Live Data Reporting System

In Real Time data collection, reporting data writes to the Unified CCE Data Server and Unified Intelligence Center queries the data periodically. In contrast, Live Data continuously processes agent and call events from the peripheral gateway and the router, and publishes data directly to Unified Intelligence Center. Live Data continuously pushes only changed data to the reporting clients without the delay of writing to, and reading from the database. Individual state values, such as agent states, refresh as they happen, while other values, such as calls in queue, refresh approximately every 3 seconds.

The Live Data report templates take advantage of the Live Data service.

The Real Time data flow is still used to support other stock and custom reports.

Live Data is a stream processing system which aggregates and processes the events in-stream and publishes the information. Unified Intelligence Center subscribes to the message stream to receive the events in real-time and continuously update the Live Data report.

- Live Data Collecting Logs, on page 1
- Live Data Failover Configuration, on page 7
- Live Data Syslog, on page 9
- Monitor and Analyze System Performance Using Nmon, on page 10
- Live Data Socket.IO, on page 11
- Live Data SNMP, on page 12
- Live Data Collecting Logs, on page 23
- Live Data Syslog, on page 28
- Live Data Socket.IO, on page 29
- Live Data Collecting Logs, on page 30
- Live Data Failover Configuration, on page 36
- Live Data Syslog, on page 38
- Monitor and Analyze System Performance Using Nmon, on page 39
- Live Data Socket.IO, on page 40
- Live Data SNMP, on page 41

Live Data Collecting Logs

The logs that the Live Data services generate are available through the same tools as the Unified Intelligence Center logs.

For example, you can use the **file get** CLI command to collect all the Live Data logs:

```
file get activelog livedata/logs/*/*
```

You can also use the Real Time Monitoring Tool (RTMT) to collect and view logs and traces of the Live Data services.

Live Data Log Levels

Use the command-line interface to set trace level settings for Live Data services. There is no method in OAMP to set the log levels for the Live Data components.

You can use the **set live-data trace** command to set the log level or apply a tracemask to the following subsystems:

- Communications - logs messages related to connections
- DataProcessing - logs messages related to the processing of messages
- Database - logs messages specific to the database
- Event-store - logs messages specific to the storage of agent call-log and state-log events

Note You cannot apply a tracemask to the event-store subsystem.

Setting the loglevel

Required Minimum Privilege Level: Advanced

Command Syntax
set live-data trace *subsystem* **loglevel** *value*
subsystem

communications, dataprocessing, event-store or database

value

The loglevel for the specified subsystem:

- DEBUG
- INFO
- NOTICE
- WARN
- ERROR
- CRITICAL
- ALERT
- EMERGENCY

Note Only the following log levels are applicable to the event-store subsystem: DEBUG, INFO, WARN, and ERROR.

Example: `admin:set live-data trace dataprocessing loglevel DEBUG`

Setting the tracemask

Required Minimum Privilege Level: Advanced

You can set detailed log levels by enabling trace flags, which allows debug statements to appear in the logs. You can control debug tracing for specific functionalities by specifying a TRACE flag name within specific subsystem components. See Infrastructure Trace Definitions in the *Administration Console User Guide for Cisco Unified Intelligence Center* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-intelligence-center/products-maintenance-guides-list.html.

Command Syntax
set live-data trace *subsystem* **tracemask** *value*
subsystem

communications, dataprocessing, or database

value

For each of the three subsystems specify one of several tracemasks:

dataprocessing

- TIP_APPL_MESSAGE_TRACEMASK
- CAMEL_JMS_TRACEMASK
- STORM_SPOUT_TRACEMASK
- TIP_PROTOCOL_TRACEMASK
- FAILOVER_HB_TRACEMASK

database

- DB_UCCE_AW_TRACEMASK

communications

- JMS_COMMUNICATION_TRACEMASK
- FAILOVER_TOS_TRACEMASK

To set multiple tracemasks, separate them with a space.

Example: `admin: set live-data trace dataprocessing tracemask TIP_APPL_MESSAGE_TRACEMASK CAMEL_JMS_TRACEMASK`

To clear all tracemasks, use the tracemask command without parameters, for example:

```
set live-data trace dataprocessing tracemask
```

You can list the trace masks available for each subsystem using the command help option, for example:

```
set live-data trace dataprocessing tracemask ?
```

Show the loglevel

Required Minimum Privilege Level: Ordinary

To display the current loglevel or tracemask, use the `show live-data trace` command, for example:

```
admin:show live-data trace dataprocessing loglevel
```

```
admin:show live-data trace dataprocessing tracemask
```

Set Live-Data Trace Agent

Required Minimum Privilege Level: Advanced

Use this command to enable detailed tracing for specific agents.

Note This command requires in-depth system knowledge. Use detailed tracing only for advanced troubleshooting. Enabling detailed tracing adds many messages to the log files. The system may quickly exceed limits on the size and number of log files, causing the oldest log files to be overwritten. As a result, running detailed tracing shortens the history covered by the log files.

Command Syntax
set live-data trace agent *AgentSkillTargetIDs*
AgentSkillTargetIDs

The AgentSkillTargetIDs of the agents you want to trace. Separate IDs with a space. You can only have detailed tracing enabled for three agents simultaneously.

Note Running this command overwrites any previous setting.

Example: `admin:set live-data trace agent 5037 6000`

Output: `Enable detailed traces for agent(s) with the following id(s): 5037 6000`

Show Agents Currently Set

Required Minimum Privilege Level: Ordinary

This command shows the IDs of the agents that have detailed trace turned on.

show live-data trace agent

No parameters are required.

Example: `admin:show live-data trace agent`

Output: `Detailed traces are turned on for the agent(s) with the following id(s): 5037`

Unset Trace

Required Minimum Privilege Level: Advanced

This command turns off detailed traces for all agents.

unset live-data trace agent

No parameters are required.

Example: `admin:unset live-data trace agent`

Output: `Disable detailed traces for all agent(s)`

Help Command

unset live-data trace agent *?*

Example: `admin:set live-data trace agent ?`

Output: `This command is used to set the trace level for Agents.`

Set Live-Data Trace Skill-Group

Required Minimum Privilege Level: Advanced

Use this command to enable detailed tracing for specific skill-groups.

Note This command requires in-depth system knowledge. Use detailed tracing only for advanced troubleshooting. Enabling detailed tracing for a skill-group with many agents, for example, more than one hundred, adds a large number of messages to the log files. The system may quickly exceed limits on the size and number of log files, causing the oldest log files to be overwritten. As a result, running detailed tracing shortens the history covered by the log files.

Command Syntax
set live-data trace skill-group *Skill-GroupSkillTargetIDs*
Skill-GroupSkillTargetIDs

The Skill-GroupSkillTargetIDs of the skill-groups you want to trace. Separate IDs with a space. You can only have detailed tracing enabled for 3 skill-groups simultaneously.

Note Running this command overwrites any previous setting.

Example: `set live-data trace skill-group 5037 6000`

Output: `Enable detailed traces for skill-group(s) with the following id(s): 5037 6000`

Show Skill-Groups Currently Set

Required Minimum Privilege Level: Ordinary

show live-data trace skill-group

No parameters are required.

Example: `admin:show live-data trace skill-group`

`Output: Detailed traces are turned on for the skill-group(s) with the following id(s): 11962`

Unset Trace

Required Minimum Privilege Level: Advanced

This command turns off detailed tracing for all skill-groups.

unset live-data trace skill-group

No parameters are required.

Example: `admin:unset live-data trace skill-group`

Output: `Disable detailed traces for all skill-group(s)`

Help Command

unset live-data trace skill-group *?*

Example: `admin:set live-data trace skill-group ?`

Output: `This command is used to set the trace level for skill-groups.`

Set Live-Data Trace Precision-Queue

Required Minimum Privilege Level: Advanced

Use this command to enable tracing for specific precision-queues.

Note This command requires in-depth system knowledge. Use detailed tracing only for advanced troubleshooting. Enabling detailed tracing for a precision-queue adds many messages to the log files. The system may quickly exceed limits on the size and number of log files, causing the oldest log files to be overwritten. As a result, running detailed tracing shortens the history covered by the log files.

Command Syntax
set live-data trace precision-queue *Precision-QueueIDs*
Precision-QueueIDs

The Precision-QueueIDs of the precision-queues you want to trace. Separate IDs with a space. You can only have detailed tracing enabled for 3 precision-queues simultaneously.

Note Running this command overwrites any previous setting.

Example: `admin:set live-data trace precision-queue 5037 6000`

Output: `Enable detailed traces for precision-queue(s) with the following id(s): 5037 6000`

Show Precision-Queues Currently Set

Required Minimum Privilege Level: Ordinary

Show the IDs of the precision-queues that have detailed trace turned on.

show live-data trace precision-queue

No parameters are required.

Example: `admin:show live-data trace precision-queue`

Output: `Detailed traces are turned on for the precision-queue(s) with the following id(s): 5000`

Unset Trace

Required Minimum Privilege Level: Advanced

This command turns off detailed traces for all precision-queues.

unset live-data trace precision-queue

No parameters are required.

Example: `admin:unset live-data trace precision-queue`

Output: `Disable detailed traces for all precision-queue(s)`

Help Command

unset live-data trace precision-queue *?*

Example: `admin:set live-data trace precision-queue ?`

Output: `This command is used to set the trace level for precision-queues.`

Live Data Failover Configuration

The following Live Data failover commands are provided to allow you to monitor the Live Data failover mechanism during troubleshooting.

By default, Live Data failover is automatically configured during installation or upgrade. Under general operations you do not need to use these commands. Use **show live-data failover** to display information on the current configuration and state of Live Data reporting system cluster. Use **set live-data failover** to enable Live Data failover, and **unset live-data failover** to unset Live Data failover.

set live-data failover

Required Minimum Privilege Level: Advanced

Use this command to enable Live Data failover. This command automatically sets the system to run in duplex mode. Run this command on both Side A and Side B.

Command Syntax
set live-data failover

There are no required parameters.

unset live-data failover

Required Minimum Privilege Level: Advanced

Use this command to unset Live Data failover. This command automatically sets the system to run in simplex mode. Run this command on both Side A and Side B.

Command Syntax
unset live-data failover

There are no required parameters.

show live-data failover

Required Minimum Privilege Level: Ordinary

Use this command to display the Live Data cluster failover status and settings.

Command Syntax
show live-data failover
There are no parameters.

The command returns information on the Live Data server on which you run the command. The Live Data server information includes the current Live Data cluster settings, the status of the ActiveMQ connection, and the state of the cluster.

The possible cluster states are:

Cluster state	Description
PAIRED-ACTIVE	The cluster is in the active state and is communicating with the remote side.
PAIRED-STANDBY	The cluster is in the standby state and is communicating with the remote side.
ISOLATED-ACTIVE	The cluster is in the active state, but it is not communicating with the remote side.
ISOLATED-STANDBY	The cluster is in the standby state, but it is not communicating with the remote side.
SIMPLEXED-MODE	The cluster is working in simplex mode.
OUT-OF-SERVICE	The cluster is out of service.
CONNECTING	The cluster is attempting to do a handshake with the remote side.
TESTING	The cluster is unable to communicate with the remote side and is using the Test-Other-Side procedure to determine whether to become active or standby.

The console output after you run this command on the publisher side of a Live Data system is similar to the following:

```
admin:show live-data failover
# Failover settings..
Cluster failover enabled: true
Cluster ID: A
Remote side addr: not applicable for the publisher in auto-config
Auto config enabled: true

# ActiveMQ NetBridge..
Established

# Cluster state..
PAIRED-ACTIVE
```

Sample console output on the subscriber side is as follows:

```
admin:show live-data failover
# Failover settings..
Cluster failover enabled: true
Cluster ID: B
Remote side addr: cuic1
Auto config enabled: true

# ActiveMQ NetBridge..
Established

# Cluster state..
PAIRED-STANDBY
```

Live Data Syslog

Syslog servers and ports for the Live Data services are configured through the Unified Intelligence Center OAMP interface in the same way as the Unified Intelligence Center servers.

Note If Live Data Service goes down, an alert CUIC_LIVE_DATA_FEEDS_STOPPED is displayed in the RTMT counters (**Alert Central > Intelligence Center**). Use the failure details provided in the counters to troubleshoot the error scenario.

set live-data syslog-server

Required Minimum Privilege Level: Advanced

Use this command to set syslog configuration.

set live-data syslog-server syslogHostPrimary [*syslogPortPrimary*] [*syslogHostSecondary syslogPortSecondary*]

syslogHostPrimary

Specifies the primary host (fully-qualified domain name or IP address) for syslog.

syslogPortPrimary

The syslogPortPrimary parameter is optional. Specifies the port for syslog. The default value is 514.

syslogHostSecondary

The syslogHostSecondary parameter is optional. Specifies the secondary host (fully-qualified domain name or IP address) for syslog.

syslogPortSecondary

The syslogPortSecondary parameter is optional. Specifies the port for syslog. The default value is 514.

unset live-data syslog-server

Required Minimum Privilege Level: Advanced

Use this command to unset syslog configuration.

unset live-data syslog-server syslogHostQualifier {*primarysecondaryall*}
primary

Unset the primary host information (fully-qualified domain name and port).

secondary

Unset the secondary host information (fully-qualified domain name and port).

all

Unset the primary and secondary host information (fully-qualified domain name and port).

show live-data syslog-server

Required Minimum Privilege Level: Ordinary

Use this command to show the current configuration for the Live Data syslog server.

show live-data syslog-server

There are no required parameters.

Monitor and Analyze System Performance Using Nmon

Nmon is a tool to monitor and analyze performance data. The following commands start and stop the nmon data collection.

utils live-data nmon start

Required Minimum Privilege Level: Advanced

Use this command to start the nmon capture.

Command Syntax
utils live-data nmon start *s* [*seconds*] *c* [*count*]
s

Specifies the time interval (1 to 60 seconds) between each collection.

c

Specifies the number of collections that you want to perform. Each collection requires about 1 Kilobyte of disk space.

utils live-data nmon stop

Required Minimum Privilege Level: Advanced

Use this command to stop the nmon capture. The data that you capture in this nmon session is saved in `nmon_output.nmon`.

Command Syntax
utils live-data nmon stop

There are no required parameters.

Live Data Socket.IO

Live Data Socket.IO pushes the Live Data to the Unified Intelligence Center Live Data reports. Socket.IO receives data from the Live Data JMS feed and pushes the data to subscribing clients.

show socketio status

Required Minimum Privilege Level: Ordinary

Use this command to show the Socket.IO service status.

show socketio status

There are no required parameters.

Socket.IO Service Attribute	Value
Server Status	• Active - The server is in service. • Not Active - The server is not in service. • Unavailable - The server is not available.
JMS Brokers	JMS brokers configured for the Socket.IO service.
Active Broker	• Local - The Socket.IO service is using the local JMS broker. • Remote - The Socket.IO service is using the remote JMS broker.
Client Count	The total number of clients currently connected to the Socket.IO service.
Polling Client Count	The number of polling clients.

Note If the server cannot establish a JMX connection, the server status is Unavailable. No other status is displayed. If the JMS Brokers, Active Broker, Client Count, or Polling Client Count are not available, the information related to that status does not display.

The console output is similar to the following:

```
Server Status: Active
JMS Brokers: tcp://localhost:61616,tcp://192.168.1.56:61616
Active Broker: Local
Client Count: 2001 (polling: 312)
```

Live Data SNMP

You can monitor the health of Cisco Live Data using an industry standard SNMP (Simple Network Management Protocol) network management station (NMS). The Live Data reporting engine exposes an SNMP Management Information Base (MIB): **CISCO-LIVEDATA-MIB**. This MIB supports instrumentation specifically for Cisco Live Data.

Because Cisco Live Data is co-resident with Unified intelligence Center, the Cisco Live Data SNMP agent integrates with the existing Unified intelligence Center agent infrastructure and uses the same Unified intelligence Center **sysObjectID**. You can use the existing Unified Intelligence Center user interfaces to configure the Live Data SNMP agent. The Unified intelligence Center primary agent uses and maintains the Live Data configuration.

The Live Data configuration includes MIB-II "system" MIB values; SNMP v1 or v2c community strings or SNMP v3 user names (with associated authentication and encryption protocols); and notification destinations (network management stations). See Configure SNMP-Associated Settings in the *Administration Console User Guide for Cisco Unified Intelligence Center* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-intelligence-center/products-maintenance-guides-list.html for more details.

Live Data CISCO-LIVEDATA-MIB

The Live Data MIB, **CISCO-LIVEDATA-MIB** defines instrumentation unique to the Live Data servers (virtual machines). The instrumentation includes the following types of objects:

- **General Items** - attributes of the device and application.
- **Cluster Information** - cluster status and identity.

  Cluster status is shared across all nodes of the cluster; cluster status is not device-specific unless there is only one node in the cluster.
- **Service Table** - service status and identity.

  Exposed as a table.
- **Reporting Connection Table** - connection status and attributes (including metrics).

  Exposed as a table.
- **Event Table** -

  Exposed as a table and as SNMP notifications/traps.

Each of these tables is described in more detail below.

Note The MIB defines a single notification type; all nodes in all clusters may emit notifications.

The number of entries within each table may change over time, adapting to changes within the cluster.

Live Data MIB Textual Conventions

Table 1: Textual Conventions

Name	Syntax	Description
CldIndex	Unsigned32 (1..4294967295)	This syntax is used as the index into a table. A positive value identifies a unique entry in the table.
CldSeverity	INTEGER emergency(1), alert(2), critical(3), error(4), warning(5), notice(6), informational(7), debug(8)	This syntax is used to indicate the severity level of a notification or a logged event (or trace) message.

The severity levels are:

- **emergency**

  Events of this severity indicate that a devastating failure occurred; the system or service is unusable. Immediate operator intervention is required.

- **alert**

  Events of this severity indicate that a devastating failure is imminent that renders the system unusable. Immediate operator attention is necessary.

- **critical**

  Events of this severity indicate that a service-impacting failure is likely to occur soon or an error occurred that the system did not handle appropriately. Operator attention is needed as soon as possible.

- **error**

  Events of this severity contain important operational state information. The operational state information may indicate that the system experienced a temporary impairment or an error that the system handled appropriately. An operator should review the notification soon as possible to determine if more action is needed.

- **warning**

  Events of this severity contain important operational state information that may be a precursor to an error occurrence. An operator should review the event soon to determine if more action is needed.

- **notice**

Events of this severity contain health or operational state information that may be pertinent to the health of the system. Administrator attention is not immediately required.

- **informational**

  Events of this severity contain interesting system-level information that is valuable to an administrator in time, however, the event itself does not indicate a fault or an impairment condition.

- **debug**

  Events of this severity provide supplemental information that may help diagnose or resolve a problem, but do not necessarily provide operational health status.

Live Data MIB General Objects

Table 2: General Objects

Object Name	Data Type	Description
cldServerName	SnmpAdminString	The server name object is the fully-qualified domain name of the Cisco Live Data server.
cldDescription	SnmpAdminString	The description object holds a textual description of the Cisco Live Data software installed on this server. The description is typically the full name of the application.
cldVersion	SnmpAdminString	The version object identifies the version number of the Cisco LiveData software that is installed on this server.
cldStartTime	DateAndTime	The start time object is the date and time that the Cisco LiveData software (the primary application service) was started on this server.
cldTimeZoneName	SnmpAdminString	The time zone name object specifies the textual name of the time zone where the Cisco LiveData server (host) is physically located.
cldTimeZoneOffset	Integer32	The time zone offset minutes object represents the number of minutes that the local time, in the time zone where the Cisco LiveData server (host) is physically located, differs from Greenwich Mean Time (GMT).
cldEventNotifEnable	TruthValue	The notification enable object allows a management station to disable, during run time, all outgoing Cisco LiveData notifications. During a maintenance window, the management station frequently stops, reconfigures, and restarts many application components which can generate periodic floods of notifications. Therefore, the management station typically disables the notifications during a maintenance window. This setting is persistent even after a restart of the agent. The management station must explicitly reset this object value back to 'true' to re-enable outgoing application notifications from this device.

Live Data MIB Cluster Information

Table 3: Cluster Information

Object Name	Data Type	Description
cldClusterID	SnmpAdminString	The cluster identifier (ID) object holds a cluster-unique textual identifier for this cluster (for example, 'sideA').
cldClusterStatus	INTEGER: pairedActive(1), pairedStandby(2), isolatedActive(3), isolatedStandby(4), testing(5), outOfService(6)	The cluster status object indicates the status of this cluster of Cisco Live Data servers. A cluster is a group of one or more Cisco Live Data servers. The cluster works cooperatively to consume and process inbound real-time data from one or more data sources. The primary node distributes work between worker nodes within the cluster. A cluster may have a peer cluster in a fault-tolerant deployment model that assumes data processing duties in the event where its active peer cluster fails. • **pairedActive** The cluster is actively processing data and is communicating with its remote peer cluster. • **pairedStandby** The cluster is standing by (waiting to process data if necessary) and is communicating with its remote peer cluster. • **isolatedActive** The cluster is actively processing data but has lost peer-to-peer communication with its remote peer cluster. • **isolatedStandby** The cluster is standing by (waiting to process data if necessary) but has lost peer-to-peer communication with its remote peer cluster. • **testing** The cluster is unable to communicate with the remote peer cluster using the peer-to-peer connection. The cluster uses the 'test-other-side' procedure to determine whether to become active or go to a standby state. • **outOfService** The cluster is out of service.
cldClusterAddress	SnmpAdminString	The cluster address object holds the hostname or the IP address of the remote peer cluster for peer-to-peer communication with the remote cluster. NOTE: On the Publisher node, the value of this object is **N/A**

Live Data Service Table

Service Table Description

The service table is a list of Cisco Live Data dependent services. A service in this context is one or more executable processes that are configured to run on this server. Service table objects include both the service name and the current run state of that service. A single Live Data server has multiple running services, each of a different type, that encompasses the Live Data solution on a particular server. Some of these services work cooperatively with similar or dependent services on other server nodes in the cluster.

The SNMP agent constructs the service table at startup. The agent refreshes this table periodically during runtime to offer a near real-time status of configured services. The management station cannot add or delete service table entries. All objects in this table are read-only.

Service Entry Description

Each service entry represents a Cisco Live Data dependent service. The Live Data application software includes a collection of related services, each of which perform a specific, necessary function of the application.

Service Table Objects

Table 4: Service Table Objects

Object Name	Data Type	Description
cldServiceIndex	CldIndex	The service index is a value that uniquely identifies an entry in the services table. The SNMP agent arbitrarily assigns this value.
cldServiceName	SnmpAdminString	The service name is a user-intuitive textual name for the Cisco Live Data dependent service. (Note: as shown in the VOS "utils service list" command.)
cldServiceState	INTEGER: 'unknown' (1), 'disabled' (2), 'starting' (3), 'started' (4), 'active' (5), 'stopping' (6), 'stopped' (7)	The service state is the last known state of the Cisco LiveData dependent service. The object value identifies the run status of a configured service installed on the Cisco LiveData server. • **unknown** The status of the service cannot be determined. • **disabled** An administrator has explicitly disabled the service. • **starting** The service is currently starting up, but has not yet completed its startup procedure. • **started** The service completed its startup procedure and is currently running. • **active** The service is started, is currently running, and is actively processing data. • **stopping** The service is stopping and is in the midst of its shutdown procedure. • **stopped** The service is stopped. The service is dysfunctional or impaired, or an administrator has explicitly stopped it.
cldServiceUpTime	DateAndTime	The up time object indicates the date and time that this service started.

Live Data Reporting Connection Table

Reporting Connection Table Description

The reporting connection table is a list of Cisco Live Data server reporting connections. A Live Data server maintains several active connections to data sources. Most often, these connections are contact center solution nodes that generate real-time data that is ultimately used for creating reports.

Reporting connection table objects include objects that identify the reporting connection, the current state of that connection and a set of metrics and attributes that indicate connection health and performance. A single Live Data server has multiple reporting connections, each to a different peer node and to multiple data sources from a single node. The SNMP agent constructs the reporting connection table at startup. The agent refreshes this table periodically during runtime when each Live Data service reports connection states.

The management station cannot add or delete reporting connection table entries from the table. All objects in this table are read-only.

Reporting Connection Entry Description

Each reporting connection entry represents a Cisco Live Data reporting connection. The Live Data application connects to a number of data sources, each of which sends real-time data as a stream to the Live Data server.

Reporting Connection Objects

Table 5: Reporting Connection Objects

Object Name	Data Type	Description
cldRptConnIndex	CldIndex	The reporting connection index is a value that uniquely identifies an entry in the reporting connection table. The SNMP agent arbitrarily assigns this value.
cldRptConnServerID	SnmpAdminString	The reporting connection server identifier (ID) is a user-intuitive textual identification for the Cisco LiveData connection. This identifier is indicative of the source of the real-time data streamed using this reporting connection.
cldRptConnServerAddress	SnmpAdminString	The reporting connection server address object holds the hostname or IP address of the peer node in this reporting connection.
cldRptConnState	INTEGER: 'inactive' (1) 'active' (2)	The reporting connection state object indicates the current state of this reporting connection. The state is either active or inactive.
cldRptConnStateTime	DateAndTime	The reporting connection state time object records the date and time that this reporting connection transitioned into its current state.
cldRptConnEventRate	Gauge32	The reporting connection event rate indicates the number of events that arrive using this connection per second.
cldRptConnHeartbeatRTT	Gauge32	The reporting connection heartbeat round-trip time object indicates the time, in milliseconds, for heartbeat requests to return from the peer node in this reporting connection.
cldRptConnSocketConnects	Counter32	The reporting connection socket connects object counts the number of successful socket connections made to the peer node in this reporting connection.
cldRptConnSocketDisconnects	Counter32	The reporting connection socket disconnects object counts the number of socket disconnects with the peer node in this reporting connection. This object is used with cldConnSocketConnects to identify unstable connections to a particular endpoint.
cldRptConnMessagesDiscarded	Counter32	The reporting connection messages discarded object counts the number of discarded messages that the peer node sent in this reporting connection.
cldRptConnDSCP	Integer32	The reporting connection DSCP (Differentiated Services Code Point) object holds the Differentiated Services (DS) value currently used by this connection for Quality of Service (QoS) marking.

Live Data Event Table

Event Table Description

The event table is a list of active Cisco Live Data events. The SNMP agent constructs the event table at startup and it fills the table as 'raise' state events are generated. Events with the same cldEventID value overwrite existing events in the table with the same EventID (in other words, only the most recent events persist). The management station cannot add or delete event table entries from the table. All objects in this table are read-only.

Event Entry Description

Each event entry represents a Cisco Live Data event. The Live Data application software generates events when an unusual condition occurs that potentially affects the functioning of the Cisco Live Data server.

Event Table Objects

Table 6: Event Table Objects

Object Name	Data Type	Description
cldEventIndex	'CldIndex' TEXTUAL-CONVENTION	The event index is a value that uniquely identifies an entry in the event table. The SNMP agent arbitrarily assigns this value. .
cldEventID	Unsigned32	The event identifier (ID) object is the unique notification message identifier that the Live Data server assigns. This identifier is unique for each different notification but consistent for each instance of the same notification. Use this id to correlate 'clear' state notifications to 'raise' state notifications.
cldEventAppName	SnmpAdminString	The event application name object specifies the service-specific name of the functional service that generated this notification.
cldEventName	SnmpAdminString	The event name object specifies the service-specific name of the LiveData notification message. The object value is used to group and correlate similar notifications.
cldEventState	INTEGER: 'raise' (1), 'clear' (2)	The event state object identifies the state (not severity) of the notification and potentially the status of the functional component that generated the notification. The possible states are: • **raise** : A raise state identifies a notification received as a result of a health-impacting condition, such as a process failure. A subsequent clear state notification follows when the error condition is resolved. A node which generates a 'raise' state event may be impaired and likely requires an administrator's attention. • **clear** : The clear state indicates that the condition which generated a previous raise notification is resolved. This state may occur automatically with fault-tolerant deployments or may occur when an administrator intervenes.
cldEventSeverity	'CldSeverity' TEXTUAL-CONVENTION	The event severity object indicates the severity level of this notification.
cldEventTimestamp	DateAndTime	The event time stamp object specifies the date and time that the notification was generated on the originating device.

Object Name	Data Type	Description
cldEventText	SnmpAdminString	The event text is the full text of the notification. This text includes a description of the generated event, component state information, and potentially a brief description of administrative action that may be necessary to correct the condition that caused the event to occur.

Live Data MIB Notifications

Notification Type

cldEventNotif

Description

This notification describes an unusual condition that occurred that can potentially affect the functioning of the Cisco Live Data server. A functional service of the Cisco Live Data server sends a notification. The notification type provides operational state information about the service generating the notification at the time such service-impacting conditions occur.

Notification Type Objects

Object Name	Description
cldEventID	The unique event message identifier that the Live Data server assigns.
cldServerName	The host name or the fully qualified domain name of the Live Data server from which this event originated.
cldEventAppName	The service-specific name of the functional service that generated this notification.
cldEventName	The service-specific name of the Live Data notification message.
cldEventState	The state of the notification, either 'raise' or 'clear'.
cldEventSeverity	The severity level of this notification.
cldEventTimestamp	The date and time that the notification was generated.
cldEventText	The full text of the notification.

Live Data SNMP Event Correlation

The CISCO-LIVEDATA-MIB notification type (cldEventNotif) defines a set of objects that are contained within a Live Data SNMP notification. Live Data notifications are "stateful." A "raise" state event indicates a problem and a "clear" state event follows when the problem resolves or after the component engages fault-tolerance mechanisms to self-heal. To maintain an accurate state at the network management station, you can write rules to automatically correlate "clear" state events to existing "raise" state events and acknowledge those notifications at the management station.

The following notification type objects are used for event correlation.

Table 7: Live Data Notification Type Objects

Object Name	Description
cldEventID	The unique numeric event message identifier for this event.
cldServerName	The fully-qualified domain name of the Cisco Live Data server that generated the notification.
cldEventAppName	The name of the Cisco Live Data functional service that generated this event.
cldEventName	The service-specific name of the Cisco LiveData event message.
cldEventState	The state of the event, either 'raise' or 'clear'. A 'raise' state event generates when an unusual or service-impacting condition occurs. A 'clear' state event generates when a prior condition is resolved.

Live Data events are numerically identified in ascending order where "raise" state events have an odd value and "clear" state events have an even value. If a "raise" state event has a matching "clear" state event, the "clear" state event has the next (higher) even value. "Single-state raise" events are "raise" state events with no matching "clear" state event. A "single-state raise" event is an error condition that typically requires manual intervention to resolve.

To match "clear" state events to existing "raise" state events, match the object cldServerName (from the same device), cldEventAppName (from the same application), and cldEventName (the same event group) value from each event. In many cases, "clear" state events map to "raise" state events. The matching "raise" state event is that event with an even valued EventID that is less than the EventID value of the "clear" state event (for example, 202 is matched to 201). There may be more than one "raise" state event associated with that "clear" state event. The "clear" state event correlates with all existing "raise" state events with the matching ServerName, EventAppName and EventName. For example, assume that the "raise" state events #301 and #303 generate, followed by the "clear" state event #304. In this case, #304 correlates to both #301 and #303, acknowledging both "raise" state events.

To understand the relationship between certain "raise and "clear" state events, see the table of Live Data events and use the "See Also" field to relate events. Each event has a textual label to identify and relate each event in the table.

Events may have certain parameters associated with the event, such as a server IP address or a service state. These parameters are expressed as "tags" within the message text. A "tag" is a name/value pair surrounded by brackets, for example: [server_address=192.168.0.1]. The parameters are expressed this way to facilitate easier (automated) parsing of event text. Because the parameters are generalized across the full set of events, a separate table describes the parameters with labels associated for easy cross-referencing of an event with the parameters used.

Live Data SNMP Parameters

This table summarizes the parameters passed into the Live Data SNMP notifications.

Table 8: Live Data SNMP Parameters

Parameter ID	Tag	Description
PARAM_JMS_URL	jms_url	URL under which JMS server is located.

Parameter ID	Tag	Description
PARAM_JMS_SUBJECT	jms_subject	Topic, or queue, about which a JMS publication is issued.
PARAM_JMS_MESSAGE	jms_message	Message (typically JSON encoded) to/from JMS broker (ActiveMQ).
PARAM_AGENT_ID	agent_id	CCE agent identifier.
PARAM_TIP_MESSAGE	tip_message_class	CCE to Live Data (TIP) message class.
PARAM_TIP_CLIENT_SEQUENCE_GROUP	tip_client_seqgrp	TIP Client Sequence Group; Live Data, low-level protocol, current group sequence number.
PARAM_TIP_SERVER_SEQUENCE_GROUP	tip_client_seqgrp	TIP Client Sequence Group; Live Data, low-level protocol, current group sequence number.
PARAM_TIP_CLIENT_SEQUENCE_NUMBER	tip_client_app_seqnum	CCE, application level, current message sequence number.
PARAM_TIP_CLIENT_APP_SEQUENCE_NUMBER	tip_server_app_seqnum	Live Data, application level, current message sequence number.
PARAM_TIP_SERVER_APP_SEQUENCE_NUMBER	tip_server_app_seqnum	CCE, application level, current message sequence number.
PARAM_ERROR_DESC	message_error	Description of error associated with encoding/decoding/processing of CCE to Live Data protocol message.
PARAM_PERIPHERAL_ID	peripheral_id	CCE peripheral ID.
PARAM_SERVER_ID	server_id	Unique identifier to server or service id in CCE (example: PG) or Live Data (example: Router Spout).
PARAM_CONNECTION_USAGE	connection_usage	Defines to which protocol, or purpose, a given connection is associated with (TOS, TIP, and so on).
PARAM_SERVER_ADDRESS	server_address	IP or hostname to a server to which Live Data is a client (example: PG).
PARAM_SERVER_URL	server_url	URL to a server to which Live Data is a client (example: ActiveMQ).
PARAM_SERVER_PORT	server_port	IP port to connect to a server to which Live Data is a client (example: PG).
PARAM_SERVER_USERNAME	server_username	Username for accessing a given server.
PARAM_DATABASE_NAME	database_name	Database name.
PARAM_OPERATION_TYPE	operation_type	Description of an operation type (example: start, stop, disable, enable, and so on).
PARAM_OPERATION_ERROR_DESC	operation_error_desc	Error description associated with a given operation failure
PARAM_CONFIGURED_LIMIT	limit	Limit (maximum, or minimum) to a given configuration element.
PARAM_CONFIGURED_PROPERTIES	properties	Configuration properties and current values.
PARAM_DATABASE_OBJECT_TYPE	db_object_type	Database object as represented in-memory.

Parameter ID	Tag	Description
PARAM_DATABASE_OBJECT_ID	db_object_id	Database object id (typically unique key).
PARAM_DATABASE_VERSION_EXPECTED	db_ver_expected	Expected database schema version.
PARAM_DATABASE_VERSION_READ	db_ver_read	Database schema version retrieved from DB.
PARAM_JMX_MBEAN_NAME	jmx_mbean_name	JMX bean name.
PARAM_STATE	state	State description of a given object (example: State Machine state transition).
PARAM_PRIOR_STATE	prior_state	Prior state of a given object (example: connection state).
PARAM_TIP_SIDE	tip_server_side	CCE server side to which Live Data is associated (example: side A or side B).
PARAM_HEARTBEAT_MISSED_COUNT	missed_hearbeats	Currently missed heartbeats during communication CCE to Live Data.
PARAM_TIME_CHANGE	tip_time_change	CCE time server adjustment in milliseconds.
PARAM_CONNECTION_STATISTICS	connection_stats	CCE to Live Data connection statistics.
PARAM_AGENT_TEAM_ID	agent_team_id	CCE Agent Team Identifier.
PARAM_MRD_ID	mrd_id	CCE Media Router Domain Identifier.
PARAM_DESCR_GENERIC	descr	Generic Description field.
PARAM_ZOOKEEPER_ZNODE	znode	Zookeeper znode name.
PARAM_VALUE	value	Value associated with a given parameter.
PARAM_INPUT	input	Input value.
PARAM_CURRENT_STATE	current_state	Current state of a given object (example: connection state).
PARAM_NEW_STATE	new_state	New state of a given object (example: connection state).
PARAM_SEQ_NUM	seqnum	Message Sequence Number.
PARAM_MESSAGE	message	Actual message in text format.
PARAM_LATENCY	latency	Latency time.
PARAM_LAST_VIRTUAL_TIMESTAMP	last_vtimestamp	Last virtual time stamp (used for Live Data cluster messaging).
PARAM_NEW_VIRTUAL_TIMESTAMP	new_vtimestamp	New virtual time stamp (used for Live Data cluster messaging).

Live Data Collecting Logs

The logs that the Live Data services generate are available through the same tools as the Unified Intelligence Center logs.

For example, you can use the **file get** CLI command to collect all the Live Data logs:

```
file get activelog livedata/logs/*/*
```

You can also use the Real Time Monitoring Tool (RTMT) to collect and view logs and traces of the Live Data services.

Live Data Log Levels

Use the command-line interface to set trace level settings for Live Data services. There is no method in OAMP to set the log levels for the Live Data components.

You can use the **set live-data trace** command to set the log level or apply a tracemask to the following subsystems:

- Communications - logs messages related to connections
- DataProcessing - logs messages related to the processing of messages
- Database - logs messages specific to the database
- Event-store - logs messages specific to the storage of agent call-log and state-log events

Note You cannot apply a tracemask to the event-store subsystem.

Setting the loglevel

Required Minimum Privilege Level: Advanced

Command Syntax
set live-data trace *subsystem* **loglevel** *value*
subsystem

communications, dataprocessing, event-store or database

value

The loglevel for the specified subsystem:

- DEBUG
- INFO
- NOTICE
- WARN
- ERROR
- CRITICAL

- ALERT
- EMERGENCY

Note Only the following log levels are applicable to the event-store subsystem: DEBUG, INFO, WARN, and ERROR.

Example: `admin:set live-data trace dataprocessing loglevel DEBUG`

Setting the tracemask

Required Minimum Privilege Level: Advanced

You can set detailed log levels by enabling trace flags, which allows debug statements to appear in the logs. You can control debug tracing for specific functionalities by specifying a TRACE flag name within specific subsystem components. See Infrastructure Trace Definitions in the *Administration Console User Guide for Cisco Unified Intelligence Center* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-intelligence-center/products-maintenance-guides-list.html.

Command Syntax
set live-data trace *subsystem* **tracemask** *value*
subsystem
communications, dataprocessing, or database

value
For each of the three subsystems specify one of several tracemasks:

dataprocessing

- TIP_APPL_MESSAGE_TRACEMASK
- CAMEL_JMS_TRACEMASK
- STORM_SPOUT_TRACEMASK
- TIP_PROTOCOL_TRACEMASK
- FAILOVER_HB_TRACEMASK

database

- DB_UCCE_AW_TRACEMASK

communications

- JMS_COMMUNICATION_TRACEMASK
- FAILOVER_TOS_TRACEMASK

To set multiple tracemasks, separate them with a space.

Example: `admin: set live-data trace dataprocessing tracemask TIP_APPL_MESSAGE_TRACEMASK CAMEL_JMS_TRACEMASK`

To clear all tracemasks, use the tracemask command without parameters, for example:

```
set live-data trace dataprocessing tracemask
```

You can list the trace masks available for each subsystem using the command help option, for example:

```
set live-data trace dataprocessing tracemask ?
```

Show the loglevel

Required Minimum Privilege Level: Ordinary

To display the current loglevel or tracemask, use the `show live-data trace` command, for example:

```
admin:show live-data trace dataprocessing loglevel
```

```
admin:show live-data trace dataprocessing tracemask
```

Set Live-Data Trace Agent

Required Minimum Privilege Level: Advanced

Use this command to enable detailed tracing for specific agents.

Note This command requires in-depth system knowledge. Use detailed tracing only for advanced troubleshooting. Enabling detailed tracing adds many messages to the log files. The system may quickly exceed limits on the size and number of log files, causing the oldest log files to be overwritten. As a result, running detailed tracing shortens the history covered by the log files.

Command Syntax
set live-data trace agent *AgentSkillTargetIDs*
AgentSkillTargetIDs

The AgentSkillTargetIDs of the agents you want to trace. Separate IDs with a space. You can only have detailed tracing enabled for three agents simultaneously.

Note Running this command overwrites any previous setting.

Example: `admin:set live-data trace agent 5037 6000`

Output: `Enable detailed traces for agent(s) with the following id(s): 5037 6000`

Show Agents Currently Set

Required Minimum Privilege Level: Ordinary

This command shows the IDs of the agents that have detailed trace turned on.

show live-data trace agent

No parameters are required.

Example: `admin:show live-data trace agent`

Output: `Detailed traces are turned on for the agent(s) with the following id(s): 5037`

Unset Trace

Required Minimum Privilege Level: Advanced

This command turns off detailed traces for all agents.

unset live-data trace agent

No parameters are required.

Example: `admin:unset live-data trace agent`

Output: `Disable detailed traces for all agent(s)`

Help Command

unset live-data trace agent *?*

Example: `admin:set live-data trace agent ?`

Output: `This command is used to set the trace level for Agents.`

Set Live-Data Trace Skill-Group

Required Minimum Privilege Level: Advanced

Use this command to enable detailed tracing for specific skill-groups.

Note This command requires in-depth system knowledge. Use detailed tracing only for advanced troubleshooting. Enabling detailed tracing for a skill-group with many agents, for example, more than one hundred, adds a large number of messages to the log files. The system may quickly exceed limits on the size and number of log files, causing the oldest log files to be overwritten. As a result, running detailed tracing shortens the history covered by the log files.

Command Syntax
set live-data trace skill-group *Skill-GroupSkillTargetIDs*
Skill-GroupSkillTargetIDs

The Skill-GroupSkillTargetIDs of the skill-groups you want to trace. Separate IDs with a space. You can only have detailed tracing enabled for 3 skill-groups simultaneously.

Note Running this command overwrites any previous setting.

Example: `set live-data trace skill-group 5037 6000`

Output: `Enable detailed traces for skill-group(s) with the following id(s): 5037 6000`

Show Skill-Groups Currently Set

Required Minimum Privilege Level: Ordinary

show live-data trace skill-group

No parameters are required.

Example: `admin:show live-data trace skill-group`

`Output: Detailed traces are turned on for the skill-group(s) with the following id(s): 11962`

Unset Trace

Required Minimum Privilege Level: Advanced

This command turns off detailed tracing for all skill-groups.

unset live-data trace skill-group

No parameters are required.

Example: `admin:unset live-data trace skill-group`

Output: `Disable detailed traces for all skill-group(s)`

Help Command

unset live-data trace skill-group *?*

Example: `admin:set live-data trace skill-group ?`

Output: `This command is used to set the trace level for skill-groups.`

Set Live-Data Trace Precision-Queue

Required Minimum Privilege Level: Advanced

Use this command to enable tracing for specific precision-queues.

Note This command requires in-depth system knowledge. Use detailed tracing only for advanced troubleshooting. Enabling detailed tracing for a precision-queue adds many messages to the log files. The system may quickly exceed limits on the size and number of log files, causing the oldest log files to be overwritten. As a result, running detailed tracing shortens the history covered by the log files.

Command Syntax
set live-data trace precision-queue *Precision-QueueIDs*
Precision-QueueIDs

The Precision-QueueIDs of the precision-queues you want to trace. Separate IDs with a space. You can only have detailed tracing enabled for 3 precision-queues simultaneously.

Note Running this command overwrites any previous setting.

Example: `admin:set live-data trace precision-queue 5037 6000`

Output: `Enable detailed traces for precision-queue(s) with the following id(s): 5037 6000`

Show Precision-Queues Currently Set

Required Minimum Privilege Level: Ordinary

Show the IDs of the precision-queues that have detailed trace turned on.

show live-data trace precision-queue

No parameters are required.

Example: `admin:show live-data trace precision-queue`

Output: `Detailed traces are turned on for the precision-queue(s) with the following id(s): 5000`

Unset Trace

Required Minimum Privilege Level: Advanced

This command turns off detailed traces for all precision-queues.

unset live-data trace precision-queue

No parameters are required.

Example: `admin:unset live-data trace precision-queue`

Output: `Disable detailed traces for all precision-queue(s)`

Help Command

unset live-data trace precision-queue *?*

Example: `admin:set live-data trace precision-queue ?`

Output: `This command is used to set the trace level for precision-queues.`

Live Data Syslog

Syslog servers and ports for the Live Data services are configured through the Unified Intelligence Center OAMP interface in the same way as the Unified Intelligence Center servers.

Note If Live Data Service goes down, an alert CUIC_LIVE_DATA_FEEDS_STOPPED is displayed in the RTMT counters (**Alert Central > Intelligence Center**). Use the failure details provided in the counters to troubleshoot the error scenario.

set live-data syslog-server

Required Minimum Privilege Level: Advanced

Use this command to set syslog configuration.

set live-data syslog-server syslogHostPrimary [*syslogPortPrimary*] [*syslogHostSecondary syslogPortSecondary*]
syslogHostPrimary
Specifies the primary host (fully-qualified domain name or IP address) for syslog.

syslogPortPrimary

The syslogPortPrimary parameter is optional. Specifies the port for syslog. The default value is 514.

syslogHostSecondary

The syslogHostSecondary parameter is optional. Specifies the secondary host (fully-qualified domain name or IP address) for syslog.

syslogPortSecondary

The syslogPortSecondary parameter is optional. Specifies the port for syslog. The default value is 514.

unset live-data syslog-server

Required Minimum Privilege Level: Advanced

Use this command to unset syslog configuration.

unset live-data syslog-server syslogHostQualifier {*primarysecondaryall*}
primary

Unset the primary host information (fully-qualified domain name and port).

secondary

Unset the secondary host information (fully-qualified domain name and port).

all

Unset the primary and secondary host information (fully-qualified domain name and port).

show live-data syslog-server

Required Minimum Privilege Level: Ordinary

Use this command to show the current configuration for the Live Data syslog server.

show live-data syslog-server

There are no required parameters.

Live Data Socket.IO

Live Data Socket.IO pushes the Live Data to the Unified Intelligence Center Live Data reports. Socket.IO receives data from the Live Data JMS feed and pushes the data to subscribing clients.

show socketio status

Required Minimum Privilege Level: Ordinary

Use this command to show the Socket.IO service status.

show socketio status

There are no required parameters.

Socket.IO Service Attribute	Value
Server Status	• Active - The server is in service. • Not Active - The server is not in service. • Unavailable - The server is not available.
JMS Brokers	JMS brokers configured for the Socket.IO service.
Active Broker	• Local - The Socket.IO service is using the local JMS broker. • Remote - The Socket.IO service is using the remote JMS broker.
Client Count	The total number of clients currently connected to the Socket.IO service.
Polling Client Count	The number of polling clients.

Note If the server cannot establish a JMX connection, the server status is Unavailable. No other status is displayed. If the JMS Brokers, Active Broker, Client Count, or Polling Client Count are not available, the information related to that status does not display.

The console output is similar to the following:

```
Server Status: Active
JMS Brokers: tcp://localhost:61616,tcp://192.168.1.56:61616
Active Broker: Local
Client Count: 2001 (polling: 312)
```

Live Data Collecting Logs

The logs that the Live Data services generate are available through the same tools as the Unified Intelligence Center logs.

For example, you can use the **file get** CLI command to collect all the Live Data logs:

```
file get activelog livedata/logs/*/*
```

You can also use the Real Time Monitoring Tool (RTMT) to collect and view logs and traces of the Live Data services.

Live Data Log Levels

Use the command-line interface to set trace level settings for Live Data services. There is no method in OAMP to set the log levels for the Live Data components.

You can use the **set live-data trace** command to set the log level or apply a tracemask to the following subsystems:

- Communications - logs messages related to connections
- DataProcessing - logs messages related to the processing of messages
- Database - logs messages specific to the database
- Event-store - logs messages specific to the storage of agent call-log and state-log events

Note You cannot apply a tracemask to the event-store subsystem.

Setting the loglevel

Required Minimum Privilege Level: Advanced

Command Syntax
set live-data trace *subsystem* **loglevel** *value*
subsystem

communications, dataprocessing, event-store or database

value

The loglevel for the specified subsystem:

- DEBUG
- INFO
- NOTICE
- WARN
- ERROR
- CRITICAL
- ALERT
- EMERGENCY

Note Only the following log levels are applicable to the event-store subsystem: DEBUG, INFO, WARN, and ERROR.

Example: `admin:set live-data trace dataprocessing loglevel DEBUG`

Setting the tracemask

Required Minimum Privilege Level: Advanced

You can set detailed log levels by enabling trace flags, which allows debug statements to appear in the logs. You can control debug tracing for specific functionalities by specifying a TRACE flag name within specific subsystem components. See Infrastructure Trace Definitions in the *Administration Console User Guide for Cisco Unified Intelligence Center* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-intelligence-center/products-maintenance-guides-list.html.

Command Syntax
set live-data trace *subsystem* **tracemask** *value*
subsystem

communications, dataprocessing, or database

value

For each of the three subsystems specify one of several tracemasks:

dataprocessing

- TIP_APPL_MESSAGE_TRACEMASK
- CAMEL_JMS_TRACEMASK
- STORM_SPOUT_TRACEMASK
- TIP_PROTOCOL_TRACEMASK
- FAILOVER_HB_TRACEMASK

database

- DB_UCCE_AW_TRACEMASK

communications

- JMS_COMMUNICATION_TRACEMASK
- FAILOVER_TOS_TRACEMASK

To set multiple tracemasks, separate them with a space.

Example: `admin: set live-data trace dataprocessing tracemask TIP_APPL_MESSAGE_TRACEMASK CAMEL_JMS_TRACEMASK`

To clear all tracemasks, use the tracemask command without parameters, for example:

```
set live-data trace dataprocessing tracemask
```

You can list the trace masks available for each subsystem using the command help option, for example:

```
set live-data trace dataprocessing tracemask ?
```

Show the loglevel

Required Minimum Privilege Level: Ordinary

To display the current loglevel or tracemask, use the `show live-data trace` command, for example:

```
admin:show live-data trace dataprocessing loglevel
```

```
admin:show live-data trace dataprocessing tracemask
```

Set Live-Data Trace Agent

Required Minimum Privilege Level: Advanced

Use this command to enable detailed tracing for specific agents.

Note This command requires in-depth system knowledge. Use detailed tracing only for advanced troubleshooting. Enabling detailed tracing adds many messages to the log files. The system may quickly exceed limits on the size and number of log files, causing the oldest log files to be overwritten. As a result, running detailed tracing shortens the history covered by the log files.

Command Syntax
set live-data trace agent *AgentSkillTargetIDs*
AgentSkillTargetIDs

The AgentSkillTargetIDs of the agents you want to trace. Separate IDs with a space. You can only have detailed tracing enabled for three agents simultaneously.

Note Running this command overwrites any previous setting.

Example: `admin:set live-data trace agent 5037 6000`

Output: `Enable detailed traces for agent(s) with the following id(s): 5037 6000`

Show Agents Currently Set

Required Minimum Privilege Level: Ordinary

This command shows the IDs of the agents that have detailed trace turned on.

show live-data trace agent

No parameters are required.

Example: `admin:show live-data trace agent`

Output: `Detailed traces are turned on for the agent(s) with the following id(s): 5037`

Unset Trace

Required Minimum Privilege Level: Advanced

This command turns off detailed traces for all agents.

unset live-data trace agent

No parameters are required.

Example: `admin:unset live-data trace agent`

Output: `Disable detailed traces for all agent(s)`

Help Command

unset live-data trace agent *?*

Example: `admin:set live-data trace agent ?`

Output: `This command is used to set the trace level for Agents.`

Set Live-Data Trace Skill-Group

Required Minimum Privilege Level: Advanced

Use this command to enable detailed tracing for specific skill-groups.

Note This command requires in-depth system knowledge. Use detailed tracing only for advanced troubleshooting. Enabling detailed tracing for a skill-group with many agents, for example, more than one hundred, adds a large number of messages to the log files. The system may quickly exceed limits on the size and number of log files, causing the oldest log files to be overwritten. As a result, running detailed tracing shortens the history covered by the log files.

Command Syntax
set live-data trace skill-group *Skill-GroupSkillTargetIDs*
Skill-GroupSkillTargetIDs

The Skill-GroupSkillTargetIDs of the skill-groups you want to trace. Separate IDs with a space. You can only have detailed tracing enabled for 3 skill-groups simultaneously.

Note Running this command overwrites any previous setting.

Example: `set live-data trace skill-group 5037 6000`

Output: `Enable detailed traces for skill-group(s) with the following id(s): 5037 6000`

Show Skill-Groups Currently Set

Required Minimum Privilege Level: Ordinary

show live-data trace skill-group

No parameters are required.

Example: `admin:show live-data trace skill-group`

`Output: Detailed traces are turned on for the skill-group(s) with the following id(s): 11962`

Unset Trace

Required Minimum Privilege Level: Advanced

This command turns off detailed tracing for all skill-groups.

unset live-data trace skill-group

No parameters are required.

Example: `admin:unset live-data trace skill-group`

Output: `Disable detailed traces for all skill-group(s)`

Help Command

unset live-data trace skill-group *?*

Example: `admin:set live-data trace skill-group ?`

Output: `This command is used to set the trace level for skill-groups.`

Set Live-Data Trace Precision-Queue

Required Minimum Privilege Level: Advanced

Use this command to enable tracing for specific precision-queues.

Note This command requires in-depth system knowledge. Use detailed tracing only for advanced troubleshooting. Enabling detailed tracing for a precision-queue adds many messages to the log files. The system may quickly exceed limits on the size and number of log files, causing the oldest log files to be overwritten. As a result, running detailed tracing shortens the history covered by the log files.

Command Syntax
set live-data trace precision-queue *Precision-QueueIDs*
Precision-QueueIDs

The Precision-QueueIDs of the precision-queues you want to trace. Separate IDs with a space. You can only have detailed tracing enabled for 3 precision-queues simultaneously.

Note Running this command overwrites any previous setting.

Example: `admin:set live-data trace precision-queue 5037 6000`

Output: `Enable detailed traces for precision-queue(s) with the following id(s): 5037 6000`

Show Precision-Queues Currently Set

Required Minimum Privilege Level: Ordinary

Show the IDs of the precision-queues that have detailed trace turned on.

show live-data trace precision-queue

No parameters are required.

Example: `admin:show live-data trace precision-queue`

Output: `Detailed traces are turned on for the precision-queue(s) with the following id(s): 5000`

Unset Trace

Required Minimum Privilege Level: Advanced

This command turns off detailed traces for all precision-queues.

unset live-data trace precision-queue

No parameters are required.

Example: `admin:unset live-data trace precision-queue`

Output: `Disable detailed traces for all precision-queue(s)`

Help Command

unset live-data trace precision-queue *?*

Example: `admin:set live-data trace precision-queue ?`

Output: `This command is used to set the trace level for precision-queues.`

Live Data Failover Configuration

The following Live Data failover commands are provided to allow you to monitor the Live Data failover mechanism during troubleshooting.

By default, Live Data failover is automatically configured during installation or upgrade. Under general operations you do not need to use these commands. Use **show live-data failover** to display information on the current configuration and state of Live Data reporting system cluster. Use **set live-data failover** to enable Live Data failover, and **unset live-data failover** to unset Live Data failover.

set live-data failover

Required Minimum Privilege Level: Advanced

Use this command to enable Live Data failover. This command automatically sets the system to run in duplex mode. Run this command on both Side A and Side B.

Command Syntax
set live-data failover

There are no required parameters.

unset live-data failover

Required Minimum Privilege Level: Advanced

Use this command to unset Live Data failover. This command automatically sets the system to run in simplex mode. Run this command on both Side A and Side B.

Command Syntax
unset live-data failover

There are no required parameters.

show live-data failover

Required Minimum Privilege Level: Ordinary

Use this command to display the Live Data cluster failover status and settings.

Command Syntax
show live-data failover
There are no parameters.

The command returns information on the Live Data server on which you run the command. The Live Data server information includes the current Live Data cluster settings, the status of the ActiveMQ connection, and the state of the cluster.

The possible cluster states are:

Cluster state	Description
PAIRED-ACTIVE	The cluster is in the active state and is communicating with the remote side.
PAIRED-STANDBY	The cluster is in the standby state and is communicating with the remote side.
ISOLATED-ACTIVE	The cluster is in the active state, but it is not communicating with the remote side.
ISOLATED-STANDBY	The cluster is in the standby state, but it is not communicating with the remote side.
SIMPLEXED-MODE	The cluster is working in simplex mode.
OUT-OF-SERVICE	The cluster is out of service.
CONNECTING	The cluster is attempting to do a handshake with the remote side.
TESTING	The cluster is unable to communicate with the remote side and is using the Test-Other-Side procedure to determine whether to become active or standby.

The console output after you run this command on the publisher side of a Live Data system is similar to the following:

```
admin:show live-data failover
# Failover settings..
Cluster failover enabled: true
Cluster ID: A
Remote side addr: not applicable for the publisher in auto-config
Auto config enabled: true

# ActiveMQ NetBridge..
Established

# Cluster state..
PAIRED-ACTIVE
```

Sample console output on the subscriber side is as follows:

```
admin:show live-data failover
# Failover settings..
Cluster failover enabled: true
```

```
Cluster ID: B
Remote side addr: cuic1
Auto config enabled: true

# ActiveMQ NetBridge..
Established

# Cluster state..
PAIRED-STANDBY
```

Live Data Syslog

Syslog servers and ports for the Live Data services are configured through the Unified Intelligence Center OAMP interface in the same way as the Unified Intelligence Center servers.

Note If Live Data Service goes down, an alert CUIC_LIVE_DATA_FEEDS_STOPPED is displayed in the RTMT counters (**Alert Central > Intelligence Center**). Use the failure details provided in the counters to troubleshoot the error scenario.

set live-data syslog-server

Required Minimum Privilege Level: Advanced

Use this command to set syslog configuration.

set live-data syslog-server syslogHostPrimary [*syslogPortPrimary*] [*syslogHostSecondary syslogPortSecondary*]

syslogHostPrimary

Specifies the primary host (fully-qualified domain name or IP address) for syslog.

syslogPortPrimary

The syslogPortPrimary parameter is optional. Specifies the port for syslog. The default value is 514.

syslogHostSecondary

The syslogHostSecondary parameter is optional. Specifies the secondary host (fully-qualified domain name or IP address) for syslog.

syslogPortSecondary

The syslogPortSecondary parameter is optional. Specifies the port for syslog. The default value is 514.

unset live-data syslog-server

Required Minimum Privilege Level: Advanced

Use this command to unset syslog configuration.

unset live-data syslog-server syslogHostQualifier {*primarysecondaryall*}
primary

Unset the primary host information (fully-qualified domain name and port).

secondary

Unset the secondary host information (fully-qualified domain name and port).

all

Unset the primary and secondary host information (fully-qualified domain name and port).

show live-data syslog-server

Required Minimum Privilege Level: Ordinary

Use this command to show the current configuration for the Live Data syslog server.

show live-data syslog-server

There are no required parameters.

Monitor and Analyze System Performance Using Nmon

Nmon is a tool to monitor and analyze performance data. The following commands start and stop the nmon data collection.

utils live-data nmon start

Required Minimum Privilege Level: Advanced

Use this command to start the nmon capture.

Command Syntax
utils live-data nmon start *s* [*seconds*] *c* [*count*]
s

Specifies the time interval (1 to 60 seconds) between each collection.

c

Specifies the number of collections that you want to perform. Each collection requires about 1 Kilobyte of disk space.

utils live-data nmon stop

Required Minimum Privilege Level: Advanced

Use this command to stop the nmon capture. The data that you capture in this nmon session is saved in `nmon_output.nmon`.

Command Syntax
utils live-data nmon stop

There are no required parameters.

Live Data Socket.IO

Live Data Socket.IO pushes the Live Data to the Unified Intelligence Center Live Data reports. Socket.IO receives data from the Live Data JMS feed and pushes the data to subscribing clients.

show socketio status

Required Minimum Privilege Level: Ordinary

Use this command to show the Socket.IO service status.

show socketio status

There are no required parameters.

Socket.IO Service Attribute	Value
Server Status	• Active - The server is in service. • Not Active - The server is not in service. • Unavailable - The server is not available.
JMS Brokers	JMS brokers configured for the Socket.IO service.
Active Broker	• Local - The Socket.IO service is using the local JMS broker. • Remote - The Socket.IO service is using the remote JMS broker.
Client Count	The total number of clients currently connected to the Socket.IO service.
Polling Client Count	The number of polling clients.

Note If the server cannot establish a JMX connection, the server status is Unavailable. No other status is displayed. If the JMS Brokers, Active Broker, Client Count, or Polling Client Count are not available, the information related to that status does not display.

The console output is similar to the following:

```
Server Status: Active
JMS Brokers: tcp://localhost:61616,tcp://192.168.1.56:61616
Active Broker: Local
Client Count: 2001 (polling: 312)
```

Live Data SNMP

You can monitor the health of Cisco Live Data using an industry standard SNMP (Simple Network Management Protocol) network management station (NMS). The Live Data reporting engine exposes an SNMP Management Information Base (MIB): **CISCO-LIVEDATA-MIB**. This MIB supports instrumentation specifically for Cisco Live Data.

Because Cisco Live Data is co-resident with Unified intelligence Center, the Cisco Live Data SNMP agent integrates with the existing Unified intelligence Center agent infrastructure and uses the same Unified intelligence Center **sysObjectID**. You can use the existing Unified Intelligence Center user interfaces to configure the Live Data SNMP agent. The Unified intelligence Center primary agent uses and maintains the Live Data configuration.

The Live Data configuration includes MIB-II "system" MIB values; SNMP v1 or v2c community strings or SNMP v3 user names (with associated authentication and encryption protocols); and notification destinations (network management stations). See Configure SNMP-Associated Settings in the *Administration Console User Guide for Cisco Unified Intelligence Center* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-intelligence-center/products-maintenance-guides-list.html for more details.

Live Data CISCO-LIVEDATA-MIB

The Live Data MIB, **CISCO-LIVEDATA-MIB** defines instrumentation unique to the Live Data servers (virtual machines). The instrumentation includes the following types of objects:

- **General Items** - attributes of the device and application.
- **Cluster Information** - cluster status and identity.

  Cluster status is shared across all nodes of the cluster; cluster status is not device-specific unless there is only one node in the cluster.
- **Service Table** - service status and identity.

  Exposed as a table.
- **Reporting Connection Table** - connection status and attributes (including metrics).

  Exposed as a table.
- **Event Table** -

  Exposed as a table and as SNMP notifications/traps.

Each of these tables is described in more detail below.

Note The MIB defines a single notification type; all nodes in all clusters may emit notifications.

The number of entries within each table may change over time, adapting to changes within the cluster.

Live Data MIB Textual Conventions

Table 9: Textual Conventions

Name	Syntax	Description
CldIndex	Unsigned32 (1..4294967295)	This syntax is used as the index into a table. A positive value identifies a unique entry in the table.
CldSeverity	INTEGER emergency(1), alert(2), critical(3), error(4), warning(5), notice(6), informational(7), debug(8)	This syntax is used to indicate the severity level of a notification or a logged event (or trace) message.

The severity levels are:

- **emergency**

  Events of this severity indicate that a devastating failure occurred; the system or service is unusable. Immediate operator intervention is required.

- **alert**

  Events of this severity indicate that a devastating failure is imminent that renders the system unusable. Immediate operator attention is necessary.

- **critical**

  Events of this severity indicate that a service-impacting failure is likely to occur soon or an error occurred that the system did not handle appropriately. Operator attention is needed as soon as possible.

- **error**

  Events of this severity contain important operational state information. The operational state information may indicate that the system experienced a temporary impairment or an error that the system handled appropriately. An operator should review the notification soon as possible to determine if more action is needed.

- **warning**

  Events of this severity contain important operational state information that may be a precursor to an error occurrence. An operator should review the event soon to determine if more action is needed.

- **notice**

  Events of this severity contain health or operational state information that may be pertinent to the health of the system. Administrator attention is not immediately required.

- **informational**

  Events of this severity contain interesting system-level information that is valuable to an administrator in time, however, the event itself does not indicate a fault or an impairment condition.

- **debug**

  Events of this severity provide supplemental information that may help diagnose or resolve a problem, but do not necessarily provide operational health status.

Live Data MIB General Objects

Table 10: General Objects

Object Name	Data Type	Description
cldServerName	SnmpAdminString	The server name object is the fully-qualified domain name of the Cisco Live Data server.
cldDescription	SnmpAdminString	The description object holds a textual description of the Cisco Live Data software installed on this server. The description is typically the full name of the application.
cldVersion	SnmpAdminString	The version object identifies the version number of the Cisco LiveData software that is installed on this server.
cldStartTime	DateAndTime	The start time object is the date and time that the Cisco LiveData software (the primary application service) was started on this server.
cldTimeZoneName	SnmpAdminString	The time zone name object specifies the textual name of the time zone where the Cisco LiveData server (host) is physically located.
cldTimeZoneOffset	Integer32	The time zone offset minutes object represents the number of minutes that the local time, in the time zone where the Cisco LiveData server (host) is physically located, differs from Greenwich Mean Time (GMT).
cldEventNotifEnable	TruthValue	The notification enable object allows a management station to disable, during run time, all outgoing Cisco LiveData notifications. During a maintenance window, the management station frequently stops, reconfigures, and restarts many application components which can generate periodic floods of notifications. Therefore, the management station typically disables the notifications during a maintenance window. This setting is persistent even after a restart of the agent. The management station must explicitly reset this object value back to 'true' to re-enable outgoing application notifications from this device.

Live Data MIB Cluster Information

Table 11: Cluster Information

Object Name	Data Type	Description
cldClusterID	SnmpAdminString	The cluster identifier (ID) object holds a cluster-unique textual identifier for this cluster (for example, 'sideA').

Object Name	Data Type	Description
cldClusterStatus	INTEGER: pairedActive(1), pairedStandby(2), isolatedActive(3), isolatedStandby(4), testing(5), outOfService(6)	The cluster status object indicates the status of this cluster of Cisco Live Data servers. A cluster is a group of one or more Cisco Live Data servers. The cluster works cooperatively to consume and process inbound real-time data from one or more data sources. The primary node distributes work between worker nodes within the cluster. A cluster may have a peer cluster in a fault-tolerant deployment model that assumes data processing duties in the event where its active peer cluster fails. • **pairedActive** The cluster is actively processing data and is communicating with its remote peer cluster. • **pairedStandby** The cluster is standing by (waiting to process data if necessary) and is communicating with its remote peer cluster. • **isolatedActive** The cluster is actively processing data but has lost peer-to-peer communication with its remote peer cluster. • **isolatedStandby** The cluster is standing by (waiting to process data if necessary) but has lost peer-to-peer communication with its remote peer cluster. • **testing** The cluster is unable to communicate with the remote peer cluster using the peer-to-peer connection. The cluster uses the 'test-other-side' procedure to determine whether to become active or go to a standby state. • **outOfService** The cluster is out of service.
cldClusterAddress	SnmpAdminString	The cluster address object holds the hostname or the IP address of the remote peer cluster for peer-to-peer communication with the remote cluster. NOTE: On the Publisher node, the value of this object is **N/A**

Live Data Service Table

Service Table Description

The service table is a list of Cisco Live Data dependent services. A service in this context is one or more executable processes that are configured to run on this server. Service table objects include both the service name and the current run state of that service. A single Live Data server has multiple running services, each of a different type, that encompasses the Live Data solution on a particular server. Some of these services work cooperatively with similar or dependent services on other server nodes in the cluster.

The SNMP agent constructs the service table at startup. The agent refreshes this table periodically during runtime to offer a near real-time status of configured services. The management station cannot add or delete service table entries. All objects in this table are read-only.

Service Entry Description

Each service entry represents a Cisco Live Data dependent service. The Live Data application software includes a collection of related services, each of which perform a specific, necessary function of the application.

Service Table Objects

Table 12: Service Table Objects

Object Name	Data Type	Description
cldServiceIndex	CldIndex	The service index is a value that uniquely identifies an entry in the services table. The SNMP agent arbitrarily assigns this value.

Object Name	Data Type	Description
cldServiceName	SnmpAdminString	The service name is a user-intuitive textual name for the Cisco Live Data dependent service. (Note: as shown in the VOS "utils service list" command.)
cldServiceState	INTEGER: 'unknown' (1), 'disabled' (2), 'starting' (3), 'started' (4), 'active' (5), 'stopping' (6), 'stopped' (7)	The service state is the last known state of the Cisco LiveData dependent service. The object value identifies the run status of a configured service installed on the Cisco LiveData server. • **unknown** The status of the service cannot be determined. • **disabled** An administrator has explicitly disabled the service. • **starting** The service is currently starting up, but has not yet completed its startup procedure. • **started** The service completed its startup procedure and is currently running. • **active** The service is started, is currently running, and is actively processing data. • **stopping** The service is stopping and is in the midst of its shutdown procedure. • **stopped** The service is stopped. The service is dysfunctional or impaired, or an administrator has explicitly stopped it.
cldServiceUpTime	DateAndTime	The up time object indicates the date and time that this service started.

Live Data Reporting Connection Table

Reporting Connection Table Description

The reporting connection table is a list of Cisco Live Data server reporting connections. A Live Data server maintains several active connections to data sources. Most often, these connections are contact center solution nodes that generate real-time data that is ultimately used for creating reports.

Reporting connection table objects include objects that identify the reporting connection, the current state of that connection and a set of metrics and attributes that indicate connection health and performance. A single Live Data server has multiple reporting connections, each to a different peer node and to multiple data sources from a single node. The SNMP agent constructs the reporting connection table at startup. The agent refreshes this table periodically during runtime when each Live Data service reports connection states.

The management station cannot add or delete reporting connection table entries from the table. All objects in this table are read-only.

Reporting Connection Entry Description

Each reporting connection entry represents a Cisco Live Data reporting connection. The Live Data application connects to a number of data sources, each of which sends real-time data as a stream to the Live Data server.

Reporting Connection Objects

Table 13: Reporting Connection Objects

Object Name	Data Type	Description
cldRptConnIndex	CldIndex	The reporting connection index is a value that uniquely identifies an entry in the reporting connection table. The SNMP agent arbitrarily assigns this value.
cldRptConnServerID	SnmpAdminString	The reporting connection server identifier (ID) is a user-intuitive textual identification for the Cisco LiveData connection. This identifier is indicative of the source of the real-time data streamed using this reporting connection.
cldRptConnServerAddress	SnmpAdminString	The reporting connection server address object holds the hostname or IP address of the peer node in this reporting connection.
cldRptConnState	INTEGER: 'inactive' (1) 'active' (2)	The reporting connection state object indicates the current state of this reporting connection. The state is either active or inactive.
cldRptConnStateTime	DateAndTime	The reporting connection state time object records the date and time that this reporting connection transitioned into its current state.
cldRptConnEventRate	Gauge32	The reporting connection event rate indicates the number of events that arrive using this connection per second.
cldRptConnHeartbeatRTT	Gauge32	The reporting connection heartbeat round-trip time object indicates the time, in milliseconds, for heartbeat requests to return from the peer node in this reporting connection.
cldRptConnSocketConnects	Counter32	The reporting connection socket connects object counts the number of successful socket connections made to the peer node in this reporting connection.
cldRptConnSocketDisconnects	Counter32	The reporting connection socket disconnects object counts the number of socket disconnects with the peer node in this reporting connection. This object is used with cldConnSocketConnects to identify unstable connections to a particular endpoint.
cldRptConnMessagesDiscarded	Counter32	The reporting connection messages discarded object counts the number of discarded messages that the peer node sent in this reporting connection.
cldRptConnDSCP	Integer32	The reporting connection DSCP (Differentiated Services Code Point) object holds the Differentiated Services (DS) value currently used by this connection for Quality of Service (QoS) marking.

Live Data Event Table

Event Table Description

The event table is a list of active Cisco Live Data events. The SNMP agent constructs the event table at startup and it fills the table as 'raise' state events are generated. Events with the same cldEventID value overwrite existing events in the table with the same EventID (in other words, only the most recent events persist). The management station cannot add or delete event table entries from the table. All objects in this table are read-only.

Event Entry Description

Each event entry represents a Cisco Live Data event. The Live Data application software generates events when an unusual condition occurs that potentially affects the functioning of the Cisco Live Data server.

Event Table Objects

Table 14: Event Table Objects

Object Name	Data Type	Description
cldEventIndex	'CldIndex' TEXTUAL-CONVENTION	The event index is a value that uniquely identifies an entry in the event table. The SNMP agent arbitrarily assigns this value. .
cldEventID	Unsigned32	The event identifier (ID) object is the unique notification message identifier that the Live Data server assigns. This identifier is unique for each different notification but consistent for each instance of the same notification. Use this id to correlate 'clear' state notifications to 'raise' state notifications.
cldEventAppName	SnmpAdminString	The event application name object specifies the service-specific name of the functional service that generated this notification.
cldEventName	SnmpAdminString	The event name object specifies the service-specific name of the LiveData notification message. The object value is used to group and correlate similar notifications.
cldEventState	INTEGER: 'raise' (1), 'clear' (2)	The event state object identifies the state (not severity) of the notification and potentially the status of the functional component that generated the notification. The possible states are: • **raise** : A raise state identifies a notification received as a result of a health-impacting condition, such as a process failure. A subsequent clear state notification follows when the error condition is resolved. A node which generates a 'raise' state event may be impaired and likely requires an administrator's attention. • **clear** : The clear state indicates that the condition which generated a previous raise notification is resolved. This state may occur automatically with fault-tolerant deployments or may occur when an administrator intervenes.
cldEventSeverity	'CldSeverity' TEXTUAL-CONVENTION	The event severity object indicates the severity level of this notification.
cldEventTimestamp	DateAndTime	The event time stamp object specifies the date and time that the notification was generated on the originating device.

Object Name	Data Type	Description
cldEventText	SnmpAdminString	The event text is the full text of the notification. This text includes a description of the generated event, component state information, and potentially a brief description of administrative action that may be necessary to correct the condition that caused the event to occur.

Live Data MIB Notifications

Notification Type

cldEventNotif

Description

This notification describes an unusual condition that occurred that can potentially affect the functioning of the Cisco Live Data server. A functional service of the Cisco Live Data server sends a notification. The notification type provides operational state information about the service generating the notification at the time such service-impacting conditions occur.

Notification Type Objects

Object Name	Description
cldEventID	The unique event message identifier that the Live Data server assigns.
cldServerName	The host name or the fully qualified domain name of the Live Data server from which this event originated.
cldEventAppName	The service-specific name of the functional service that generated this notification.
cldEventName	The service-specific name of the Live Data notification message.
cldEventState	The state of the notification, either 'raise' or 'clear'.
cldEventSeverity	The severity level of this notification.
cldEventTimestamp	The date and time that the notification was generated.
cldEventText	The full text of the notification.

Live Data SNMP Event Correlation

The CISCO-LIVEDATA-MIB notification type (cldEventNotif) defines a set of objects that are contained within a Live Data SNMP notification. Live Data notifications are "stateful." A "raise" state event indicates a problem and a "clear" state event follows when the problem resolves or after the component engages fault-tolerance mechanisms to self-heal. To maintain an accurate state at the network management station, you can write rules to automatically correlate "clear" state events to existing "raise" state events and acknowledge those notifications at the management station.

The following notification type objects are used for event correlation.

Table 15: Live Data Notification Type Objects

Object Name	Description
cldEventID	The unique numeric event message identifier for this event.
cldServerName	The fully-qualified domain name of the Cisco Live Data server that generated the notification.
cldEventAppName	The name of the Cisco Live Data functional service that generated this event.
cldEventName	The service-specific name of the Cisco LiveData event message.
cldEventState	The state of the event, either 'raise' or 'clear'. A 'raise' state event generates when an unusual or service-impacting condition occurs. A 'clear' state event generates when a prior condition is resolved.

Live Data events are numerically identified in ascending order where "raise" state events have an odd value and "clear" state events have an even value. If a "raise" state event has a matching "clear" state event, the "clear" state event has the next (higher) even value. "Single-state raise" events are "raise" state events with no matching "clear" state event. A "single-state raise" event is an error condition that typically requires manual intervention to resolve.

To match "clear" state events to existing "raise" state events, match the object cldServerName (from the same device), cldEventAppName (from the same application), and cldEventName (the same event group) value from each event. In many cases, "clear" state events map to "raise" state events. The matching "raise" state event is that event with an even valued EventID that is less than the EventID value of the "clear" state event (for example, 202 is matched to 201). There may be more than one "raise" state event associated with that "clear" state event. The "clear" state event correlates with all existing "raise" state events with the matching ServerName, EventAppName and EventName. For example, assume that the "raise" state events #301 and #303 generate, followed by the "clear" state event #304. In this case, #304 correlates to both #301 and #303, acknowledging both "raise" state events.

To understand the relationship between certain "raise and "clear" state events, see the table of Live Data events and use the "See Also" field to relate events. Each event has a textual label to identify and relate each event in the table.

Events may have certain parameters associated with the event, such as a server IP address or a service state. These parameters are expressed as "tags" within the message text. A "tag" is a name/value pair surrounded by brackets, for example: [server_address=192.168.0.1]. The parameters are expressed this way to facilitate easier (automated) parsing of event text. Because the parameters are generalized across the full set of events, a separate table describes the parameters with labels associated for easy cross-referencing of an event with the parameters used.

Live Data SNMP Parameters

This table summarizes the parameters passed into the Live Data SNMP notifications.

Table 16: Live Data SNMP Parameters

Parameter ID	Tag	Description
PARAM_JMS_URL	jms_url	URL under which JMS server is located.

Parameter ID	Tag	Description
PARAM_JMS_SUBJECT	jms_subject	Topic, or queue, about which a JMS publication is issued.
PARAM_JMS_MESSAGE	jms_message	Message (typically JSON encoded) to/from JMS broker (ActiveMQ).
PARAM_AGENT_ID	agent_id	CCE agent identifier.
PARAM_TIP_MESSAGE	tip_message_class	CCE to Live Data (TIP) message class.
PARAM_TIP_CLIENT_SEQUENCE_GROUP	tip_client_seqgrp	TIP Client Sequence Group; Live Data, low-level protocol, current group sequence number.
PARAM_TIP_SERVER_SEQUENCE_GROUP	tip_client_seqgrp	TIP Client Sequence Group; Live Data, low-level protocol, current group sequence number.
PARAM_TIP_CLIENT_SEQUENCE_NUMBER	tip_client_app_seqnum	CCE, application level, current message sequence number.
PARAM_TIP_CLIENT_APP_SEQUENCE_NUMBER	tip_server_app_seqnum	Live Data, application level, current message sequence number.
PARAM_TIP_SERVER_APP_SEQUENCE_NUMBER	tip_server_app_seqnum	CCE, application level, current message sequence number.
PARAM_ERROR_DESC	message_error	Description of error associated with encoding/decoding/processing of CCE to Live Data protocol message.
PARAM_PERIPHERAL_ID	peripheral_id	CCE peripheral ID.
PARAM_SERVER_ID	server_id	Unique identifier to server or service id in CCE (example: PG) or Live Data (example: Router Spout).
PARAM_CONNECTION_USAGE	connection_usage	Defines to which protocol, or purpose, a given connection is associated with (TOS, TIP, and so on).
PARAM_SERVER_ADDRESS	server_address	IP or hostname to a server to which Live Data is a client (example: PG).
PARAM_SERVER_URL	server_url	URL to a server to which Live Data is a client (example: ActiveMQ).
PARAM_SERVER_PORT	server_port	IP port to connect to a server to which Live Data is a client (example: PG).
PARAM_SERVER_USERNAME	server_username	Username for accessing a given server.
PARAM_DATABASE_NAME	database_name	Database name.
PARAM_OPERATION_TYPE	operation_type	Description of an operation type (example: start, stop, disable, enable, and so on).
PARAM_OPERATION_ERROR_DESC	operation_error_desc	Error description associated with a given operation failure
PARAM_CONFIGURED_LIMIT	limit	Limit (maximum, or minimum) to a given configuration element.
PARAM_CONFIGURED_PROPERTIES	properties	Configuration properties and current values.
PARAM_DATABASE_OBJECT_TYPE	db_object_type	Database object as represented in-memory.

Parameter ID	Tag	Description
PARAM_DATABASE_OBJECT_ID	db_object_id	Database object id (typically unique key).
PARAM_DATABASE_VERSION_EXPECTED	db_ver_expected	Expected database schema version.
PARAM_DATABASE_VERSION_READ	db_ver_read	Database schema version retrieved from DB.
PARAM_JMX_MBEAN_NAME	jmx_mbean_name	JMX bean name.
PARAM_STATE	state	State description of a given object (example: State Machine state transition).
PARAM_PRIOR_STATE	prior_state	Prior state of a given object (example: connection state).
PARAM_TIP_SIDE	tip_server_side	CCE server side to which Live Data is associated (example: side A or side B).
PARAM_HEARTBEAT_MISSED_COUNT	missed_hearbeats	Currently missed heartbeats during communication CCE to Live Data.
PARAM_TIME_CHANGE	tip_time_change	CCE time server adjustment in milliseconds.
PARAM_CONNECTION_STATISTICS	connection_stats	CCE to Live Data connection statistics.
PARAM_AGENT_TEAM_ID	agent_team_id	CCE Agent Team Identifier.
PARAM_MRD_ID	mrd_id	CCE Media Router Domain Identifier.
PARAM_DESCR_GENERIC	descr	Generic Description field.
PARAM_ZOOKEEPER_ZNODE	znode	Zookeeper znode name.
PARAM_VALUE	value	Value associated with a given parameter.
PARAM_INPUT	input	Input value.
PARAM_CURRENT_STATE	current_state	Current state of a given object (example: connection state).
PARAM_NEW_STATE	new_state	New state of a given object (example: connection state).
PARAM_SEQ_NUM	seqnum	Message Sequence Number.
PARAM_MESSAGE	message	Actual message in text format.
PARAM_LATENCY	latency	Latency time.
PARAM_LAST_VIRTUAL_TIMESTAMP	last_vtimestamp	Last virtual time stamp (used for Live Data cluster messaging).
PARAM_NEW_VIRTUAL_TIMESTAMP	new_vtimestamp	New virtual time stamp (used for Live Data cluster messaging).