Serviceability, Logging, Monitoring, and Metrics

This section describes serviceability information about Expressway, including logging, system monitoring, metrics collection, and email notifications. For information about the optional Dedicated Management Interface (DMI) to use LAN3 for management traffic, see Configuring the Dedicated Management Interface (DMI).

Diagnostic and debugging tools, network testing utilities, and incident reporting are covered in Diagnostics and Troubleshooting.

- Configure Logging, on page 1
- Capture Call Detail Records, on page 6
- Configure Alarm-Based Email Notifications, on page 10
- System Metrics Collection, on page 13

Configure Logging

Expressway provides syslogging features for troubleshooting and auditing purposes. The Event Log is a rotating local log that records information about things like calls, registrations, and messages sent and received.

To configure Expressway logging options, go to **Maintenance** > **Logging**. From the **Logging** page you can do the following tasks:

- Specify the Change the Event Log Verbosity to change the depth of event information recorded locally
- Toggle Media Statistics Logging for Calls
- Toggle Capture Call Detail Records
- Toggle Certificate-Compliant Logging
- Define one or more Publishing Logs to Remote Syslog Servers addresses
- Filter by severity the events sent to each remote syslog server
- Toggle How to Configure System Metrics collection (collected)

Change the Event Log Verbosity

You can optionally control the local log verbosity by setting the **Local event log verbosity** between 1 and 4. All events have an associated level in the range 1-4, with Level 1 Events considered the most important.

Note Logging at level 3 or level 4 is not recommended for normal operation, because such detailed logging may cause the 2GB log to rotate too quickly. However, you may need to record this level of detail for troubleshooting.

Events are always logged locally - to the Event Log - regardless of whether or not remote logging is enabled.

The table gives an overview of the levels assigned to different events:

Level	Assigned events
1	High-level events such as registration requests and call attempts. Easily human readable. For example: • call attempt/connected/disconnected • registration attempt accepted/rejected
2	All Level 1 events, plus: logs of protocol messages sent and received (SIP, H.323, LDAP and so on) excluding noisy messages such as H.460.18 keepalives and H.245 video fast-updates
3	All Level 1 and Level 2 events, plus: • protocol keepalives • call-related SIP signaling messages
4	The most verbose level: all Level 1, Level 2 and Level 3 events, plus: • network level SIP messages

Changes to the log level affect both the Event Log that you view through the web interface, and the information that is copied to any remote log server. Changes are not retrospective and only affect what is logged after the change.

Expressway uses the following facilities for local logging. The software components / logs that map to the (local) facilities are emphasized:

- 0 (kern)
- 3 (daemon)
- 16 (local0) *Administrator*
- 17 (local1) *Config*
- 18 (local2) *Mediastats*
- 19 (local3) *Apache error*
- 20 (local4) *etc/opt/apache2*
- 21 (local5) *Developer*
- 22 (local6) *Network*

The Events and levels section has a complete list of all events that are logged by the Expressway, and the level at which they are logged.

Certificate-Compliant Logging

In some environments you may want to ensure that the Expressway logs are compliant with the requirements of your security certification. There is a trade-off between security and the purpose of the logs for diagnostics, and in the certification-compliant modes it may be impossible to establish the exact cause of a problem call.

How to Configure Certification-compliant Logging

Step 1 Go to **Maintenance** > **Logging**.

Step 2 In the **Logging options** section, set the **Certification logging** mode to one of the following:

Certification logging mode	Description
Diagnostic	This mode is not certification-compliant, but is most useful for diagnosing call issues.
Secretive	This mode is certification-compliant.
Secretive and Verbose	This mode is also certification-compliant, but enables you to collect some log information using a secure connection to a syslog server. These logs are not particularly useful in the diagnostic sense.

Publishing Logs to Remote Syslog Servers

Syslog is a convenient way to aggregate log messages from multiple systems to a single location. This is particularly recommended for peers in a cluster.

- You can configure the Expressway to publish log messages to up to 4 remote syslog servers.
- The syslog servers must support one of the following standard protocols:
  - BSD (as defined in RFC 3164)
  - IETF (as defined in RFC 5424)

Configuring Remote Syslog Servers

Note

- The **Filter by Keywords** option is applied to messages already filtered by severity.
- You can use up to five keywords, which includes groups of words (for example “login successful”), separated by commas.
- You can use a maximum of 256 characters in the keyword search.
- We recommend that you search for the most relevant keywords first to avoid any impact on system performance. This ensures the system pushes the relevant log messages to the syslog server at the earliest opportunity.

Step 1 Go to **Maintenance** > **Logging**, and enter the IP addresses or Fully Qualified Domain Names (FQDNs) of the **Remote syslog servers** to which this system will send log messages.

Step 2 Click on the **Options** button for each server.

Step 3 Specify the **Transport** protocol and **Port** you wish to use. If you choose to use TLS, you will see the option to enable Certificate Revocation List (CRL) checking for the syslog server.

Step 4 In the **Message Format** field, select the writing format for remote syslog messages. The default is *Legacy BSD*.

Step 5 Use the **Filter by Severity** option to select how much detail to send. The Expressway sends messages of the selected severity and all of the more severe messages.

Step 6 Use the **Filter by Keywords** option if you only want to send messages with certain keywords.

Step 7 Click **Save**.

Typical Values Used

The following table should help you select the format that best matches your logging server(s) and network configuration and shows the typical values used.

Table 1: Syslog message formats

Message format	Transport protocol	Suggested port	RFC
Legacy BSD format	UDP	514	BSD format. See RFC 3164
IETF syslog format	UDP	514	IETF format. See RFC 5424
IETF syslog using TLS connection	TLS	6514	IETF format. See RFC 5424

Note

- The UDP protocol is stateless. If reliability of syslog messages is very important in your environment, you should use a different transport protocol.
- If there is a firewall between the Expressway and the syslog server, you must open the appropriate port to allow the messages through.
- If you select TLS transport, the Expressway must trust the syslog server's certificate. Upload the syslog server's CA certificate to the local trust store if necessary.
- CRL checking when using TLS is disabled by default. To enable CRL, set **CRL checking** to *On* and ensure that relevant certificate revocation lists (CRLs) are loaded.

  See Security Basics for more information.
- The remote server cannot be another Expressway.
- An Expressway cannot act as a remote log server for other systems.
- The Expressway uses the following facilities for remote logging. The software components / logs that map to the (local) facilities are emphasised:
  - 0 (kern)
  - 3 (daemon)
  - 16 (local0) *Administrator*
  - 17 (local1) *Config*
  - 18 (local2) *Mediastats*
  - 19 (local3) *Apache error*
  - 20 (local4) *etc/opt/apache2*
  - 21 (local5) *Developer*
  - 22 (local6) *Network*

Media Statistics Logging for Calls

How to Enable Media Statistics

To optionally enable media statistics collection on the Expressway, go to **Maintenance** > **Logging** and set **Media statistics** to *On*. The system starts logging media statistics for each call, to the local hard disk in **/mnt/harddisk/log**. Up to 200 files of 10MB each are stored, and the oldest is deleted when file 200 is full.

The media statistics collected include packets forwarded, packets lost, jitter, media type, codec, and actual bitrate.

Media statistics are also published as syslog messages. While Media statistics logging is on, the Expressway publishes statistics using facility 18 (local2) to all remote syslog servers you have configured. The message severity is *Informational* but the media statistics messages are published irrespective of severity filter settings.

Capture Call Detail Records

Subject to enabling the service (which is off by default) Expressway can optionally capture CDRs. The CDRs are stored locally for seven days, and, if you use remote logging, can also be published as syslog messages.

How to Configure CDRs

To configure CDRs on Expressway:

Step 1 Go to **Maintenance** > **Logging**.

Step 2 In the **Logging Options** section, set the **Call Detail Records** field to the required option:

- *Services and Logging* - The CDRs are stored locally for 7 days and then deleted. The records are accessible from the local Event Log, and are also sent as INFO messages to your syslog host if external logging is enabled.
- *Service Only* - The CDRs are stored locally for 7 days and then deleted. The records are not accessible through the web user interface. The CDRs can only be read via the REST API.
- *Off* - CDRs are not logged locally. This is the default setting.

CDR Properties

This table defines the properties that are visible in CDRs:

Field	Definition
uuid	ID of the CDR entry.
service_uuid	ID used to identify whether a record is from a proxy, Lync B2BUA or Encryption B2BUA.
active	Whether a call is a live or a historical one.
initial_call	Used internally to tie to a B2BUA call when it is a multiple-component one (involves a B2BUA hop).
licensed	Shows if a call used a license.
licensed_as_traversal	Shows if a call used a traversal license.
status	200 OK message indicates a call was successful. Contains an error message if the call was unsuccessful.
tag	Call ID.
box_call_serial_number	Extra ID added to tie multiple calls together (for example, through the B2BUA).

Field	Definition
start_time	Date and time of the call. Time zone can be set in **System** > **Times** > **Time Zone** and the date format is YYYY-MM-DD.
end_time	End time of the call.
source_alias	Alias of the caller.
destination_alias	Alias of the callee.
aside_destination_alias	Alias of the caller (or MS Lync client if Lync Interop).
bside_destination_alias	Alias of the callee (or non-Lync client).
aside_request_uri	Request uri of the caller (or MS Lync client if Lync Interop).
bside_request_uri	Request uri of the callee (or non-Lync client).
protocol	Shows if the call was SIP <-> SIP, SIP <-> H323, H323 <-> SIP, or H323 <-> H323.
protocol_summary	As above but can have extra info like if a call was multi-component, DVO, etc.
media_routed	Shows if media was sent during the call (e.g. NAT/IWF/B2BUA).
audio	Shows if the call was an audio-only one.
traversal_license_tokens	Indicates if a call fork/branch took media (audio equates to 1 token and video 2).*
non_traversal_license_tokens	Indicates if a call fork/branch did not need to take media (audio equates to 1 token and video 2).*
disconnect_reason	Gives reasons for a call drop such as normal call teardown or other errors (that is, the last status).
details	Gives more details of the call, including media statistics.
last_updated_timestamp	Last time that any of the above fields were updated.

* Once a call is set up only one of these entries will have a non-zero value (i.e. only for the answered fork/branch).

APIs to Access CDRs

You can use the following secure REST APIs to gather CDRs:

- get_all_records (returns all records up to seven days old).
- get_records_for_interval (returns records from during the time specified).
- get_records_for_filter (filters results using any combination).
- get_all_csv_records (returns all records up to seven days old in csv format).

Important The call history is stored locally for seven days only, and is deleted automatically.

To access the desired API use the following URL:

https://%3CExpressway_IP%3E/api/external/callusage/%3CAPI%3E

API examples

- http://%3CExpressway_IP%3E/api/external/callusage/get_all_records
- `http://<Expressway_IP>/api/external/callusage/get_records_for_interval?fromtime=<fromtime>&totime=<to_time>`

  for example,

  ```
  https://203.0.113.17/api/external/callusage/
  get_records_for_interval?fromtime=2014-05-09 2000:00:00&totime=
  2014-05-10 2000:00:00
  ```

Input Parameters

Parameter	Description
fromtime	Mandatory. The start time from which the CDR records are required. **Format**: *YYYY-MM-DD HH:MI:SS*
totime	Mandatory. The end time from which the CDR records are required. **Format**: *YYYY-MM-DD HH:MI:SS*

- `http://<Expressway_IP>/api/external/callusage/get_records_for_interval?fromtime=<fromtime>&totime=<to_time>`

  for example,

  ```
  https://203.0.113.17/api/external/callusage/
  get_records_for_interval?fromtime=2014-05-09 2000:00:00&totime=
  2014-05-10 2000:00:00
  ```

- `http://<Expressway_IP>/api/external/callusage/get_records_for_filter?uuid=<uuid>&src_alias=<src_alias>&dest_alias=<dest_alias>&protocol=<protocol>`

  for example,

  ```
  https://203.0.113.17/api/external/callusage/
  get_records_for_filter?uuid=6e3b5a8a-346c-421b-aa2e-f4409c43a81a
  &src_alias=TC149-057-h323@domain.com&dest_alias=
  TC149-065-h323@domain.com&protocol=H323 <-> H323
  ```

Input Parameters

Parameter	Description
uuid	Unique identifier of the record.
src_alias	Origin point of the call.

Parameter	Description
dest_ alias	Destination point of the call.
protocol	Protocol that was used for the call (SIP, H323 etc).

- http://%3CExpressway_IP%3E/api/external/callusage/get_all_csv_records

CDR Examples

Sample CDR

This sample CDR applies to all APIs except csv:

[{"initial_call": "false", "protocol": "SIP <-> SIP", "protocol_summary": "", "disconnect_reason": "200 OK", "licensed": "false", "tag": "b8d52a60-16a1-4bdb-be93-f5a675408811", "aside_request_uri": "", "box_call_serial_number": "22cd0e7d-c498-4068-9239-624038fe5130", "source_alias": "sip:10000005@10.196.4.82", "uuid": "800fe013-83f4-4094-a5e6-e2f9489912e2", "last_updated_timestamp": 1444725389, "details": "{\"Call\":{\"SerialNumber\":
\"800fe013-83f4-4094-a5e6-e2f9489912e2\",\"BoxSerialNumber\":
\"22cd0e7d-c498-4068-9239-624038fe5130\",\"Tag\": \"b8d52a60-16a1-4bdb-be93-f5a675408811\",\"State\":
\"Disconnected\",\"StartTime\": \"2015-10-13 01:36:26.485636\",\"InitialCall\": \"False\",\"Licensed\":
\"False\",\"LicensedAsTraversal\": \"False\",\"SourceAlias\":
\"sip:10000005@10.196.4.82\",\"DestinationAlias\": \"sip:10000010@cucm-82\",\"ToLocalB2BUA\":
\"False\",\"Audio\": \"False\",\"License\":{\"Traversal\": \"0\",\"NonTraversal\": \"0\",\"DemotedTraversal\":
\"0\",\"CollaborationEdge\": \"0\",\"Cloud\": \"0\"},\"Duration\": \"3\",\"Legs\":[{\"Leg\":{\"Protocol\":
\"SIP\",\"SIP\":{\"Address\": \"10.196.4.61:5073\",\"Transport\": \"TLS\",\"Aliases\":[{\"Alias\":{\"Type\":
\"Url\",\"Origin\": \"Unknown\",\"Value\":
\"sip:10000005@10.196.4.82\"}}]},\"Targets\":[{\"Target\":{\"Type\": \"Url\",\"Origin\":
\"Unknown\",\"Value\": \"sip:10000010@10.196.4.116\"}}],\"BandwidthNode\":
\"DefaultZone\",\"EncryptionType\": \"AES\",\"Cause\": \"200\",\"Reason\": \"OK\"}},{\"Leg\":{\"Protocol\":
\"SIP\",\"SIP\":{\"Address\": \"10.196.4.71:7001\",\"Transport\": \"TLS\",\"Aliases\":[{\"Alias\":{\"Type\":
\"Url\",\"Origin\": \"Unknown\",\"Value\":
\"sip:10000010@cucm-82\"}}]},\"Source\":{\"Aliases\":[{\"Alias\":{\"Type\": \"Url\",\"Origin\":
\"Unknown\",\"Value\": \"10000005@10.196.4.82\"}}]},\"BandwidthNode\":
\"Traversal-zone\",\"EncryptionType\": \"AES\",\"Cause\": \"200\",\"Reason\":
\"OK\"}}],\"Sessions\":[{\"Session\":{\"Status\": \"Completed\",\"MediaRouted\": \"False\",\"CallRouted\":
\"True\",\"Participants\":{\"Leg\": \"1\",\"Leg\": \"2\",\"Incoming\":{\"Leg\": \"1\"},\"Outgoing\":{\"Leg\":
\"2\"}}}}],\"EndTime\": \"2015-10-13 01:36:29.745651\"}}", "status": "Disconnected", "destination_alias":
"sip:10000010@cucm-82", "licensed_as_traversal": "false", "service_uuid":
"e6723fd0-5ca2-11e1-b86c-0800200c9a66", "start_time": "2015-10-13 01:36:26.485636",
"traversal_license_tokens": 0, "bside_destination_alias": "", "active": "false", "media_routed": "false",
"aside_destination_alias": "", "non_traversal_license_tokens": 0, "bside_request_uri": "", "end_time":
"2015-10-13 01:36:29.745651", "audio": "false”}]

Sample csv CDR

```
uuid,service_uuid,active,initial_call,licensed,licensed_as_traversal,
status,tag,box_call_serial_number,start_time,end_time,source_alias,
destination_alias,aside_destination_alias,bside_destination_alias,
aside_request_uri,bside_request_uri,protocol_summary,protocol,
```

```
media_routed,audio,traversal_license_tokens,non_traversal_license_tokens,
disconnect_reason,details,last_updated_timestamp

800fe013-83f4-4094-a5e6-e2f9489912e2,e6723fd0-5ca2-11e1-
b86c-0800200c9a66,false,false,false,false,Disconnected,b8d52a60-16a1-
4bdb-be93-f5a675408811,22cd0e7d-c498-4068-9239-624038fe5130,2015-10-
13 01:36:26.485636,2015-10-13
```

01:36:26.485636,2015-10-13 01:36:29.745651,sip:10000005@10.196.4.82,sip:10000010@cucm-82,,,,,,SIP <-> SIP,false,false,0,0,200 OK,"{""Call"":{""SerialNumber"": ""800fe013-83f4-4094-a5e6-e2f9489912e2"",""BoxSerialNumber"": ""22cd0e7d-c498-4068-9239-624038fe5130"",""Tag"": ""b8d52a60-16a1-4bdb-be93-f5a675408811"",""State"": ""Disconnected"",""StartTime"": ""2015-10-13 01:36:26.485636"",""InitialCall"": ""False"",""Licensed"": ""False"",""LicensedAsTraversal"": ""False"",""SourceAlias"": ""sip:10000005@10.196.4.82"",""DestinationAlias"": ""sip:10000010@cucm-82"",""ToLocalB2BUA"": ""False"",""Audio"": ""False"",""License"":{""Traversal"": ""0"",""NonTraversal"": ""0"",""DemotedTraversal"": ""0"",""CollaborationEdge"": ""0"",""Cloud"": ""0""},""Duration"": ""3"",""Legs"":[{""Leg"":{""Protocol"": ""SIP"",""SIP"":{""Address"": ""10.196.4.61:5073"",""Transport"": ""TLS"",""Aliases"":[{""Alias"":{""Type"": ""Url"",""Origin"": ""Unknown"",""Value"": ""sip:10000005@10.196.4.82""}}]},""Targets"":[{""Target"":{""Type"": ""Url"",""Origin"": ""Unknown"",""Value"": ""sip:10000010@10.196.4.116""}}],""BandwidthNode"": ""DefaultZone"",""EncryptionType"": ""AES"",""Cause"": ""200"",""Reason"": ""OK""}},{""Leg"":{""Protocol"": ""SIP"",""SIP"":{""Address"": ""10.196.4.71:7001"",""Transport"": ""TLS"",""Aliases"":[{""Alias"":{""Type"": ""Url"",""Origin"": ""Unknown"",""Value"": ""sip:10000010@cucm-82""}}]},""Source"":{""Aliases"":[{""Alias"":{""Type"": ""Url"",""Origin"": ""Unknown"",""Value"": ""10000005@10.196.4.82""}}]},""BandwidthNode"": ""Traversal-zone"",""EncryptionType"": ""AES"",""Cause"": ""200"",""Reason"": ""OK""}}],""Sessions"":[{""Session"":{""Status"": ""Completed"",""MediaRouted"": ""False"",""CallRouted"": ""True"",""Participants"":{""Leg"": ""1"",""Leg"": ""2"",""Incoming"":{""Leg"": ""1""},""Outgoing"":{""Leg"": ""2""}}}}],""EndTime"": ""2015-10-13 01:36:29.745651""}}",1444725389

Configure Alarm-Based Email Notifications

Expressway supports email-based notifications based on alarm severity and optionally by alarm ID. If configured, when an alarm is generated in the system an email notification is sent to the configured destination address. For each alarm severity classification you can define a different email ID, to differentiate the urgency of the notification. Multiple email IDs can be configured for alarms of the same severity.

From X12.6.2 you can also direct notifications for a specific alarm ID to a particular email ID, or disable notifications for a specific alarm ID.

Important The maximum permitted length for email IDs is 256 characters.

This functionality is also available to U.S.-based customers who want to implement Kari's Law. When a 9-1-1 call is made which meets the criteria for direct 9-1-1 dialing through Expressway, an alarm of severity *Emergency* is generated, and (if configured) a notification will be sent to the email ID configured for the alarm severity *Emergency*.

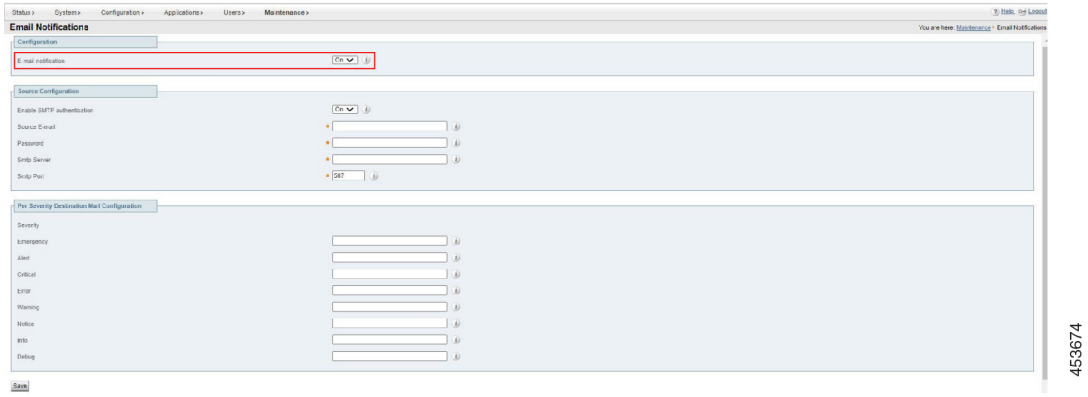
Before You Begin

- You will need to provide SMTP server details to establish the connection for sending the email.
- Expressway only supports the TLS connection with the SMTP server.
- The SMTP server must be reachable from Expressway either directly or using an SMTP proxy. Using an HTTP proxy for SMTP is not supported.
- The source email and password are validated in the SMTP server before sending the mail.

Process to Configure Alarm-Based Email Notifications

Step 1 Go to **Maintenance** > **Email Notifications**.

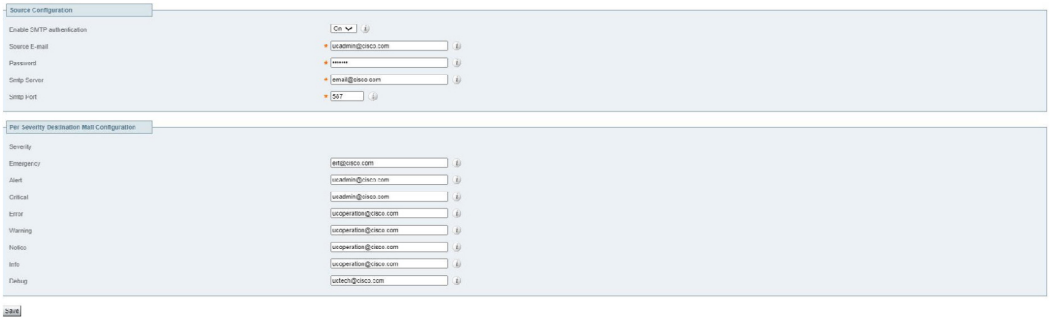
Step 2 In the **Email notification** drop-down list, select *On*.



Step 3 In the **Source Configuration** section, enter the following information:

- **Enable SMTP authentication** - The SMTP server authentication is enabled (in *On* state) by default. The drop-down list allows you to toggle between *On* and *Off* state. It should be turned *Off* only if the User's SMTP server doesn't require authentication. Most of the Users require it.
- Source email address from which notifications are sent to the configured destination address.
- IP address or FQDN of the SMTP server to be used to send the notifications.
- In the **Per Severity Destination Mail Configuration** section, enter the email address that you want to receive notifications for alarms of a given severity.
- Click **Save**.

Figure 1: Example configuration for email notifications



How to Customize Notifications - Disable or Send to an Email Address

Optionally use this process to send notifications for a given alarm ID to a specific email address, or to disable notifications for a given alarm ID altogether. For example to send threshold warning alarms to a designated individual, or to stop notifications from an unwanted alarm.

Step 1 Go to the **Custom notifications** section of the **Email notifications** page. Here you can view, edit, and delete existing custom notifications, and add new ones.

Step 2 To create a customized notification:

a. Click **Add**.

b. Select the alarm ID you want to work with.

c. In the **Notification** drop-down select *Custom* to define an email destination for the selected alarm, or select *Disable* if you don't want emails to be sent for this alarm.

d. If you selected Custom, in the **Email** field type the destination email address to which the selected alarm notifications are to be sent.

e. Click **Save**.

f. To test that the alarm notifications work as you intended:

1. Select the alarm to test from the **Select Alarm** dropdown.
2. Click the **Test Now** button.
3. Check that the email notification(s) received from the test are as expected.

System Metrics Collection

System Metrics Collection is a feature on Expressway that publishes system performance statistics, to allow remote monitoring of performance. Expressway collects statistics about the performance of the hardware, OS, and the application, and publishes these statistics to a remote host (typically a data analytics server) that aggregates the data. You can configure this feature on Expressway through the web interface or the command line.

Note Configuration from one peer applies throughout the cluster, so if you are monitoring a cluster we recommend configuring System Metrics Collection on the primary peer.

Configuration is also required on the remote server. The collectd daemon must be running on the server, with the collectd network plugin configured to listen on an address that can be seen by the clients. The configuration details depend on your monitoring environment and are beyond the scope of this guide.

How to use the collected data

You can use tools such as Circonus and Graphite to generate graphs and aggregate statistics, and to analyze performance, based on the data collected from Expressway. You can also use it to visualize trends and even to predict potential issues. Metrics that you can visualize include:

- Active calls per zone and by system
- Key process metrics: System CPU, user CPU, and memory usage of key processes
- Alarms

How to Configure System Metrics collection (collected)

Configure on Expressway

Use this procedure to optionally configure Expressway from the web user interface, to collect statistics and publish them to a specified server.

Step 1 Log on to the Expressway and go to **Maintenance** > **Logging**.

Step 2 Toggle **System Metrics Collection** to *On*.

Step 3 Enter the **`Collection server address`**.

You can use IP address, hostname, or FQDN to identify the remote server.

Step 4 Change the default **Collection server port** (the listening port) if necessary - if the collection server is listening on a non-default port.

Step 5 Change the default **Collection Interval** if necessary - if your policy requires finer-grained metrics than the default interval of 60 seconds.

Step 6 Click **Save**.

Examples of the CLI commands to configure collectd

If you prefer to use the CLI, these are examples of the relevant commands:

Table 2: CLI commands to configure collectd

What the command does	Example command
Toggle Metrics Collection on/off	**xconfig log SystemMetrics mode: on**
Specify the server address	**xconfig log SystemMetrics network address: address**
Specify the listening port	**xconfig log SystemMetrics network port: 25826**
Specify the collection interval	**xconfig log SystemMetrics interval: 60**
Read System Metrics configuration	**xstatus SystemMetrics**

Configure a Remote Server

Selection and configuration of the server you use for data analytics in your environment is beyond the scope of this document. Circonus and Graphite are examples of applications that can handle collectd information. Your analytics tool must support receiving data from the collectd daemon. This daemon is running on the Expressway and pushes the metrics to your analytics server, using the collectd network plugin.

The network plugin implements the collectd binary protocol for data encapsulation. The analytics server must be able to parse and present this data. Your analytics server will probably have its own UI for configuring how it collects and shows the data, which could be based on collectd or an alternative software.

If you are using collectd on the analytics server, modify *collectd.conf* file so that the server:

- Listens for data from the collectd clients (such as Expressway). You need to enable the network plugin and configure the listen block with the server's IP address. For example:

```
<Plugin "network">
              Listen "198.51.100.15"
        </Plugin>
```

- Stores the data it receives in a human readable form (such as CSV files). You need to enable the csv plugin to tell it where to write the files. For example:

```
<Plugin "csv">
               DataDir "/var/lib/collectd/csv"
               StoreRates true
        </Plugin>
```

More information

- https://collectd.org/wiki/index.php/Networking_introduction
- https://collectd.org/documentation/manpages/collectd.conf.5.shtml#plugin_network
- https://collectd.org/wiki/index.php/Binary_protocol
- https://collectd.org/wiki/index.php/Plugin:CSV
- https://collectd.org/documentation/manpages/collectd.conf.5.shtml#plugin_csv

Troubleshooting

To check whether Expressway is sending data, configure a TCP dump from the Expressway and check for packets sent to the address of the data analytics server. Go to **Maintenance** > **Diagnostics** > **Diagnostics logging**, check **Take tcpdump while logging** and start logging.

Metrics Collected from Expressway

The following hardware statistics are monitored:

- aggregation-cpu-sum
- aggregation-cpu-average
- Per-core CPU usage for each core in the system
- df
- disk
- load
- protocols-Tcp
- protocols-Udp
- swap
- Users
- memory
- Uptime
- Process

The following application data is monitored by the custom exec-app plugin for collectd:

- `gauge-active_alarms` is the count of active alarms on this Expressway
- `gauge-active_calls` is the count of calls being handled by this Expressway
- `gauge-<service name>` is the status of each system service.
- `gauge-<zone name>_ActiveCalls` counts the active calls in the named zone
- `gauge-<zone name>_BandwidthAllocated` measures the total bandwidth allocated to the named zone
- `gauge-<zone name>_BandwidthLimit`

Each of these metrics uses the collectd GAUGE data source type, which allows free-form data. On the collection server, the full collectd value name will be shown, for example *collectdHostnamecollectd.exec-app.gauge-active_calls*.

Note Zone names are user-configurable and may thus be in conflict with the naming schema for collectd metrics. If your collection server is enforcing the schema, there is a chance that metrics from some zones will not be accepted.

Data that's sent to the collection server

The network plugin uses the collectd binary protocol to encapsulate numeric, string, and value data representing the monitored hardware resources and software processes. The plugin pushes the metrics data packets to the analytics server once every interval, using UDP 25826 by default. The analytics server parses and presents the data in human readable form.

If the analytics server is using the collectd network plugin and csv plugin, then the metrics are stored as small CSV files, using the metric name and timestamp to create the filename. For example, *gauge-H323-2015-05-21*

collectd plugins

These collectd plugins are implemented in Expressway:

Plugin name	Description
Aggregation	Aggregates CPU values into the counters `aggregation_cpu_sum and aggregation_cpu_average`.
CPU	Processor information. The raw information is `aggregated into aggregation_cpu_average and aggregation_cpu_sum`.
DF	File system information; see DF description on collectd Wiki.
Disk	Hard disk performance; see Disk description on collectd Wiki.

Plugin name	Description
Exec-app	Customized version of **exec** that returns specific Expressway information on calls, alarms, zones, and services. • gauge-active_alarms • gauge-active_calls • gauge-B2BUA • gauge-cafemanager • gauge-callusagemanager • gauge-<zone>_ActiveCalls • gauge-<zone>_BandwidthAllocated • gauge-c_mgmt • gauge-collectd • gauge-developer • gauge-edgeconfigprovisioning • gauge-fail2ban • gauge-findmed • gauge-forwardproxy • gauge-H323 • gauge-http • gauge-https • gauge-importcontrol • gauge-jabberd • gauge-LCDd • gauge-managementconnector • gauge-opends • gauge-phonebookserver • gauge-portforwarding • gauge-provisioningd • gauge-provisioningserver • gauge-proxy-registrationd

Plugin name	Description
Exec-app	• gauge-restmanager • gauge-samlverifier • gauge-singlesignon • gauge-SIP • gauge-snmpd • gauge-sshd • gauge-sshdpfwd • gauge-sslh • gauge-telnetd • gauge-trafficserver • gauge-transcodermanager • gauge-tty • gauge-winbindd
Load	System load based on task queue.
Memory	Memory statistics.
Network	Enables publishing to a remote address. The plugin implements the collectd binary protocol for data encapsulation. The remote server must have the appropriate parsing tool.
Protocols	Configurable subset of the protocols used by the Expressway.

Plugin name	Description
Process	Counts the system processes and groups them by state (such as running, sleeping, zombies). Also collects detailed statistics about specific processes. The plugin monitors the following processes in detail: • app • bramble • credentialmanagerservermain • cvs_main • erlang-beam • erlang-epmd • httpd • httpserver • ivy • licensemanagerservermain • managementconnectormain • managementframework • openssl2nss • policyservermain • sshdpfwd • syslog-ng • traffic_server • XCP

Plugin name	Description
Statsd	Customized version that returns specific Expressway information. For example, ICE usage. • gauge-ICEPassthroughMetrics.b2buacalls • gauge-ICEPassthroughMetrics.candidatesofferedmissingiceconfig • gauge-ICEPassthroughMetrics.failedicenegotiationcalls • gauge-ICEPassthroughMetrics.hosthostcalls • gauge-ICEPassthroughMetrics.hostrelaycalls • gauge-ICEPassthroughMetrics.hostsrvrflxcalls • gauge-ICEPassthroughMetrics.icecalls • gauge-ICEPassthroughMetrics.icecandidatecalls • gauge-ICEPassthroughMetrics.iceconfiguredcalls • gauge-ICEPassthroughMetrics.noicecandidatesoffered • gauge-ICEPassthroughMetrics.onepartyicecandidatecalls • gauge-ICEPassthroughMetrics.relayrelaycalls • gauge-ICEPassthroughMetrics.srvrflxrelaycalls • gauge-ICEPassthroughMetrics.srvrflxsrvrflxcalls
Swap	Amount of system memory written to disk.
Uptime	Tracks system uptime, providing counters like average running time or maximum uptime for a particular period; see Uptime description on collectd Wiki.
Users	Count of currently logged in users.