are either:

- Forward
- Drop

Filter policies also include a number of filter policy entries that are identified with an entry ID and define specific match criteria and a forward or drop action for the match criteria.

Each entry contains a combination of matching criteria that define the application, event number, router, severity, and subject conditions. The entry's action determines how the packets should be treated if they have met the match criteria.

Entries are evaluated in order from the lowest to the highest entry ID. The first matching event is subject to the forward or drop action for that entry.

Valid operators are displayed in [Table 41](#):

**Table 41: Valid Filter Policy Operators**

Operator	Description
eq	equal to
neq	not equal to
lt	less than
lte	less than or equal to
gt	greater than
gte	greater than or equal to

A match criteria entry can include combinations of:

- Equal to or not equal to a given system application.
- Equal to, not equal to, less than, less than or equal to, greater than or greater than or equal to an event number within the application.
- Equal to, not equal to, less than, less than or equal to, greater than or greater than or equal to a severity level.
- Equal to or not equal to a router name string or regular expression match.
- Equal to or not equal to an event subject string or regular expression match.

## Event Log Entries

Log entries that are forwarded to a destination are formatted in a way appropriate for the specific destination whether it be recorded to a file or sent as an SNMP trap, but log event entries have common elements or properties. All application generated events have the following properties:

- A time stamp in UTC or local time.
- The generating application.
- A unique event ID within the application.
- A router name identifying the VRF-ID that generated the event.
- A subject identifying the affected object.
- A short text description.

The general format for an event in an event log with either a memory, console or file destination is as follows.

```
nnnn YYYY/MM/DD HH:MM:SS.SS <severity><application> # <event_id> <router-name> <subject>
<message>
```

The following is an event log example:

```
475 2006/11/27 00:19:40.38 WARNING: SNMP #2007 Base 1/1/1
"interface 1/1/1 came up"
```

The specific elements that compose the general format are described in [Table 42](#).

**Table 42: Log Entry Field Descriptions**

Label	Description
nnnn	The log entry sequence number.
YYYY/MM/DD	The UTC date stamp for the log entry. <i>YYYY</i> — Year <i>MM</i> — Month <i>DD</i> — Date
HH:MM:SS.SS	The UTC time stamp for the event. <i>HH</i> — Hours (24 hour format) <i>MM</i> — Minutes <i>SS.SS</i> — Seconds
<severity>	The severity level name of the event. CLEARED — A cleared event (severity number 1). INFO — An indeterminate/informational severity event (severity level 2). CRITICAL — A critical severity event (severity level 3). MAJOR — A major severity event (severity level 4). MINOR — A minor severity event (severity level 5). WARNING — A warning severity event (severity 6).

**Table 42: Log Entry Field Descriptions (Continued)**

<b>Label</b>	<b>Description</b>
<application>	The application generating the log message.
<event_id>	The application's event ID number for the event.
<router>	The router name representing the VRF-ID that generated the event.
<subject>	The subject/affected object for the event.
<message>	A text description of the event.



## Simple Logger Event Throttling

Simple event throttling provides a mechanism to protect event receivers from being overloaded when a scenario causes many events to be generated in a very short period of time. A throttling rate, # events/# seconds, can be configured. Specific event types can be configured to be throttled. Once the throttling event limit is exceeded in a throttling interval, any further events of that type cause the dropped events counter to be incremented. Dropped events counts are displayed by the **show>log>event-control** context. Events are dropped before being sent to one of the logger event collector tasks. There is no record of the details of the dropped events and therefore no way to retrieve event history data lost by this throttling method.

A particular event type can be generated by multiple managed objects within the system. At the point this throttling method is applied the logger application has no information about the managed object that generated the event and cannot distinguish between events generated by object “A” from events generated by object “B”. If the events have the same event-id, they are throttled regardless of the managed object that generated them. It also does not know which events may eventually be logged to destination log-id <n> from events that will be logged to destination log-id <m>.

Throttle rate applies commonly to all event types. It is not configurable for a specific event-type.

A timer task checks for events dropped by throttling when the throttle interval expires. If any events have been dropped, a TIMETRA-SYSTEM-MIB::tmnxTrapDropped notification is sent.

## Default System Log

Log 99 is a pre-configured memory-based log which logs events from the main event source (not security, debug, etc.). Log 99 exists by default.

The following example displays the log 99 configuration.

```
ALA-1>config>log# info detail
#-----
echo "Log Configuration "
#-----
...
    snmp-trap-group 7
    exit
...
    log-id 99
        description "Default system log"
        no filter
        from main
        to memory 500
        no shutdown
    exit
#-----
ALA-1>config>log#
```

## Event Handling System

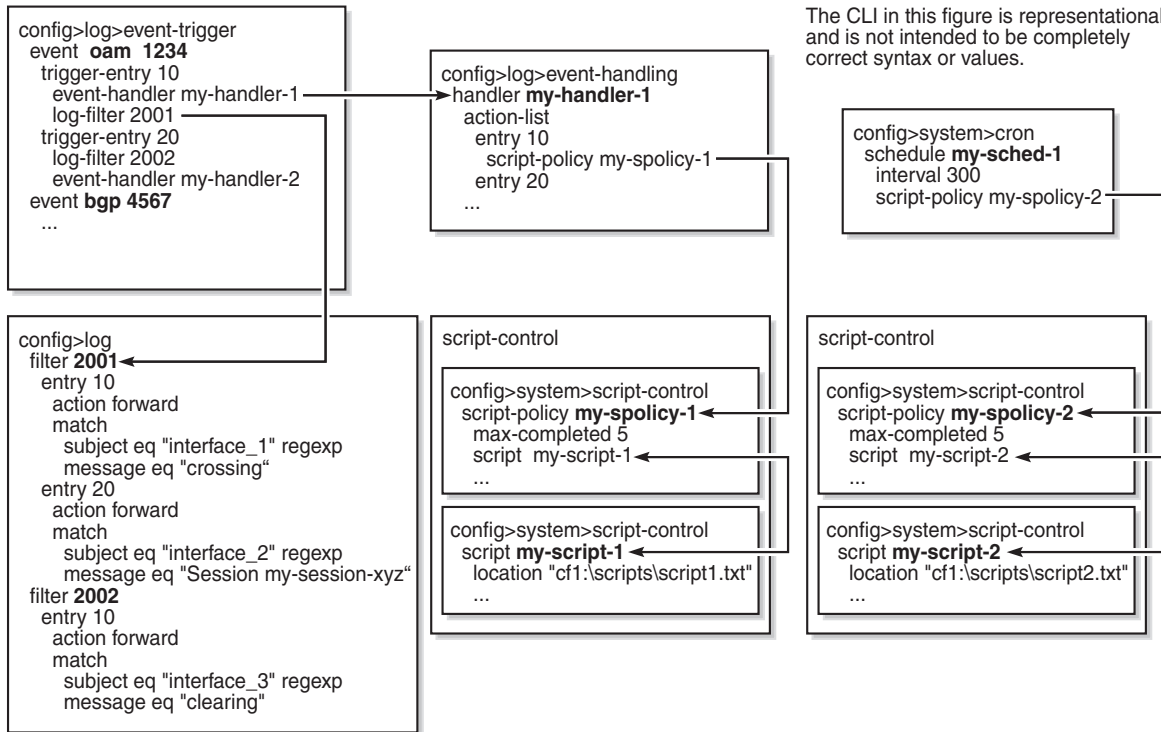
The Event Handling System (EHS) is a tool that allows operator-defined behavior to be configured on the router. EHS adds user-controlled programmatic exception handling by allowing a CLI script to be executed upon the detection of a log event (the 'trigger'). Regexp style expression matching is available on various fields in the log event to give flexibility in the trigger definition.

EHS handler objects are used to tie together:

- trigger events (typically log events that match some configurable criteria)
- a set of actions to perform (typically one or more CLI scripts)

EHS, along with CRON, makes use of the generic SR OS CLI script-control functions for scripts. Any command available in CLI (with some limited exceptions such as 'candidate' commands) can be executed in a script as the result of an EHS handler being triggered.

The following figure illustrates the relationships between the different configurable objects used by EHS (and CRON).



24884

**Figure 13: EHS Object Relationships**

Complex rules can be configured to match on log events as a trigger for an EHS handler.

When a log event is generated in SR OS it will be subject to discard via suppression and throttling (**config>log>event-control**) before it is evaluated as a trigger for EHS:

- EHS will not trigger on log events that are suppressed through **config>log>event-control**
- EHS will not trigger on log events that are throttled by the logger

EHS will trigger on log events that are dropped by user configured log filters that are assigned to individual logs (**config>log>filter**). The EHS event trigger logic occurs before the distribution of log event streams into individual logs.

## Accounting Logs

Before an accounting policy can be created a target log file must be created to collect the accounting records. The files are stored in system memory on compact flash (*cf1:* or *cf2:*) in a compressed (tar) XML format and can be retrieved using FTP or SCP.

A file ID can only be assigned to either one event log ID or one accounting log.

---

## Accounting Records

An accounting policy must define a record name and collection interval. Only one record name can be configured per accounting policy. Also, a record name can only be used in one accounting policy.

The record name, sub-record types, and default collection period for service and network accounting policies are shown below. [Table 45](#), [Table 46](#), and [Table 47](#) provide field descriptions.

**Table 43: Accounting Record Name and Collection Periods**

Record Name	Sub-Record Types	Accounting Object	Default Collection Period (minutes)
service-ingress-octets	sio	SAP	5
service-egress-octets	seo	SAP	5
service-ingress-packets	sip	SAP	5
service-egress-packets	sep	SAP	5
network-ingress-octets	nio	Network port	15
network-egress-octets	neo	Network port	15
network-egress-packets	nep	Network port	15
network-ingress-packets	nio	Network port	15
compact-service-ingress-octets	ctSio	SAP	5
combined-service-ingress	cmSipo	SAP	5
combined-network-ing-egr-octets	cmNio & cmNeo	Network port	15
combined-service-ing-egr-octets	cmSio & cmSeo	SAP	5
complete-network-ingr-egr	cpNipo & cpNepo	Network port	15
complete-service-ingress-egress	cpSipo & cpSepo	SAP	5
combined-sdp-ingress-egress	cmSdpipo and cmSdpepo	SDP and SDP binding	5
complete-sdp-ingress-egress	cmSdpipo, cmSdpepo, cpSdpipo and cpSdpepo	SDP and SDP binding	5

**Table 43: Accounting Record Name and Collection Periods (Continued)**

Record Name	Sub-Record Types	Accounting Object	Default Collection Period (minutes)
complete-subscriber-ingress-egress	cpSBipo & cpSBepo	Subscriber profile	5
aa-protocol	aaProt	AA ISA Group	15
aa-application	aaApp	AA ISA Group	15
aa-app-group	aaAppGrp	AA ISA Group	15
aa-subscriber-protocol	aaSubProt	Special study AA subscriber	15
aa-subscriber-application	aaSubApp	Special study AA subscriber	15
custom-record-aa-sub	aaSubCustom	AA subscriber	15
combined-mpls-lsp-egress	mplsLspEgr	LSP	5
combined-mpls-lsp-ingress	mplsLspIn	LSP	5
saa	saa png trc hop	SAA or SAA test	5
complete-ethernet-port	enet	Ethernet port	15

When creating accounting policies, one service accounting policy and one network accounting policy can be defined as default. If statistics collection is enabled on a SAP or network port and no accounting policy is applied, then the respective default policy is used. If no default policy is defined, then no statistics are collected unless a specifically defined accounting policy is applied.

Each accounting record name is composed of one or more sub-records which is in turn composed of multiple fields.

Refer to the Application Assurance Statistics Fields Generated per Record table in the 7750 SR-Series OS Integrated Services Adapter Guide for fields names for Application Assurance records.

**Table 44: Accounting Record Name Details**

Record Name	Sub-Record	Field	Field Description
Service-ingress-octets (sio) (**)	sio	svc	SvcId
		sap	SapId
		qid	QueueId
		hoo	OfferedHiPrioOctets
		hod	DroppedHiPrioOctets
		loo	LowOctetsOffered
		lod	LowOctetsDropped
		uco	UncoloredOctetsOffered
		iof	InProfileOctetsForwarded
		oof	OutOfProfileOctetsForwarded

**Table 44: Accounting Record Name Details (Continued)**

Record Name	Sub-Record	Field	Field Description
Service-egress-octets (seo) (**)	seo	svc	SvcId
		sap	SapId
		qid	QueueId
		iof	InProfileOctetsForwarded
		iod	InProfileOctetsDropped
		oof	OutOfProfileOctetsForwarded
		ood	OutOfProfileOctetsDropped
Service-ingress-packets (sip) (*) (**)	sip	svc	SvcId
		sap	SapId
		qid	QueueId
		hpo	HighPktsOffered
		hpd	HighPktsDropped
		lpo	LowPktsOffered
		lpd	LowPktsDropped
		ucp	UncoloredPacketsOffered
		ipf	InProfilePktsForwarded
		opf	OutOfProfilePktsForwarded
Service-egress-packets (sep) (*) (**)	sep	svc	SvcId
		sap	SapId
		qid	QueueId
		ipf	InProfilePktsForwarded
		ipd	InProfilePktsDropped
		opf	OutOfProfilePktsForwarded
		opd	OutOfProfilePktsDropped
Network-ingress-octets (nio)	nio	port	PortId
		qid	QueueId
		iof	InProfileOctetsForwarded
		iod	InProfileOctetsDropped
		oof	OutOfProfileOctetsForwarded
		ood	OutOfProfileOctetsDropped
Network-egress-octets (neo)	neo	port	PortId
		qid	QueueId
		iof	InProfileOctetsForwarded
		iod	InProfileOctetsDropped
		oof	OutOfProfileOctetsForwarded
		ood	OutOfProfileOctetsDropped

**Table 44: Accounting Record Name Details (Continued)**

Record Name	Sub-Record	Field	Field Description
Network-ingress-packets (nip)	nip	port	PortId
		qid	QueueId
		ipf	InProfilePktsForwarded
		ipd	InProfilePktsDropped
		opf	OutOfProfilePktsForwarded
		opd	OutOfProfilePktsDropped
Network Egress Packets (nep)	nep	port	PortId
		qid	QueueId
		ipf	InProfilePktsForwarded
		ipd	InProfilePktsDropped
		opf	OutOfProfilePktsForwarded
		opd	OutOfProfilePktsDropped
Compact-service-ingress-octets (ctSio)	ctSio	svc	SvcId
		sap	SapId
		qid	QueueId
		hoo	OfferedHiPrioOctets
		hod	DroppedHiPrioOctets
		loo	LowOctetsOffered
		lod	LowOctetsDropped
		uco	UncoloredOctetsOffered
Combined-service-ingress (cmSipo)	cmSipo	svc	SvcId
		sap	SapId
		qid	QueueId
		hpo	HighPktsOffered
		hpd	HighPktsDropped
		lpo	LowPktsOffered
		lpd	LowPktsDropped
		ucp	UncoloredPacketsOffered
		hoo	OfferedHiPrioOctets
		hod	DroppedHiPrioOctets
		loo	LowOctetsOffered
		lod	LowOctetsDropped
		uco	UncoloredOctetsOffered
		ipf	InProfilePktsForwarded
		opf	OutOfProfilePktsForwarded
		iof	InProfileOctetsForwarded
oof	OutOfProfileOctetsForwarded		

**Table 44: Accounting Record Name Details (Continued)**

Record Name	Sub-Record	Field	Field Description	
Combined-network-ing-egr-octets (cmNio & cmNeo )	cmNio	port	PortId	
		qid	QueueId	
		iof	InProfileOctetsForwarded	
		iod	InProfileOctetsDropped	
		oof	OutOfProfileOctetsForwarded	
		ood	OutOfProfileOctetsDropped	
	cmNeo	port	PortId	
		qid	QueueId	
		iof	InProfileOctetsForwarded	
		iod	InProfileOctetsDropped	
		oof	OutOfProfileOctetsForwarded	
		ood	OutOfProfileOctetsDropped	
	Combined-service-ingr-egr-octets (cmSio & CmSeo)	cmSio	svc	SvcId
			sap	SapId
qid			QueueId	
hoo			OfferedHiPrioOctets	
hod			DroppedHiPrioOctets	
loo			LowOctetsOffered	
lod			LowOctetsDropped	
uco			UncoloredOctetsOffered	
iof			InProfileOctetsForwarded	
oof			OutOfProfileOctetsForwarded	
cmSeo			svc	SvcId
		sap	SapId	
		qid	QueueId	
		iof	InProfileOctetsForwarded	
		iod	InProfileOctetsDropped	
		oof	OutOfProfileOctetsForwarded	
		ood	OutOfProfileOctetsDropped	



**Table 44: Accounting Record Name Details (Continued)**

Record Name	Sub-Record	Field	Field Description
Complete-network-ingr-egr (cpNipo & cpNepo)	cpNipo	port	PortId
		qid	QueueId
		ipf	InProfilePktsForwarded
		ipd	InProfilePktsDropped
		opf	OutOfProfilePktsForwarded
		opd	OutOfProfilePktsDropped
		iof	InProfileOctetsForwarded
		iod	InProfileOctetsDropped
		oof	OutOfProfileOctetsForwarded
		ood	OutOfProfileOctetsDropped
	cpNepo	port	PortId
		qid	QueueId
		ipf	InProfilePktsForwarded
		ipd	InProfilePktsDropped
		opf	OutOfProfilePktsForwarded
		opd	OutOfProfilePktsDropped
		iof	InProfileOctetsForwarded
		iod	InProfileOctetsDropped
		oof	OutOfProfileOctetsForwarded
		ood	OutOfProfileOctetsDropped

**Table 44: Accounting Record Name Details (Continued)**

Record Name	Sub-Record	Field	Field Description
Complete-service-ingress-egress (cpSipo & cpSepa)	cpSipo	svc	SvcId
		sap	SapId
		pid	PolicerId
		hpo	HighPktsOffered
		hpd	HighPktsDropped
		lpo	LowPktsOffered
		lpd	LowPktsDropped
		ucp	UncoloredPacketsOffered
		hoo	OfferedHiPrioOctets
		hod	DroppedHiPrioOctets
		loo	LowOctetsOffered
		lod	LowOctetsDropped
		uco	UncoloredOctetsOffered
		apo	AllPacketsOffered
		aoo	AllOctetsOffered
		apd	AllPacketsDropped
		aod	AllOctetsDropped
		apf	AllPacketsForwarded
		aof	AllOctetsForwarded
		ipd	InProfilePktsDropped
		iod	InProfileOctetsDropped
		opd	OutOfProfilePktsDropped
		ood	OutOfProfileOctetsDropped
		hpf	HighPriorityPacketsForwarded
		hof	HighPriorityOctetsForwarded
		lpf	LowPriorityPacketsForwarded
		lof	LowPriorityOctetsForwarded
		ipf	InProfilePktsForwarded
		opf	OutOfProfilePktsForwarded
		iof	InProfileOctetsForwarded
oof	OutOfProfileOctetsForwarded		

Table 44: Accounting Record Name Details (Continued)

Record Name	Sub-Record	Field	Field Description
Complete-service-ingress-egress (cpSipo & cpSepa) continued	cpSepa	svc	SvcId
		sap	SapId
		qid	QueueId
		ipf	InProfilePktsForwarded
		ipd	InProfilePktsDropped
		opf	OutOfProfilePktsForwarded
		opd	OutOfProfilePktsDropped
		iof	InProfileOctetsForwarded
		iod	InProfileOctetsDropped
		oof	OutOfProfileOctetsForwarded
		ood	OutOfProfileOctetsDropped
Complete-sdp-ingress-egress (cpSdpipo & cpSdpepo)	cpSdpipo	sdp	SdpID
		tpf	TotalPacketsForwarded
		tpd	TotalPacketsDropped
		tof	TotalOctetsForwarded
		tod	TotalOctetsDropped
	cpSdpepo	sdp	SdpID
		tpd	TotalPacketsDropped
		tod	TotalOctetsDropped
Combined-sdp-ingress-egress (cmSdpipo & cmSdpepo)	cmSdpipo	svc	SvcID
		sdp	SdpID
		tpf	TotalPacketsForwarded
		tpd	TotalPacketsDropped
		tof	TotalOctetsForwarded
		tod	TotalOctetsDropped
	cmSdpepo	svc	SvcID
		sdp	SdpID
		tpf	TotalPacketsForwarded
		tof	TotalOctetsForwarded

**Table 44: Accounting Record Name Details (Continued)**

Record Name	Sub-Record	Field	Field Description
Complete-sdp-ingress-egress (cmSdpipo & cmsdpepo) (cpSdpip & cpSdpepo)	cmSdpipo	svc	SvcID
		sdp	SdpID
		tpf	TotalPacketsForwarded
		tpd	TotalPacketsDropped
		tof	TotalOctetsForwarded
		tod	TotalOctetsDropped
	cmSdpepo	svc	SvcID
		sdp	SdpID
		tpf	TotalPacketsForwarded
		tof	TotalOctetsForwarded
	cpSdpipo	sdp	SdpID
		tpf	TotalPacketsForwarded
		tpd	TotalPacketsDropped
		tof	TotalOctetsForwarded
		tod	TotalOctetsDropped
	cpSdpepo	sdp	SdpID
		tpf	TotalPacketsForwarded
		tof	TotalOctetsForwarded

Table 44: Accounting Record Name Details (Continued)

Record Name	Sub-Record	Field	Field Description
Complete-subscriber-ingress-egress (cpSBipo & cpSBepo) (cpSBipoooc & cpSBepoooc) ***	SubscriberInformation	subId	SubscriberId
		subProfile	SubscriberProfile
	Sla-Information****	svc	SvcId
		sap	SapId
		slaProfile	SlaProfile
	cpSBipo	qid	QueueId
		hpo	HighPktsOffered ****
		hpd	HighPktsDropped
		lpo	LowPktsOffered ****
		lpd	LowPktsDropped
		ucp	UncolouredPacketsOffered
		hoo	OfferedHiPrioOctets ****
		hod	DroppedHiPrioOctets
		loo	LowOctetsOffered ****
		lod	LowOctetsDropped
		apo	AllPktsOffered ****
		aoo	AllOctetsOffered ****
		uco	UncolouredOctetsOffered
		ipf	InProfilePktsForwarded
		opf	OutOfProfilePktsForwarded
		iof	InProfileOctetsForwarded
		oof	OutOfProfileOctetsForwarded
		v4pf	IPv4PktsForwarded
		v6pf	IPv6PktsForwarded
		v4pd	IPv4PktsDropped
		v6pd	IPv6PktsDropped
		v4of	IPv4OctetsForwarded
		v6of	IPv6OctetsForwarded
		v4od	IPv4OctetsDropped
	v6od	IPv6OctetsDropped	
	cpSBepo	qid	QueueId
		ipf	InProfilePktsForwarded
		ipd	InProfilePktsDropped
		opf	OutOfProfilePktsForwarded
		opd	OutOfProfilePktsDropped
		iof	InProfileOctetsForwarded
		iod	InProfileOctetsDropped
		oof	OutOfProfileOctetsForwarded
	ood	OutOfProfileOctetsDropped	

**Table 44: Accounting Record Name Details (Continued)**

<b>Record Name</b>	<b>Sub-Record</b>	<b>Field</b>	<b>Field Description</b>
		v4pf	IPv4PktsForwarded
		v6pf	IPv6PktsForwarded
		v4pd	IPv4PktsDropped
		v6pd	IPv6PktsDropped
		v4of	IPv4OctetsForwarded
		v6of	IPv6OctetsForwarded
		v4od	IPv4OctetsDropped
		v6od	IPv6OctetsDropped

Table 44: Accounting Record Name Details (Continued)

Record Name	Sub-Record	Field	Field Description
(continued) Complete-subscriber-ingress-egress (cpSBipo & cpSBepo) (cpSBipoooc & cpSBepooc) ***	cpSBipoooc ***	cid	OverrideCounterId
		apo	AllPktsOffered
		hpd	HighPktsDropped
		lpd	LowPktsDropped
		aoo	AllOctetsOffered
		hod	DroppedHiPrioOctets
		lod	LowOctetsDropped
		ipf	InProfilePktsForwarded
		opf	OutOfProfilePktsForwarded
		iof	InProfileOctetsForwarded
		oof	OutOfProfileOctetsForwarded
		ucp	UncolouredPacketsOffered
		uco	UncolouredOctetsOffered
	cpSBepooc ***	cid	OverrideCounterId
		ipf	InProfilePktsForwarded
		ipd	InProfilePktsDropped
		ofp	OutOfProfilePktsForwarded
		opd	OutOfProfilePktsDropped
		ipd	InProfileOctetsForwarded
		iod	InProfileOctetsDropped
oof		OutOfProfileOctetsForwarded	
ood	OutOfProfileOctetsDropped		
saa	saa	tmd	TestMode
		own	OwnerName
		tst	TestName
		png	PingRun subrecord
		rid	RunIndex
		trr	TestRunResult
		mnr	MinRtt
		mxr	MaxRtt
		avr	AverageRtt
		rss	RttSumOfSquares
		pbr	ProbeResponses
		spb	SentProbes
		mnt	MinOutTt
		mxt	MaxOutTt
		avt	AverageOutTt

**Table 44: Accounting Record Name Details (Continued)**

Record Name	Sub-Record	Field	Field Description
		tss	OutTtSumOfSquares
		mni	MinInTt
		mxi	MaxInTt
		avi	AverageInTt
		iss	InTtSumOfSqr
		ojt	OutJitter
		ijt	InJitter
		rjt	RtJitter
		prt	ProbeTimeouts
		prf	ProbeFailures
	trc	rid	RunIndex
		trr	TestRunResult
		lgp	LastGoodProbe
	hop	hop	TraceHop
		hid	HopIndex
		mnr	MinRtt
		mrx	MaxRtt
		avr	AverageRtt
		rss	RttSumOfSquares
		pbr	ProbeResponses
		spb	SentProbes
		mnt	MinOutTt
		mxt	MaxOutTt
		avt	AverageOutTt
		tss	OutTtSumOfSquares
		mni	MinInTt
		mxi	MaxInTt
		avi	AverageInTt
		iss	InTtSumOfSqr
		ojt	OutJitter
		ijt	InJitter
		rjt	RtJitter
		prt	ProbeTimeouts
		prf	ProbeFailures
tat		TraceAddressType	
tav		TraceAddressValue	



Table 44: Accounting Record Name Details (Continued)

Record Name	Sub-Record	Field	Field Description
Complete-ethernet-port (enet)	enet	port	PortId
		to	EtherStatsOctets
		tp	EtherStatsPkts
		de	EtherStatsDropEvents
		tbcp	EtherStatsBroadcastPkts
		mcp	EtherStatsMulticastPkts
		cae	EtherStatsCRCAlignErrors
		up	EtherStatsUndersizePkts
		op	EtherStatsOversizePkts
		fgm	EtherStatsFragments
		jab	EtherStatsJabbers
		col	EtherStatsCollisions
		p64o	EtherStatsPkts64Octets
		p127o	EtherStatsPkts65to127Octets
		p255o	EtherStatsPkts128to255Octets
		p511o	EtherStatsPkts256to511Octets
		p1023o	EtherStatsPkts512to1023Octets
		p1518o	EtherStatsPkts1024to1518Octets
		po1518o	EtherStatsPktsOver1518Octets
		ae	Dot3StatsAlignmentErrors
		fe	Dot3StatsFCSErrors
		scf	Dot3StatsSingleCollisionFrames
		mcf	Dot3StatsMultipleCollisionFrames
		sqe	Dot3StatsSQETestErrors
		dt	Dot3StatsDeferredTransmissions
		lcc	Dot3StatsLateCollisions
		exc	Dot3StatsExcessiveCollisions
		imt	Dot3StatsInternalMacTransmitErrors
		cse	Dot3StatsCarrierSenseErrors
		ftl	Dot3StatsFrameTooLongs
		imre	Dot3StatsInternalMacReceiveErrors
		se	Dot3StatsSymbolErrors
ipf	Dot3InPauseFrames		
opf	Dot3OutPauseFrames		

(\*) For a SAP in AAL5 SDU mode, packet counters refer to the number of SDU.

(\*) For a SAP in N-to-1 cell mode, packet counters refer to the number of cells.

(\*\*) The number of octets in an ATM sap excludes the Header Error Control (HEC) byte, thus meaning each packet/cell has only 52 bytes instead of the usual 53.

(\*\*\*) If override counters on the HSMDA are configured (see the 7750 SR Quality of Service Guide).

(\*\*\*\*) Not used to identify stats from HSMDA due to MDA architecture. If the statistics are from HSMDA: apo, aoo else lpo/hpo, loo/hoo.

Table 45, Table 46, and Table 47 provide field descriptions.

**Table 45: Policer Stats Field Descriptions**

Field	Field Description
pid	PolicerId
statmode	PolicerStatMode
aod	AllOctetsDropped
aof	AllOctetsForwarded
aoo	AllOctetsOffered
apd	AllPacketsDropped
apf	AllPacketsForwarded
apo	AllPacketsOffered
hod	HighPriorityOctetsDropped
hof	HighPriorityOctetsForwarded
hoo	HighPriorityOctetsOffered
hpd	HighPriorityPacketsDropped
hpf	HighPriorityPacketsForwarded
hpo	HighPriorityPacketsOffered
iod	InProfileOctetsDropped
iof	InProfileOctetsForwarded
ioo	InProfileOctetsOffered
ipd	InProfilePacketsDropped
ipf	InProfilePacketsForwarded
ipo	InProfilePacketsOffered
lod	LowPriorityOctetsDropped
lof	LowPriorityOctetsForwarded
loo	LowPriorityOctetsOffered
lpd	LowPriorityPacketsDropped
lpf	LowPriorityPacketsForwarded
lpo	LowPriorityPacketsOffered
opd	OutOfProfilePacketsDropped
opf	OutOfProfilePacketsForwarded

**Table 45: Policer Stats Field Descriptions (Continued)**

Field	Field Description
opo	OutOfProfilePacketsOffered
ood	OutOfProfileOctetsDropped
oof	OutOfProfileOctetsForwarded
ooo	OutOfProfileOctetsOffered
uco	UncoloredOctetsOffered
v4po	IPv4PktsOffered *
v4oo	IPv4OctetsOffered *
v6po	IPv6PktsOffered *
v6oo	IPv6OctetsOffered *
v4pf	IPv4PktsForwarded *
v6pf	IPv6PktsForwarded *
v4pd	IPv4PktsDropped *
v6pd	IPv6PktsDropped *
v4of	IPv4OctetsForwarded *
v6of	IPv6OctetsForwarded *
v4od	IPv4OctetsDropped *
v6od	IPv6OctetsDropped *

\* Enhanced Subscriber Management (ESM) only.

**Table 46: Queue Group Record Types**

Record Name	Description
qgone	PortQueueGroupOctetsNetworkEgress
qgosi	PortQueueGroupOctetsServiceIngress
qgose	PortQueueGroupOctetsServiceEgress
qgpne	PortQueueGroupPacketsNetworkEgress
qgpsi	PortQueueGroupPacketsServiceIngress
qgpse	PortQueueGroupPacketsServiceEgress
fpqgosi	ForwardingPlaneQueueGroupOctetsServiceIngress
fpqgoni	ForwardingPlaneQueueGroupOctetsNetworkIngress
fpqgpsi	ForwardingPlaneQueueGroupPacketsServiceIngress
fpqgpni	ForwardingPlaneQueueGroupPacketsNetworkIngress

**Table 47: Queue Group Record Type Fields**

Field	Field Description
data port	Port (used for port based Queue Groups)
member-port	LAGMemberPort (used for port based Queue Groups)
data slot	Slot (used for Forwarding Plane based Queue Groups)

**Table 47: Queue Group Record Type Fields (Continued)**

<b>Field</b>	<b>Field Description</b>
forwarding-plane	ForwardingPlane (used for Forwarding Plane based Queue Groups)
queue-group	QueueGroupName
instance	QueueGroupInstance
qid	QueueId
pid	PolicerId
statmode	PolicerStatMode
aod...ucp	same as above

## Accounting Files

When a policy has been created and applied to a service or network port, the accounting file is stored on the compact flash in a compressed XML file format. The router creates two directories on the compact flash to store the files. The following output displays a directory named **act-collect** that holds accounting files that are open and actively collecting statistics. The directory named **act** stores the files that have been closed and are awaiting retrieval.

```
ALA-1>file cfl:\# dir act*
12/19/2006 06:08a      <DIR>          act-collect
12/19/2006 06:08a      <DIR>          act

ALA-1>file cfl:\act-collect\ # dir
Directory of cfl:\act-collect#

12/23/2006 01:46a      <DIR>          .
12/23/2006 12:47a      <DIR>          ..
12/23/2006 01:46a      112 act1111-20031223-014658.xml.gz
12/23/2006 01:38a      197 act1212-20031223-013800.xml.gz
```

Accounting files always have the prefix **act** followed by the accounting policy ID, log ID and timestamp. The accounting log file naming and log file destination properties like rollover and retention are discussed in more detail in [Log Files on page 383](#).

## Design Considerations

The router has ample resources to support large scale accounting policy deployments. When preparing for an accounting policy deployment, verify that data collection, file rollover, and file retention intervals are properly tuned for the amount of statistics to be collected.

If the accounting policy collection interval is too brief there may be insufficient time to store the data from all the services within the specified interval. If that is the case, some records may be lost or incomplete. Interval time, record types, and number of services using an accounting policy are all factors that should be considered when implementing accounting policies.

The rollover and retention intervals on the log files and the frequency of file retrieval must also be considered when designing accounting policy deployments. The amount of data stored depends on the type of record collected, the number of services that are collecting statistics, and the collection interval that is used. For example, with a 1GB CF and using the default collection interval, the system is expected to hold 48 hours worth of billing information.

## Reporting and Time-Based Accounting

Node support for volume and time-based accounting concept provides an extra level of intelligence at the network element level in order to provide service models such as “prepaid access” in a scalable manner. This means that the network element gathers and stores per-subscriber accounting information and compare it with “pre-defined” quotas. Once a quota is exceeded, the pre-defined action (such as re-direction to a web portal or disconnect) is applied.

## Overhead Reduction in Accounting: Custom Record

---

### User Configurable Records

Users can define a collection of fields that make up a record. These records can be assigned to an accounting policy. These are user-defined records rather than being limited to pre-defined record types. The operator can select what queues and the counters within these queues that need to be collected. Refer to the predefined records containing a given field for XML field name of a custom record field.

---

### Changed Statistics Only

A record is only generated if a significant change has occurred to the fields being written in a given the record. This capability applies to both ingress and egress records regardless on the method of delivery (such as RADIUS and XML). The capability also applies to Application Assurance records; however without an ability to specify different significant change values and per-field scope (for example, all fields of a custom record are collected if any activity was reported against any of the statistics that are part of the custom record).

## Configurable Accounting Records

- [XML Accounting Files for Service and ESM-Based Accounting on page 418](#)
  - [RADIUS Accounting in Networks Using ESM on page 418](#)
- 

### XML Accounting Files for Service and ESM-Based Accounting

The `custom-record` command in the `config>log>accounting-policy` context provide the flexibility to reduce the volume of data generated, network operators can define the record that needs to be collected. This can eliminate queues or selected counters within these queues that are not relevant for billing.

Record headers including information such as service-ID, SAP-ID, etc., will always be generated.

---

### RADIUS Accounting in Networks Using ESM

The `custom-record` command in the `config>subscr-mgmt>radius-accounting-policy` context provide the flexibility to include individual counters in RADIUS accounting messages. See the CLI tree for commands and syntax.

---

### Significant Change Only Reporting

Another way to decrease accounting messaging related to overhead is to include only “active” objects in a periodical reporting. An “active object” in this context is an object which has seen a “significant” change in corresponding counters. A significant change is defined in terms of a cumulative value (the sum of all reference counters).

This concept is applicable to all methods used for gathering accounting information, such as an XML file and RADIUS, as well as to all applications using accounting, such as service-acct, ESM-acct, and Application Assurance.

Accounting records are reported at the periodical intervals. This periodic reporting is extended with an internal filter which omits periodical updates for objects whose counter change experienced lower changes than a defined (configurable) threshold.

Specific to RADIUS accounting the `significant-change` command does not affect ACCT-STOP messages. ACCT-STOP messages will be always sent, regardless the amount of change of the corresponding host.



For Application Assurance records, a significant change of 1 in any field of a customized record (send a record if any field changed) is supported. When configured, if any statistic field records activity, an accounting record containing all fields will be collected.

## Immediate Completion of Records

---

### Record Completion for XML Accounting

For ESM RADIUS accounting, an accounting stop message is sent when:

- A subscriber/subscriber-host is deleted.
- An SLA profile instance (non-HSMDA) or subscriber instance (HSMDA) is changed.

A similar concept is also used for XML accounting. In case the accounted object is deleted or changed, the latest information will be written in the XML file with a “final” tag indication in the record header.

---

### AA Accounting per Forwarding Class

This feature allows the operator to report on protocol/application/app-group volume usage per forwarding class by adding a bitmap information representing the observed FC in the XML accounting files. In case the accounted object is deleted or changed, the latest information will be written in the XML file with a “final” tag indication in the record header.

## Configuration Notes

This section describes logging configuration caveats.

---

- A file or filter cannot be deleted if it has been applied to a log.
- File IDs, syslog IDs, or SNMP trap groups must be configured before they can be applied to a log ID.
- A file ID can only be assigned to *either* one log ID *or* one accounting policy.
- Accounting policies must be configured in the **config>log** context before they can be applied to a service SAP or service interface, or applied to a network port.
- The **snmp-trap-id** must be the same as the **log-id**.

