Monitoring SNMP Health

- SNMP Overview, on page 1
- Base-Level SNMP MIB Support, on page 3
- CISCO-CONTACT-CENTER-APPS-MIB, on page 6
- Configuring the SNMP Agents, on page 25

SNMP Overview

Faults

Unified CCE has an internal, proprietary, event management system (EMS) that provides guaranteed delivery of application faults and status events from distributed nodes to the Logger component. Alarms are delivered (via MDS) to the Logger where they are stored in the database; alarms are subsequently forwarded to configured interfaces for external delivery, for instance, to an SNMP network management station (NMS) via SNMP or syslog or both.

SNMP notifications generated by the contact center application are always generated as SNMP traps from the Logger; only generic traps or traps from other subagents (such as the platform subagents provided by Hewlett Packard or IBM) are generated from Unified CCE nodes other than the Logger.

Events destined to be sent beyond just the local trace logs are stored in the local Windows Event log and then forwarded via MDS to the Logger. The Logger stores all received events in the database and then forwards them to the syslog interface (if configured). A subset of the alarms becomes SNMP notifications – only those deemed to be health-impacting are sent to SNMP notification destinations. Thus, all SNMP notifications are sent to syslog collectors; all syslog events are also stored in the Unified CCE database; every event that becomes a syslog event is stored in the Windows Event log on the server that generated the event and it is also stored in the trace log of the process that generated the event.

The following is the format of Unified CCE SNMP notifications (as defined in CISCO-CONTACT-CENTER-APPS-MIB):

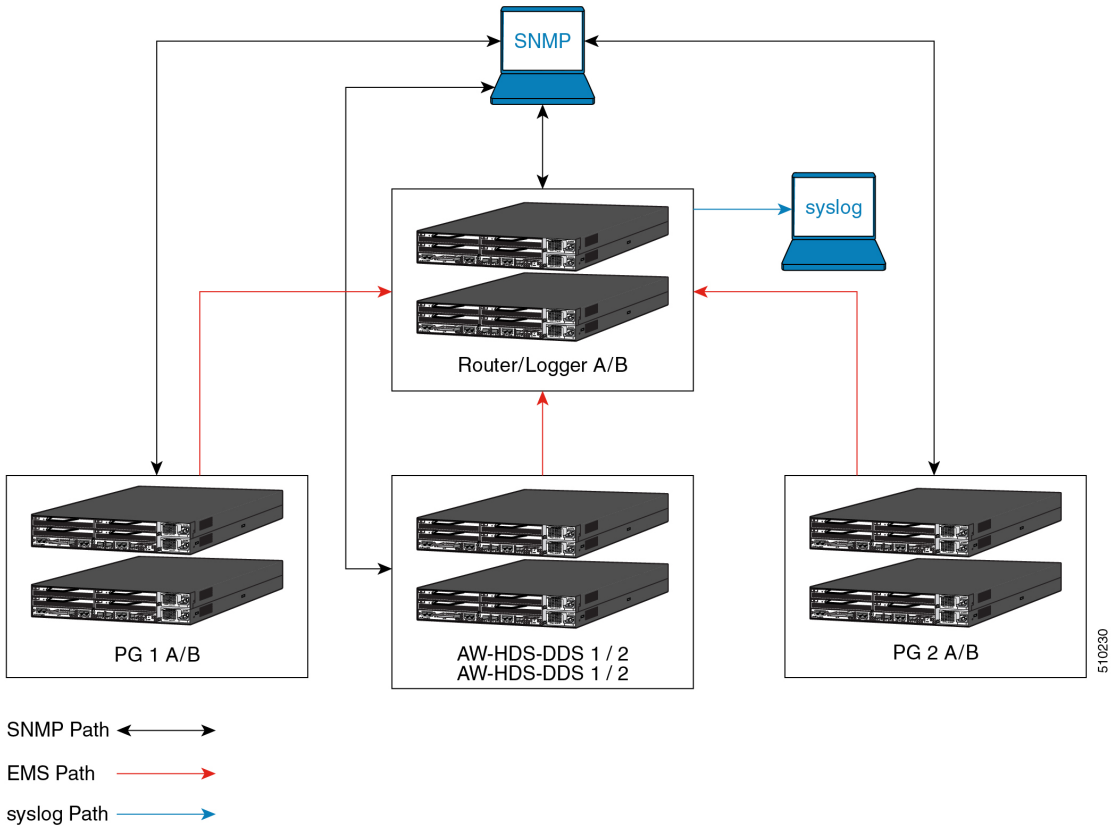
```
cccaIcmEvent NOTIFICATION-TYPE
    OBJECTS {
        cccaEventComponentId,
        cccaEventState,
        cccaEventMessageId,
        cccaEventOriginatingNode,
        cccaEventOriginatingNodeType,
        cccaEventOriginatingProcessName,
        cccaEventOriginatingSide,
```

```
        cccaEventDmpId,
        cccaEventSeverity,
        cccaEventTimestamp,
        cccaEventText
    }
```

A detailed description of each object in the notification type is found in cccaIcmEvent .

The following illustration shows the path alarms take from distributed nodes, via the Logger component to an external NMS or alarm collector.

Figure 1: CCE Event Message Flow



The red lines denote the path that alarms and event messages take within the Unified CCE event management system (EMS). These are one way from component node to the Logger (via the Router). Events are stored in the database and forwarded to the SNMP and syslog interfaces for distribution to configured collectors. Syslog is not supported on any Unified CCE nodes other than the Loggers.

The black lines denote the path of generic, or non-Unified CCE agent, SNMP notifications from device to a configured SNMP management station or stations. These are bidirectional in that SNMP management stations may poll (appropriately configured) devices for instrumentation. (Agents, by default, listen for polls on port 161.) With Unified CCE, SNMP agent processes run at a reduced priority, receiving only idle CPU time slices. As such, agent performance is throttled to ensure that a polling device cannot adversely impact the real-time Unified CCE application processes and cause a failure or impairment.

The blue lines denote the path of syslog events. Only the Loggers may generate syslog events. Syslog events are sent only to configured collectors. If no syslog collector is configured, the CW2KFeed process does not

run and no syslog events are generated. The syslog feed can be quite verbose with more than 1,000 unique events possible depending on deployment model and optional components installed.

There are over 400 configured SNMP notifications for Unified CCE.

Instrumentation

All Unified CCE servers expose instrumentation defined by the following MIBs:

- MIB-II
- CISCO-CONTACT-CENTER-APPS-MIB
- HOST-RESOURCES-MIB
- SYSAPPL-MIB

The servers may (optionally) expose platform MIBs appropriate for the vendor-originated server model; these MIBs and subagents are provided by the server vendor. If the provided subagent is a Microsoft Windows extension agent (designed to integrate with the Windows SNMP service), it seamlessly integrates with the SNMP agent implementation installed by Unified CCE.

Tables within the CISCO-CONTACT-CENTER-APPS-MIB are populated depending on which Unified CCE components are installed and configured on the server. If a certain component is not installed, that component-specific table is empty.

Base-Level SNMP MIB Support

SNMP Primary Agent

Unified CCE uses the SNMP Research International EMANATE SNMP agent infrastructure. The agent infrastructure employs typical primary/subagent architecture; the primary agent supports industry-standard MIB-II instrumentation. Subagents service polls for instrumentation from the MIBs listed here. There is also a built-in subagent adapter process that integrates Microsoft Windows extension agents, which operate using the built-in Windows primary/subagent interface. Thus, existing extension agents are seamlessly integrated into the infrastructure.

The SNMP primary agent supports SNMP v1, v2c, and v3. For SNMP v3, the primary agent supports both authentication and privacy, offering MD5 and SHA-1 for authentication and 3DES, AES-192, and AES-256 for privacy.

The primary agent listens for polls on port 161 (gets or sets) and by default, sends traps to the network management station on port 162. You can configure either port other than the well-known ports via the Unified CCE Microsoft Management Console (MMC) snap-in configuration tool.

Base Level SNMP Subagents

The SNMP subagents are processes that provide access to the application instrumentation within the server. The subagents do not interact with the management station directly. Each subagent responds to the get and set requests forwarded to them by the SNMP primary agent.

Platform MIB Support

A platform MIB/subagent is provided by the hardware vendor. This subagent provides instrumentation for low-level attributes of the specific hardware.

Host Resources MIB Subagent

The Host Resources MIB is an implementation of RFC-2790. The Host Resources MIB is a standard MIB which provides attributes common to all hosts, including but not limited to Windows- and Linux-based servers. Thus, the attributes defined are independent of the operating system, network services, or software applications. The instrumentation is focused on host memory, processors, storage devices, run-time system data, and software running on the host.

The Unified CCE Host Resources MIB subagent supports the following MIB objects/tables:

- hrSystem group
- hrMemorySize object
- hrStorage table
- hrDevice table
- hrProcessor table
- hrNetwork table
- hrDiskStorage table
- hrFS table
- hrSWRun table
- hrSWRunPerf table
- hrSWInstalledLastChange object
- hrSWInstalledLastUpdateTime object
- hrSWInstalled table

The Host Resources MIB SNMP Agent is a complete implementation of the Host Resources MIB, proposed standard RFC-1514. The Host Resources MIB is also compliant with Host Resources MIB, draft standard RFC-2790. The agent provides SNMP access to useful host information, such as the storage resources, process table, device information, and the installed software base.

Each cccaComponentElmtEntry in the cccaComponentElmtTable in the Cisco Contact Center Applications MIB corresponds to a Unified CCE managed process. The cccaComponentElmtName field contains the process executable name without the .exe extension. The cccaComponentElmtRunID field contains the process ID, which you can use as an index to the Host Resources MIB to obtain current values from the hrSWRunTable and hrSWRunPerfTable tables. The following example shows the relationship forcccaComponentElmtRunID.0.1.5 = 5384 using the results in Appendix A and a subset of the results provided by the Host Resources MIB SNMP agent on the same system:

```
cccaComponentElmtName.0.1.5 = router
cccaComponentElmtRunID.0.1.5 = 4040
cccaComponentElmtStatus.0.1.5 = active(5)
hrSWRunIndex.4040 = 4040
hrSWRunName.4040 = router.exe
```

```
hrSWRunPath.4040 = C:/icm/bin/router.exe
hrSWRunType.4040 = application(4)
hrSWRunStatus.4040 = notRunnable (3)
hrSWRunPerfCPU.4040 = 20
hrSWRunPerfMem.4040 = 6428
```

Note The implementation approach for standardized MIBs, such as the Host Resources MIB, can vary from vendor to vendor, subject to interpretation. For example, the hrSWRunStatusobject value (notRunnable) shown in the preceding example is subjective; notRunnable implies that the process is not allocated CPU cycles at the precise moment that the MIB was polled. However, any row in the hrSWRunTable indicates a process was loaded and assigned a process ID regardless of whether it is receiving CPU cycles at the moment this object value is polled. Later changes to the SNMP subagent are aligned with this assumption: any process loaded is considered running even it is not allocated CPU cycles.

MIB2

The MIB2 is defined in RFC-1213. It contains objects such as interfaces, IP, ICMP.

This MIB is fully supported on Unified CCE deployments.

SYSAPPL MIB Subagent

The System-Level Managed Objects for Applications MIB (also known as SYSAPPL MIB) is an implementation of RFC-2287. The information allows for the description of applications as collections of executables and files installed and running on a host computer. The MIB enumerates applications installed and provides application run status, associated processes and locations of executables and files on the disk.

The Unified CCE SYSAPPL-MIB subagent supports the following SYSAPPL-MIB objects/tables:

- sysApplInstallPkg table
- sysApplInstallElmt table
- sysApplElmtRun table
- sysApplPastRunMaxRows scalar
- sysApplPastRunTableRemItems scalar
- sysApplPastRunTblTimeLimit scalar
- sysApplElemPastRunMaxRows scalar
- sysApplElemPastRunTableRemItems scalar
- sysApplElemPastRunTblTimeLimit scalar
- sysApplAgentPollInterval scalar
- sysApplMap table – sysApplMapInstallPkgIndex

The SYSAPPL-MIB is a good way to capture a software inventory – applications installed on the server.

The SYSAPPL MIB supports configuration, fault detection, performance monitoring, and control of application software. It contains tables that define an application as a series of processes and services. This includes objects for applications installed on the system, elements and processes that comprise an application, and currently running and previously run applications.

CISCO-CONTACT-CENTER-APPS-MIB

The Cisco Contact Center Applications MIB contains tables of objects for the following Unified CC components:

- Router (and NICs for Unified ICM)
- Logger
- Peripheral Gateways (PGs) (and PIMs)
- Administration Server and Real-time Data Server (AWs and HDSs)
- CTI Gateways (CGs)
- Outbound Option Campaign Manager
- Outbound Option Dialers

The Cisco Contact Center Applications MIB SNMP subagent provides access to component inventory, component status, performance metrics, and links to IETF standard host-based MIBs. Appendix A, section 0 provides an example of the data provided by a Unified CC installation.

CISCO-CONTACT-CENTER-APPS-MIB Overview

The CISCO-CONTACT-CENTER-APPS-MIB is implemented on all major components of the Unified CCE solution. That is, the Router, Logger, Peripheral Gateway and the AW/HDS.

Note In prior versions, the CTI Gateway components were supported installed on separate servers; however, are now only supported co-located on the Peripheral Gateway.

The SNMP agent infrastructure is installed on all of these component servers with a subagent that serves CISCO-CONTACT-CENTER-APPS-MIB instrumentation for that server. The MIB defines a number of tables of instrumentation – one set for discovery and basic health monitoring and an additional set of tables of component-specific instrumentation. Each common component of a Unified CCE deployment has a table of objects – the Router (with a sub-table of NICs), the Logger, the Administration Server and Real-time Data Server (AW), the PG (with a sub-table of PIMs), and the CG as well as Outbound Option components, Campaign Managers on the Logger and the Dialer on the PG. The component-specific tables are only populated if that component is installed on the server.

CISCO-CONTACT-CENTER-APPS-MIB Structure

At the base, tables in the CISCO-CONTACT-CENTER-APPS-MIB are indexed by the Unified CCE instance (the instance name is a unique textual identifier that relates components that are part of the same Unified CCE system); most are secondarily indexed by the Component index. In a typical Unified CCE deployment, there is only a single instance on a single server.

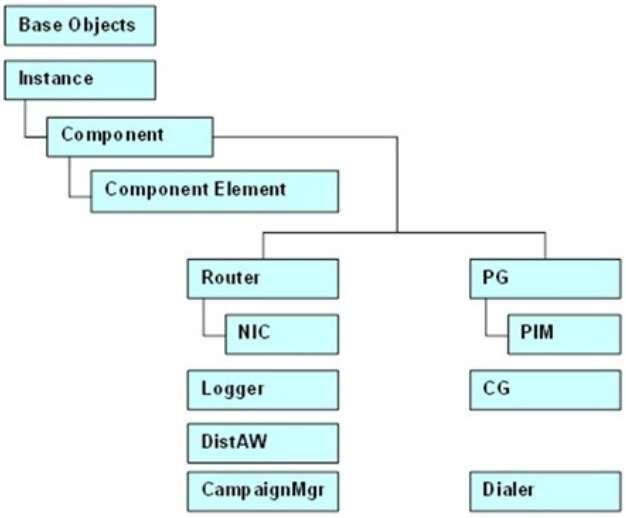
Thus, to inventory a particular server, the NMS should query the Instance table first; then query the Component table to assign components to an instance. Lastly, query the Component Elmt table for the processes associated with each component.

Using the Instance and Component indexes, the NMS can then drill down further using it to query the component-specific instrumentation for each component installed.

The component-specific table of instrumentation provides (where possible) links to dependent components that are distributed within the solution (for example, which Router a peripheral gateway communicates with or which Logger is the primary for a particular Administration Server and Real-time Data Server).

The CISCO-CONTACT-CENTER-APPS-MIB is structured as follows:

Figure 2: CISCO-CONTACT-CENTER-APPS-MIB Structure



The Instance table is indexed by the instance number – a value ranging from 1 to 25.

The Component table is indexed by Instance, and Component number that is arbitrarily assigned by the agent; the value of the Component number could change from one run period to another.

The Component Element table is indexed by Instance, Component number, and Component Element number, which is arbitrarily assigned by the agent; the value of the Component Element number could change from one run period to another.

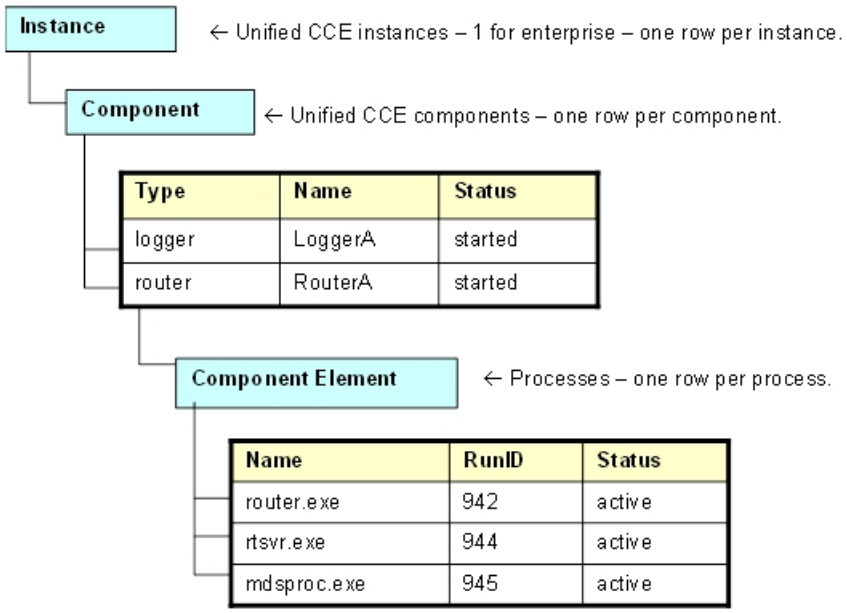
Each component-specific table of instrumentation is indexed by Component number.

From an inventory standpoint (a network management station taking inventory of the server itself), the Network Management Station (NMS) first polls the Instance table. Typically, for Unified CCE, there is only one instance. From that, the NMS polls all components that are part of this instance. Now the NMS knows what is installed on this server and can see what is running. For example, this is a Unified CCE central controller and the NMS wants to know what the inbound call rate is. With the Component entry for the Router, using the Component index of that entry, the NMS then polls the cccaRouterCallsPerSec object within the Router table (indexed by Instance number and Component index).

Additional inventory can be accomplished by drilling a little deeper. For example, assume the NMS wants to list what PIMs are installed on PG4A. Again, poll the Instance table to get the instance number. Using that, get all components for that instance. Find PG4A and using the component index for PG4A, get the PG table objects for PG4A. Then get the PIM table for PG4A that returns a list of PIMs installed.

The following figure illustrates content for the application components installed:

Figure 3: CCCA MIB – Component Inventory Example



Typically, for a Unified CCE deployment, a single instance is configured. In this case, all installed/configured components are a part of that same instance.

The Component table comprises a list of installed Unified CCE components (for example, Router and Logger).

The Component Element table is a list of installed processes that should be running.

Real-time status of each component may be monitored by polling the cccaComponentTable. The status of a Unified CCE component is derived by analyzing the collective status of each component element (the processes) as best it can.

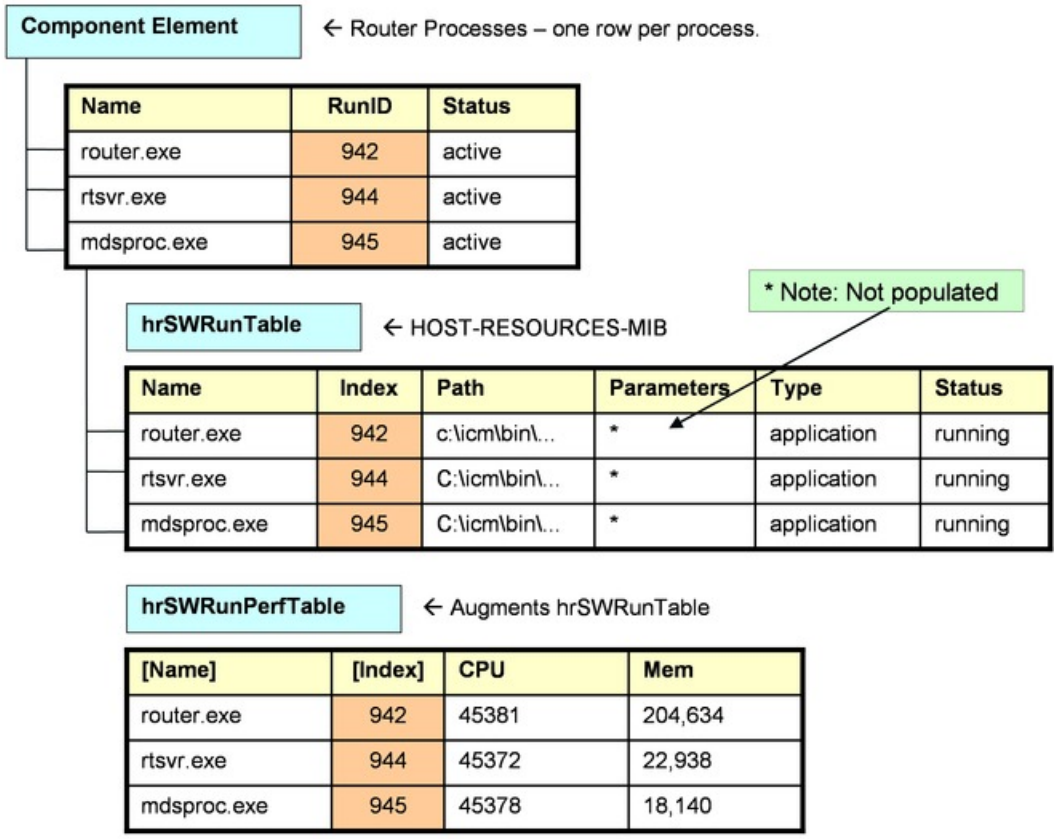
The Component Element table lists all Unified CCE processes that should be running, and exposes the (operating system) process identifier and the current status of the process.

Note The information in the figure is an example, only; there can be many more processes listed in the Component Element table.

Mapping CCCA-MIB to Standard Host MIBs

The Component Element table also provides a row-by-row mapping of Unified CCE processes to corresponding rows of instrumentation in the HOST-RESOURCES-MIB and SYSAPPL-MIB. The direct mapping is accomplished using the RunID object. Thus, rather than duplicate instrumentation already provided by the HOST-RESOURCES-MIB and SYSAPPL-MIB, these standard MIBs augment the application MIB with important process-related information.

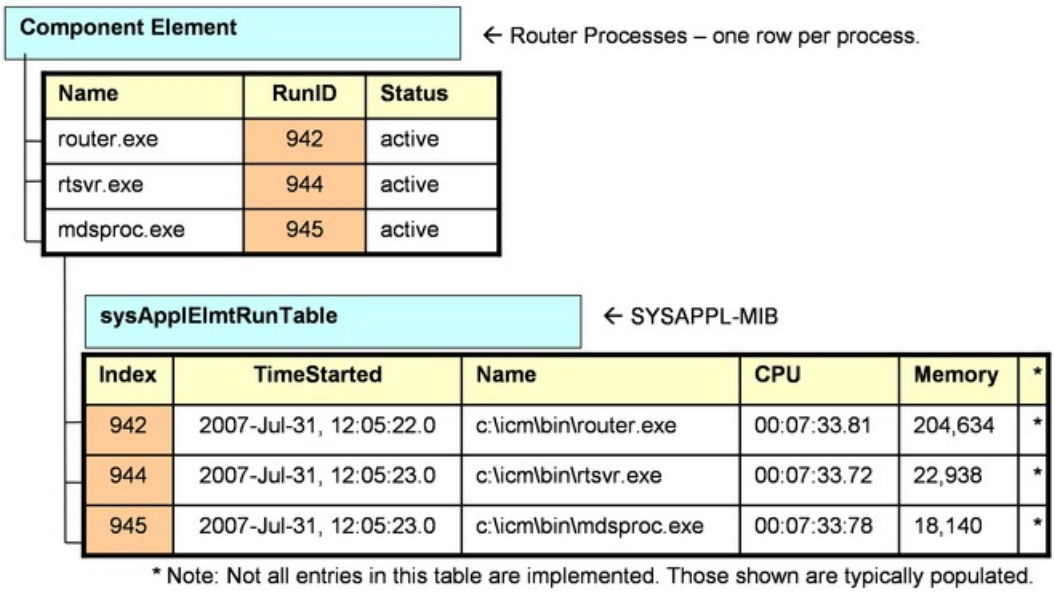
Figure 4: Mapping CCCA MIB Objects to Host MIB Objects



Using the cccaComponentElmtRunID object, a monitoring application can use this value as an index into the HOST-RESOURCES-MIB hrSWRunTable as well as the hrSWRunPerfTable (which augments it). From this, the monitoring application can acquire CPU and memory usage metrics for each process of Unified CCE. The application could also poll the remaining rows of the hrSWRunTable/hrSWRunPerfTable for processes that are consuming excessive CPU cycles and/or system memory.

You must note that there is some level of interpretation open to an implementer of a HOST-RESOURCES-MIB subagent. The implementer may decide that some columns of the table cannot be implemented or simply are not necessary. There are no strict rules. That some objects within these tables do not have values is not necessarily indicative of a failed implementation.

Figure 5: Mapping CCCA MIB to SYSAPPL MIB



If a monitoring application prefers to acquire CPU and/or memory metrics on a per-process basis, the cccaComponentElmtRunID value may also be used as an index into the SYSAPPL-MIB sysApplElmtRunTable.

The component-specific and subcomponent-specific tables include a separate table of instrumentation for each possible Unified CCE component. The list of tables includes:

- Router Table (cccaRouterTable)
    - NIC Table (cccaNicTable) – because nearly always installed on the Router, this is considered a subcomponent of the Router
- Logger Table (cccaLoggerTable)
- Distributor Admin Workstation Table (cccaDistAwTable)
- Peripheral Gateway Table (cccaPgTable)
    - Peripheral Interface Manager Table (cccaPimTable) – because always installed on the PG, this is a subcomponent of the PG
- CTI Gateway Table (cccaCgTable)
- Outbound Option Campaign Manager (cccaCampaignMgrTable)
- Outbound Option Dialer (cccaDialerTable)

A single notification object is defined in the MIB, which is used to describe the format and content of all notifications generated by Unified ICM and Unified Contact Center.

CISCO-CONTACT-CENTER-APPS-MIB Objects

The following section provides a more detailed description of each object in the CISCO-CONTACT-CENTER-APPS-MIB (CCCA MIB).

CCCA MIB Base Objects

cccaName

The fully-qualified domain name of the enterprise contact center application server.

cccaDescription

A textual description of the enterprise contact center application installed on this server. This is typically the full name of the application.

cccaVersion

Identifies the version number of the enterprise contact center application software installed on this server.

cccaTimeZoneName

The name of the time zone where the enterprise contact center application server is physically located.

cccaTimeZoneOffsetHours

The number of hours that the local time, in the time zone where the enterprise contact center application server is physically located, differs from Greenwich Mean Time (GMT).

cccaTimeZoneOffsetMinutes

The number of minutes that the local time, in the time zone where the enterprise contact center application server is physically located, differs from Greenwich Mean Time (GMT). This object is combined with the cccaTimeZoneOffsetHours object to represent the local time zone total offset from GMT.

cccaSupportToolsURL (Deprecated)

The URL for the enterprise contact center application Support Tools application server. The Support Tools application server is an optional component of the solution and offers a centralized server for diagnostic and troubleshooting tools. This application server resides on a Administration Server and Real-time Data Server host. This object offers a navigation point from the management station (assuming a web interface) can quickly access the Support Tools application server.

cccaWebSetupURL

The web setup URL object holds the URL for the enterprise contact center application setup web service. The setup web service is a component of every Unified ICM and Unified CCE server and allows for an administrator to configure parameters of the contact center application as it relates to the installation of the product itself (not to be confused with provisioning).

cccaNotificationsEnabled

The notifications enabled object allows a management station to (temporarily) disable, during run time, all outgoing contact center application notifications. This is typically done during a maintenance window where many application components are frequently stopped, reconfigured and restarted, which can generate periodic floods of notifications that are not desirable during that maintenance period. Note that this setting is persistent even after a restart of the agent; the management station must explicitly reset this object value to true to re-enable outgoing application notifications.

CCCA MIB Instance Table Objects

The instance table is a list of enterprise contact center application instances. Each instance represents a contact center application solution. A solution includes a collection of interconnected functional components (for example, a Router, a Logger and a PG), each of which perform a specific, necessary function of the contact center application.

cccaInstanceNumber

A numeric value that uniquely identifies an enterprise contact center application instance. The instance number is a user-defined value configured when the instance is created by the administrator.

cccaInstanceName

The configured textual identification for the enterprise contact center application instance.

CCCA MIB Component Table Objects

The component table is a list of enterprise contact center application functional components. A Unified CCE solution includes a collection of interconnected functional components (for example, a Router, a Logger and a Peripheral Gateway), each of which perform a specific, necessary function of the contact center application. This table enumerates and lists all contact center application functional components installed and configured on this server.

A single server is permitted to have multiple functional components of a different type, but also multiple components of the same type.

This table has an expansion relationship with the instance table; one or many entries in this table relate to a single entry in the instance table.

cccaComponentIndex

A numeric value that uniquely identifies an entry in the component table. This value is arbitrarily assigned by the SNMP subagent.

cccaComponentType

Identifies the type of enterprise contact center application functional component.

router(1), Logger(2), distAW(3), pg(4), cg(5), ctios(6)

cccaComponentName

A user-intuitive textual name for the enterprise contact center application functional component. Typically, this name is constructed using the component type text, the letter that indicates which side this component represents of a fault tolerant duplex pair and potentially a configured numeric identifier assigned to the component. For example, a Router component might be RouterB; a peripheral gateway might be PG3A. Often, this name is used elsewhere (in contact center application tools) to identify this functional component.

cccaComponentStatus

The last known status of the enterprise contact center application functional component.

Unknown (1)

The status of the functional component cannot be determined.

Disabled (2)

The functional component was explicitly disabled by an administrator.

Stopped (3)

The functional component is stopped. The component may be dysfunctional or impaired.

Started (4)

The functional component was started.

Disconnected (7)

The component is unexpectedly disconnected from a dependent component or service.

Uninitialized (8)

The component has not yet completed its initialization process.

NotRoutable (9)

The component is currently unable to make routing decisions.

CCCA MIB component element table objects

The component element table provides a list of component (operating system) services or processes that are elements of an enterprise Contact Center application functional component. Each entry identifies a single process that is a necessary element of the functional component.

This table also provides a one-to-one mapping of entries to a corresponding entry in the IETF standard host and application MIB tables. The HOST-RESOURCES and SYSAPPL MIBs expose tables that provide additional instrumentation for software and applications and for the processes that make up that software or those applications. The HOST-RESOURCES-MIB entries in hrSWRunTable and hrSWRunPerfTable and the SYSAPPL-MIB entries in sysApplElmtRunTable have a one-to-one relationship to entries in the component element table. The entries in these standard MIB tables are solely or partially indexed by the operating system process identifier (ID). The process ID is an integer value that uniquely identifies a single process that is currently running on the host. Entries in the component element table maintain its process ID; this value is used to relate the entry to a corresponding entry in the referenced tables of HOST-RESOURCES-MIB and SYSAPPL-MIB.

cccaComponentElmtIndex

A unique numeric identifier for a system process or service that is a necessary element of an enterprise Contact Center application functional component. This value is arbitrarily assigned by the SNMP subagent.

cccaComponentElmtName

The textual name of the component element, as known by the Contact Center application. The component element is an operating system process, which is a necessary element of the enterprise Contact Center application functional component. Most often, this name is the host executable filename, without the file extension.

cccaComponentElmtRunID

The operating system process ID for the process or service that is an element of this enterprise Contact Center application functional component. The component element run ID maps directly to the hrSWRunIndex value of hrSWRunTable and hrSWRunPerfTable (which augments hrSWRunTable) of the HOST-RESOURCES-MIB and the sysApplElmtRunIndex value of sysApplElmtRunTable of the SYSAPPL-MIB. This object value provides the mechanism for a one-to-one relationship between an entry in the referenced tables of these standard MIBs and an entry in the component element table.

cccaComponentElmtStatus

The last known status of a system process or service that is a necessary element of an enterprise Contact Center application functional component.

Unknown(1)

The status of the component element can’t be determined.

Disabled(2)

The component element was explicitly disabled by an administrator.

Stopped(3)

The component element is stopped; it may be dysfunctional or impaired.

Started(4)

The component element was started.

Active(5)

The component element is currently running.

Standby (6)

The functional component was started, is currently running and is the hot-standby side of a fault-tolerant duplex pair.

Disconnected (7)

The component is unexpectedly disconnected from a dependent component or service.

Uninitialized (8)

The component hasn’t yet completed its initialization process.

NotRoutable (9)

The component is currently unable to make routing decisions.

Ready (10)

This status is specific to the dialer component. The dialer component is ready to go active but waiting for the MR interface to become active.

CCCA MIB Router Table Objects

The Router table lists each enterprise contact center application Router component configured on this server. Each entry in the table defines a separate Router functional component; a single server is permitted to have multiple Router components for deployments but only has one Router for Unified CCE deployments.

The Router table has a sparse dependent relationship with the component table. The instance number acts as the primary index for the Router table to properly relate a Router component entry to the appropriate instance entry. The component index acts as the secondary index, relating the entry to the corresponding entry in the component table.

cccaRouterSide

Indicates which of the duplex pair this entry represents of an enterprise contact center application fault tolerant router functional component. The Router side value is either 'A' or 'B'. For simplex configurations, the Router side value defaults to 'A'.

cccaRouterCallsPerSec

Indicates the current inbound call rate; that is, the calculated number of inbound calls per second.

cccaRouterAgentsLoggedOn

The number of contact center agents currently managed by the enterprise contact center application. This does not necessarily represent the number of contact center agents that can receive routed calls, but rather the number of agents for which the application is recording statistical information.

cccaRouterCallsInProgress

Indicates the current number of active (voice) calls being managed by the enterprise contact center application. The calls are in various states of treatment.

cccaRouterDuplexPairName

The hostname of the duplex pair (for example, the other side) server of an enterprise contact center application fault tolerant Router component. If this component is not part of a duplex pair (for example, simplex), the object value is the null string.

cccaRouterNicCount

The number of network interface controllers configured and enabled for this enterprise contact center application Router functional component. There is an imposed architectural limit of 32 configured NICs per Router.

cccaRouterCallsInQueue

The Router calls in queue object indicates the total number of calls queued in all network Voice Response Units (VRUs), from the Router's perspective, including those calls that are in the process of transferring to the VRU for queuing.

cccaRouterAppGwEnabled

The Router application gateway enabled object indicates whether an application gateway is configured and a part of this contact center application deployment. An application gateway provides an external interface to business back-end systems that may be used as external input to call scripting logic, or, that logic which controls how a customer call is handled (routed).

cccaRouterDBWorkerEnabled

The Router database worker enabled object indicates whether a database worker process was configured and is a part of this contact center application deployment. A database worker provides an interface to an external database from which data may be retrieved and used as input to call scripting logic, or, that logic which controls how a customer call is handled (routed).

cccaRouterPGsEnabledCount

The Router PGs enabled count object holds the number of PGs that were enabled for this Router; during usual operation, this is the number of PGs that connect to this Router functional component. There is an imposed architectural limit of 150 peripheral gateways per deployment.

cccaRouterPublicHighAddr

The Router public high address object holds the address of the local high-priority interface of this Router functional component to the public network. The public network interface is exposed outside the realm of the Unified Contact Center application and is used for the transfer of data between this Router and other functional components of the contact center deployment. This interface is reserved for high-priority messages; network prioritization is typically configured for this interface to ensure a level of quality of service.

cccaRouterPublicNonHighAddr

The Router public non-high address object holds the address of the local interface of this Router functional component to the public network that is used for best effort priority messages. The public network interface is exposed outside the realm of the Unified CC application and is used for the transfer of data between this Router and other functional components of the deployment. This interface is used for normal-priority messages.

cccaRouterPrivateHighAddr

The Router private high address object holds the address of the local high-priority interface of this Router functional component to the private network. The private network interface is used exclusively by the Unified Contact Center application for the transfer of synchronization data between duplexed pairs and for the transfer of application data from the Router to the Logger. This interface is reserved for high-priority messages and as much as 90% of the available network bandwidth is allocated to this interface.

cccaRouterPrivateNonHighAddr

The Router private non-high address object holds the address of the local interface of this Router functional component to the private network that is used for best effort priority messages. The private network is used exclusively by the Unified Contact Center application for the transfer of synchronization data between duplexed pairs and for the transfer of application data from the Router to the Logger. This interface is used for normal-priority messages.

CCCA MIB NIC Table Objects

The NIC table lists the enterprise contact center application network interface controllers enabled on this Router functional component.

The NIC table has an expansion dependent relationship with the Router table. There may be one or more NIC entries associated with a single Router entry. The instance index acts as the primary index and the component index a secondary index. This indexing method ensures that NIC entries are properly related to its parent Router and to the appropriate instance. The SNMP agent arbitrarily assigns the NIC index when each NIC table entry is created.

cccaNicIndex

A value that uniquely identifies an entry in the network interface controller table. The value of this object is arbitrarily assigned by the SNMP subagent.

cccaNicType

Indicates to which telephony network this NIC functional component provides an interface.

cccaNicStatus

The last known status of the enterprise contact center application network interface controller functional component.

CCCA MIB Logger Table Objects

The Logger table lists the enterprise contact center application Logger functional components installed and enabled on this server.

The Logger table has a sparse dependent relationship with the component table. The instance number acts as the primary index for the Logger table to properly relate a Logger component entry to the appropriate instance entry. The component index acts as the secondary index, relating the entry to the corresponding entry in the component table.

cccaLoggerSide

Which of the duplex pair this entry represents, of an enterprise contact center application fault tolerant Logger functional component. The Logger side value is either 'A' or 'B'. For simplex configurations, the Logger side value defaults to 'A'.

cccaLoggerType

Which type of enterprise contact center application Logger, is installed on this server. The Logger type varies based on the configuration of the contact center solution.

cccaLoggerRouterSideAName

The hostname of the side 'A' Router that this enterprise contact center application Logger functional component is associated. The Logger component must be connected to a Router that is part of the same instance.

cccaLoggerRouterSideBName

The hostname of the side 'B' Router that this enterprise contact center application Logger functional component is associated. The Logger component must be connected to a Router that is part of the same instance.

cccaLoggerDuplexPairName

The hostname of the duplex pair (for example, the other side) server of an enterprise contact center application fault tolerant Logger component. If this component is not part of a duplex pair (for example, simplex), the object value is the null string.

The Logger connects to its duplex pair via a private' interface a closed subnet that guarantees a quality of service level that does not impact the performance of the contact center application. This private subnet is not accessible by the management station.

cccaLoggerHDSReplication

Indicates whether the Logger component replicates data to a Administration Server, Real-time and Historical Data Server, and Detail Data Server. If true, the Logger feeds historical data at regular intervals to the HDS for long-term storage. In this configuration, administrator reports are generated by accessing data from the HDS rather than the Logger in order to remove the performance impact of reporting on the Logger.

cccaLoggerAvgDBWriteTime

The Logger average database write time expresses the average amount of time, in 100 nanosecond units, required to write data to a table in the central controller database. This value represents the average time per write of the write operations that occurred in the past second. This object is a good indicator of contention for database access.

CCCA MIB Administration Server and Real-Time Data Server Table Objects

The Administration Server and Real-time Data Server table lists the enterprise contact center application Administration Server and Real-time Data Server functional components installed and enabled on this server.

The Administration Server and Real-time Data Server table has a sparse dependent relationship with the component table. The instance number acts as the primary or the Administration Server and Real-time Data Server table to properly relate an Administration Server and Real-Time Data Server component entry to the appropriate instance entry. The component index acts as the secondary index, relating the entry to the corresponding entry in the component table.

cccaDistAwSide

Which of the duplex pair this entry represents, of an enterprise contact center application fault tolerant distributor administrator workstation functional component. The Administration Server and Real-time Data Server side value is either A or B. For simplex configurations, the Administration Server and Real-time Data Server side value defaults to A.

cccaDistAwType

Which type of enterprise contact center application distributor administrator workstation, is installed on this server. The Administration Server and Real-time Data Server type varies based on the configuration of the contact center solution.

cccaDistAwAdminSiteName

A user-defined textual name that uniquely identifies the location or the configuration of the Administration Server and Real-time Data Server component.

cccaDistAwRouterSideAName

The hostname of the side A Router that this enterprise contact center application Administration Server and Real-time Data Server functional component is associated. The the Administration Server and Real-time Data Server component must be connected to a Router that is part of the same instance. If the side B Router is the active Router and a failure occurs, the side A Router then immediately assumes the role. In this case, the Administration Server and Real-Time Data Server lose their connection to the side B Router and thus use this object value to connect to the side A Router.

cccaDistAwRouterSideBName

The hostname of the side B Router that this enterprise contact center application Administration Server and Real-time Data Server functional component is associated. The Administration Server and Real-time Data Server component must be connected to a Router that is part of the same instance. If the side A Router is the active Router and a failure occurs, the side B Router then immediately assumes the role. In this case, the Administration Server and Real-Time Data Server lose their connection to the side A Router and thus use this object value to connect to the side B Router.

cccaDistAwLoggerSideAName

The hostname of the side A Logger that this enterprise contact center application Administration Server and Real-time Data Server functional component is associated. The Administration Server and Real-time Data Server component must be connected to a Logger that is part of the same instance. If the side B Logger is the active Logger and a failure occurs, the side A Logger then immediately assumes the role. In this case, the Administration Server and Real-time Data Server lose their connection to the side B Logger and thus use this object value to connect to the side A Logger.

cccaDistAwLoggerSideBName

The hostname of the side B Logger that this enterprise contact center application Administration Server and Real-time Data Server functional component is associated. The Administration Server and Real-time Data Server component must be connected to a Logger that is part of the same instance. If the side A Logger is the active Logger and a failure occurs, the side B Logger then immediately assumes the role. In this case, the distributor AW loses its connection to the side A Logger and use this object value to connect to the side B Logger.

cccaDistAwDuplexPairName

The hostname of the duplex pair (for example, the other side) server of an enterprise contact center application fault tolerant Administration Server and Real-time Data Server component. If this component is not part of a duplex pair (for example, simplex), the object value is the null string.

cccaDistAwHDSEnabled

Indicates whether this enterprise contact center application distributor administrator workstation has a historical database server (HDS) configured and enabled. If so, this Administration Server and Real-time Data Server receive replicated data from the Logger at periodic intervals and add the data to the HDS. Client administrator workstations generate reports based on the data in this HDS.

cccaDistAwWebViewEnabled (Deprecated)

Indicates whether this enterprise contact center application distributor administrator workstation has a web-based reporting server (WebView) configured and enabled. Having WebView configured and enabled does not imply that a historical database server is also present on this server; the data may be accessed by the WebView server from a database on a different host.

cccaDistAwWebViewServerName (Deprecated)

The server (universal naming convention [UNC]) name of the server where the enterprise contact center application database resides. This database holds the real-time and/or historical data that is requested when generating reports..

CCCA MIB Peripheral Gateway Table Objects

The PG table lists the enterprise contact center application PG functional components installed and enabled on this server.

The PG table has a sparse dependent relationship with the component table. The instance number acts as the primary index for the PG table to properly relate a PG component entry to the appropriate instance entry. The component index acts as the secondary index, relating the entry to the corresponding entry in the component table.

cccaPgNumber

A user-defined numeric identifier for this enterprise contact center application peripheral gateway.

cccaPgSide

Which of the duplex pair this entry represents of an enterprise contact center application fault tolerant peripheral gateway functional component. The PG side value is either 'A' or 'B'. For simplex configurations, the PG side value defaults to 'A'.

cccaPgRouterSideAName

The hostname of the side A Router that this enterprise contact center application peripheral gateway functional component is associated. The peripheral gateway component must be connected to a Router that is part of the same instance. If the side B Router is the active Router and a failure occurs, the side A Router then immediately assumes the role. In this case, the peripheral gateway loses its connection to the side B Router and thus use this object value to connect to the side A Router.

cccaPgRouterSideBName

The hostname of the side B Router that this enterprise contact center application peripheral gateway functional component is associated. The peripheral gateway component must be connected to a Router that is part of the same instance. If the side A Router is the active Router and a failure occurs, the side B Router then immediately assumes the role. In this case, the peripheral gateway loses its connection to the side A Router and thus use this object value to connect to the side B Router.

cccaPgDuplexPairName

The hostname of the duplex pair (for example, the other side) server of an enterprise contact center application fault tolerant peripheral gateway component. If this component is not part of a duplex pair (for example, simplex), the object value is the null string.

cccaPgPimCount

The number of peripheral interface managers configured and enabled for this enterprise contact center application peripheral gateway functional component.

cccaPgCallsInProgress

The call in progress object shows the number of calls that are currently active and being managed or monitored by this peripheral gateway.

cccaPgAgentsLoggedOn

The agents logged in object shows the number of agents associated with this peripheral gateway that are currently logged in and are being managed or monitored by this peripheral gateway.

cccaPgAgentsReady

The agents ready object shows the number of agents associated with this peripheral gateway that are currently logged in and in a 'Ready' state, for example,, ready to receive calls.

cccaPgAgentsTalking

The agents talking object shows the number of agents associated with this peripheral gateway that are currently logged in and taking a call (in a 'Talking' state).

cccaPgID

The PG identifier is a unique numeric identifier for this enterprise contact center application peripheral gateway. The identifier is assigned by the contact center application.

CCCA MIB Peripheral Interface Manager Table Objects

The PIM table lists the enterprise contact center application PIM configured and enabled on this Peripheral Gateway functional component.

The PIM table depends on both the instance table and the PG table; the instance index acts as the primary index and the PG index a secondary index. This indexing method ensures that PIM entries are properly related to its parent PG and to the appropriate instance.

The PIM table has an expansion dependent relationship with the PG table. There may be one or more PIM entries associated with a single PG entry. The instance index acts as the primary index and the component index a secondary index. This indexing method ensures that PIM entries are properly related to its parent PG and to the appropriate instance. The SNMP agent assigns the PIM number, based upon the configuration, when each PIM table entry is created.

cccaPimNumber

The numeric identifier for this enterprise contact center application PIM. This object value is a user-defined numeric value and is limited to a maximum of 32 because this is the maximum number of PIMs supported on a single peripheral gateway.

cccaPimPeripheralName

The user-defined textual name of the enterprise contact center application PIM. This name uniquely identifies the PIM.

cccaPimPeripheralType

The type of the enterprise contact center application PIM, for example, the brand name and model of the ACD, private branch exchange (PBX), or VRU.

cccaPimStatus

The last known status of the enterprise contact center application peripheral interface manager functional component.

cccaPimPeripheralHostName

The hostname or IP address of the peripheral (the PBX, ACD, or VRU) to which the enterprise contact center application PIM is connected. If there are multiple interfaces to the peripheral, each hostname or IP address is separated by a comma.

CCCA MIB CTI Gateway Table Objects

The CG table lists the enterprise contact center application computer telephony integration (CTI) gateway functional components installed and enabled on this server.

The CTI gateway table has a sparse dependent relationship with the component table. The instance number acts as the primary index for the CTI gateway table in order to properly relate a CTI gateway component entry to the appropriate instance entry. The component index acts as the secondary index, relating the entry to the corresponding entry in the component table.

cccaCgNumber

A numeric identifier for this enterprise contact center application CTI Gateway. This is a user-defined numeric value and may not be identical to the table index.

cccaCgSide

Which of the duplex pair this entry represents of an enterprise contact center application fault tolerant CTI gateway functional component. The CG side value is either 'A' or 'B'. For simplex configurations, the CG side value defaults to 'A'.

cccaCgPgSideAName

The hostname of the side ‘A’ PG that this enterprise contact center application CTI gateway (CG) functional component is associated. The CG component must be connected to a PG that is part of the same instance. If the side ‘B’ PG is the active PG and a failure occurs, the side ‘A’ PG then immediately assumes the role. In this case, the CG loses its connection to the side ‘B’ PG and thus use this object value to connect to the side ‘A’ PG.

cccaCgPgSideBName

The hostname of the side ‘B’ peripheral gateway (PG) that this enterprise contact center application CTI gateway (CG) functional component is associated. The CG component must be connected to a PG that is part of the same instance. If the side ‘A’ PG is the active PG and a failure occurs, the side ‘B’ PG then immediately assumes the role. In this case, the CG loses its connection to the side ‘A’ PG and thus use this object value to connect to the side ‘B’ PG.

cccaCgDuplexPairName

The hostname of the duplex pair (for example, the other side) server of an enterprise contact center application fault tolerant CTI gateway component. If this component is not part of a duplex pair (for example, simplex), the object value is the null string.

cccaCgOpenSessions

The CG open sessions object indicates the number of sessions (connections) that were established between the CTI Gateway and CTI clients. These are active sessions that are functioning usually.

cccaCgOtherSessions

The CG other sessions objects indicates the total number of sessions (connections) between the CTI Gateway and CTI clients that are not usual, open/active sessions. This includes sessions that are 'opening' (not yet established and initialized), session that are 'closing' (connections being torn down) as well as sessions that are in an 'unknown' state and sessions that have failed. While this object value fluctuates from time to time, it stabilizes during usual operation. A steadily increasing value indicates a problem that should be investigated.

cccaCgID

The CG number is a unique numeric identifier for this enterprise contact center application CTI gateway. The identifier is assigned by the contact center application.

CCCA MIB Outbound Option Campaign Manager Table Objects

The Campaign Manager table lists the enterprise contact center application Outbound Option Campaign Manager functional components installed and enabled on this server. In virtually all single-instance enterprise deployments, the Campaign Manager is coresident with the Logger.

The Campaign Manager table has a sparse dependent relationship with the component table. The instance number acts as the primary index for the Campaign Manager table to properly relate a Campaign Manager component entry to the appropriate instance entry.

The component index acts as the secondary index, relating the entry to the corresponding entry in the component table.

The SNMP agent constructs the Campaign Manager table at startup. Because you can only configure Campaign Manager components while the enterprise contact center application is stopped, Campaign Manager table entries cannot be added to or deleted from the table either by the agent or the management station when the application is running. The agent updates the values of Campaign Manager entry objects as their values change when the application is running. All objects in this table are read-only to the management station.

Each Campaign Manager entry represents an enterprise contact center application Campaign Manager server functional component configured on the server. The Campaign Manager component, which resides on the Unified CCE Logger, is responsible for:

- Managing when a campaign runs
- Maintaining system and Dialer configurations
- Making decisions about which contact records to retrieve from a campaign based on configurable query rules and then delivering those contact records to Dialers
- Distributing configuration data to the import process and all available Dialers in the system
- Collecting real-time and historical data and sending it to the Router for subsequent storage and distribution
- Managing the Do Not Call list, ensuring no numbers on it are sent to the Dialers

The objects in each campaign manager entry provide configuration, performance, and component status information.

cccaCampaignMgrDbUtilization

The campaign manager and Import processes share a private database on the Logger. The campaign manager database utilization object shows what percentage of allocated space in the database is currently utilized. An administrator should monitor this object when its value exceeds 80 percent.

cccaCampaignMgrQueueDepth

The campaign manager is a multithreaded process. One main dispatch thread is involved in most processing. The queue depth object indicates how many messages are queued to this internal dispatch thread.

cccaCampaignMgrAvgQueueTime

The campaign manager is a multithreaded process; however, there is one main dispatch thread that is involved in most message processing. The average queue time object shows the average amount of time a message spends in the main dispatch thread queue awaiting processing (in milliseconds).

cccaCampaignMgrActiveDialers

The campaign manager process feeds several Dialer components which manage the dialing of customers for outbound campaigns. The active Dialers counter indicates how many Dialers are currently registered to this campaign manager.

CCCA MIB Outbound Option Dialer Table Objects

The Dialer table lists each enterprise contact center application Outbound Option Dialer component configured on this server. Each entry in the table defines a separate Dialer functional component.

The Dialer table has a sparse dependent relationship with the component table. The instance number acts as the primary index for the Dialer table to properly relate a Dialer component entry to the appropriate instance entry. The component index acts as the secondary index, relating the entry to the corresponding entry in the component table.

The SNMP agent constructs the Dialer table at startup. Because you can only configure a Dialer while the enterprise contact center application is stopped, Dialer table entries cannot be added to or deleted from the table either by the agent or the management station when the application is running. The agent updates Dialer entry objects as their values change when the application is running. All objects in this table are read-only to the management station.

Each Dialer entry represents an enterprise contact center application Outbound Option Dialer functional component configured on the server. The Dialer component maximizes the resources in a contact center by dialing several customers per agent. The Dialer component resides on the peripheral gateway (PG) server, where it does the following:

- Dials customers
- Reserves agents
- Performs call classification
- Calculates agent availability
- Keeps Outbound Dialing at a level where the abandon rate is below the maximum allowed abandon rate

The objects in the Dialer entry provide information about dependent components, performance metrics, and port usage.

cccaDialerCampaignMgrName

The Dialer campaign manager name object holds the hostname or IP address of the Outbound Option Campaign Manager to which this Dialer is associated. The Dialer connects to the campaign manager to exchange data related to an Outbound Dialing campaign.

cccaDialerCampaignMgrStatus

The Dialer campaign manager status indicates the current connection status between this Dialer and the Outbound Option Campaign Manager component, which is coresident with the Logger.

cccaDialerCtiServerAName

The Dialer CTI server A name object holds the hostname or IP address of the contact center application CTI Server side A functional component, which this Dialer depends. The Dialer connects to the CTI Server to monitor skill group statistics (to choose an agent) and runs call control after an available agent is selected.

cccaDialerCtiServerBName

The Dialer CTI server B name object holds the hostname or IP address of the contact center application CTI Server side B functional component, which this Dialer depends. The Dialer connects to the CTI Server to monitor skill group statistics (to choose an agent) and runs call control after an available agent is selected.

cccaDialerCtiServerStatus

The Dialer CTI server status indicates the current connection status between this Dialer and the active CTI server component.

cccaDialerMediaRouterStatus

The Dialer media Router status indicates the current connection status between this Dialer and the Media Routing (MR) Peripheral Interface Manager (PIM) component. The Dialer uses the MR PIM interface to reserve an available agent as a recipient for a dialed customer call.

cccaDialerQueueDepth

The Dialer is a multithreaded process that communicates between threads using inter-thread messaging. The queue depth object indicates how many messages are currently queued for the main dispatch thread. When this object is used in combination with the average queue time object, message processing performance can be gauged.

cccaDialerAvgQueueTime

The Dialer is a multithreaded process that communicates between threads using messaging. One main dispatch thread is involved in most message processing. The average queue time shows the average amount of time (in milliseconds) that a message spent in the queue before being de-queued processing. When this object used in combination with the queue depth object, message processing performance can be gauged.

cccaDialerTalkingAgents

For an agent campaign, the Dialer places calls to customers and transfers those customer calls to agents. The talking agents object indicates how many agents are currently talking in the monitored campaign skill group.

cccaDialerCallAttemptsPerSec (Deprecated)

The call attempts per second object tracks how many calls the Dialer is placing per second, rounded to the nearest integer. If the dialing rate is too high, it can result in network congestion on the voice network, which can result in inefficient dialing.

cccaDialerConfiguredPorts

The Dialer configured ports object is a count of the total number of ports that are configured for placing calls to customers and for transferring calls to agents during outbound calling campaigns. During usual operation, the Dialer configured ports object value is equal to a sum of busy and idle ports.

cccaDialerBusyCustomerPorts

The Dialer busy customer ports object is a count of the number of ports currently in use for customer calls. The port is the unit on the Dialer that places calls to reserve agents and to contact customers.

cccaDialerBusyReservationPorts

The Dialer busy reservation ports object tracks how many ports are currently busy reserving agents. The port is the unit on the Dialer that places calls to reserve agents and to contact customers.

cccaDialerIdlePorts

The Dialer idle ports object is a count of the number of ports that are currently idle, for example, there are no calls to customers or to agents using these ports and they are available to the Dialer for placing new calls.

cccaDialerBlockedPorts

The Dialer blocked ports object is a count of the number of ports that are currently unusable for placing calls. A blocked port may be an impaired or inoperable port or one that has a ‘stuck’ call that was not dropped. A ‘stuck’ call is a call that is identified by the application as exceeding a duration threshold.

Configuring the SNMP Agents

Installation Prerequisites for SNMP Support

Unified CCE SNMP support is automatically installed during setup. No extra steps must be taken during setup for SNMP support to be enabled. However, you must install Microsoft Windows SNMP optional components on the Unified CCE servers for any SNMP agents to function.

Note Install the appropriate Microsoft Windows SNMP components before you install any Unified CCE components that require SNMP monitoring. Instructions for installing the Microsoft Windows SNMP component are below.

You require the Microsoft SNMP components are required for Cisco SNMP support. However, the Microsoft Windows SNMP service is disabled as part of the Unified ICM setup and is replaced by the Cisco Contact Center SNMP Management service to process SNMP requests in its place. The Cisco Contact Center SNMP Management service provides for more sophisticated SNMP capabilities than the standard Microsoft SNMP Service.

Note The AppInfo feature provided by the VMware tools has to be disabled. For instructions to disable the AppInfo feature, see the VMware documentation.

SNMP Agent Configuration

While all SNMP components are installed and enabled by default, the device is not manageable via an NMS until you properly configure the solution. You configure the Cisco Contact Center SNMP solution using a Microsoft Management Console (MMC) snap-in. There are many functions of a Windows-based server that are configured using an MMC snap-in so the interface is familiar.

Add Cisco SNMP Agent Management Snap-In

Before you begin

To configure the Cisco SNMP agents, you must first add the Cisco SNMP Agent Configuration snap-in to a Microsoft Management Console. You can then change and save SNMP agent settings. To add the snap-in:

Procedure

Step 1 From the Start menu select **Run**.

Step 2 In the Start box type in `mmc /32` and press **Enter**.

Step 3 From the Console, select **File** > **Add/Remove Snap-in**

A new window appears.

Step 4 From the **Standalone** tab, verify **Console Root** is selected in the **Snap-ins added to:** field and click **Add**.

Step 5 In the Add Snap-in window scroll down and select **Cisco SNMP Agent Management**.

Step 6 In the Add Snap-in window click **Add**.

Step 7 In the Add Snap-in window click **Close**.

Step 8 Click **OK** in the Add/Remove Snap-in window.

The Cisco SNMP Agent Management snap-in is now loaded in the console.

Save Snap-In View

After you load the Cisco SNMP Agent Management MMC Snap-in, you can save that console view to a file (with an .MSC file extension) that you can launch directly instead of repeatedly adding the Snap-in to a new MMC console view.

Procedure

Step 1 Click **Console** > **Save As**.

Save As dialog appears.

Step 2 Enter a memorable file name.

Example:

```
Cisco SNMP Agent Management.msc
```

Retain the .msc file extension.

Step 3 Click **OK** to save the file to the desired location.

The Administrative Tools (start) menu is the default location, which makes it conveniently available for later access via the Start menu.

Configure Community Names for SNMP v1 and v2c

If you are using SNMP v1 or v2c you must configure a Community Name so that Network Management Stations (NMSs) can access the data provided by your server. These names are left blank during installation for security reasons.

SNMP Community Names are used to authenticate data exchange of SNMP information. An NMS can exchange SNMP information only with servers that use the same Community Name.

To configure the Community Name for SNMP v1 and v2c:

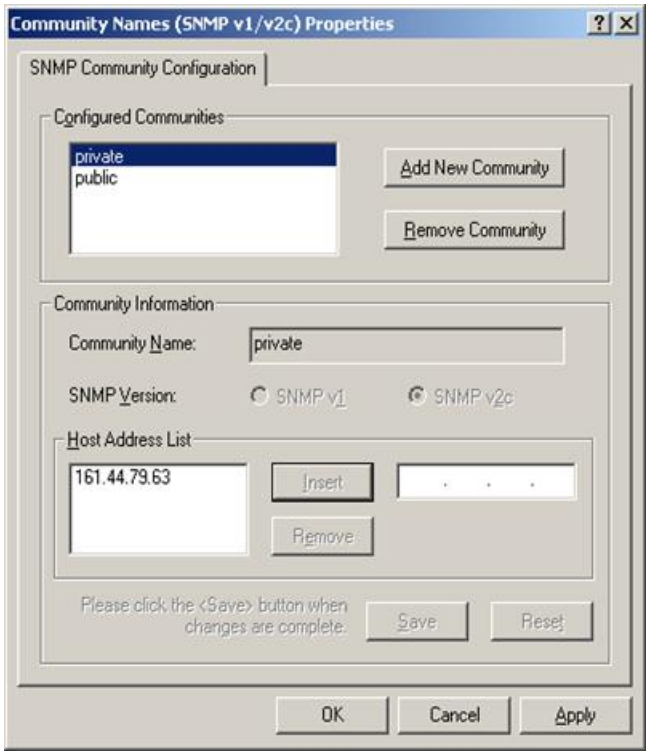
Procedure

Step 1 Expand **Cisco SNMP Agent Management** in the left pane of the MMC snap-in.

Step 2 Highlight **Community Names (SNMP v1/v2c)** in the left pane under Cisco SNMP Agent Management Community Name, SNMP Version, and Restricted Access columns appear in the right pane.

Step 3 Right-click the white space in the right pane and choose **Properties**.
A dialog box appears.

Figure 6: SNMP Community Name Configuration Dialog



Step 4 Click **Add New Community**.

Step 5 In the dialog box, under **Community Information**, provide a community name.

Step 6 Select the **SNMP Version** by selecting the radio box for SNMP v1 or SNMP V2c.

Step 7 Optionally, enter one or more IP addresses in the IP Address entry field (containing “dots”) and click **Insert** to enable the access solely for this community from the NMS with the IP Address provided.

This applies to both polling and traps.

Step 8 Click **Save**.

The community name appears in the Configured Communities section at the top of the dialog box.

Note
You can remove the community name by highlighting the name in the Configured Communities section and clicking Remove Community.

Step 9 Click **OK**.

Configure User Names for SNMP v3

If you are using SNMP v3 you must configure a User Name so that Network Management Stations (NMSs) can access the data provided by your server. By default, these names are left blank for security reasons.

Procedure

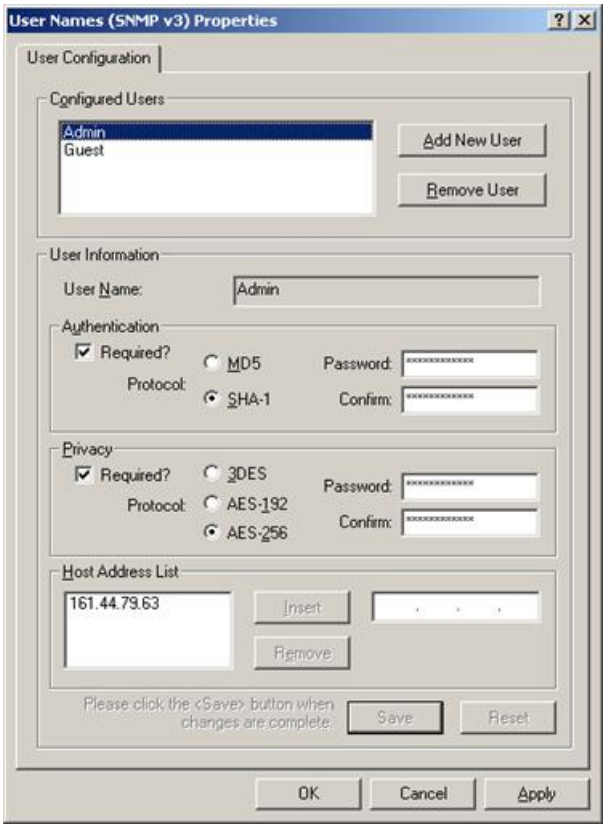
Step 1 Expand **Cisco SNMP Agent Management** in the left pane of the MMC snap-in.

Step 2 Highlight **User Names (SNMP v3)** in the left pane under Cisco SNMP Agent Management.

User Name, Authentication, Privacy, and Restricted Access columns appear in the right pane.

Step 3 Right-click the white space in the right pane and choose **Properties**.
User Names (SNMP v3) Properties dialog box appears.

Figure 7: SNMP User Name Configuration Dialog Box



Step 4 Click **Add User**.

Step 5 Enter user name in **User Configuration** text box .

Step 6 (Optional) Check **Required?** under Authentication to use SNMP v3 authentication.

a) choose an authentication protocol

b) Enter and confirm a password.

Note
This setting encrypts the password information as it is sent over the network. You must use these settings on your NMS to access SNMP data from this server.

Step 7 (Optional) Check **Required?** under Privacy to use SNMP v3 privacy.

a) Choose an encryption type.
b) Enter and confirm a password.

Note
This setting encrypts all SNMP information as it is sent over the network. If privacy is configured, authentication is required, but you can configure authentication without configuring privacy. You must use these settings on your NMS to access SNMP data from this server.

Step 8 (Optional) Enter one or more IP addresses in the IP Address entry field (containing dots) and click **Insert** to enable access solely from the NMS with the IP Address provided.

This applies to both polling and traps.

Step 9 Click **Save**
The new User Name appears in the **Configured Users** section at the top of the dialog box.

Note
You can remove the user by highlighting the name in the Configured Users section and clicking **Remove User**.

Step 10 Click **OK**.

Configure General Information Properties

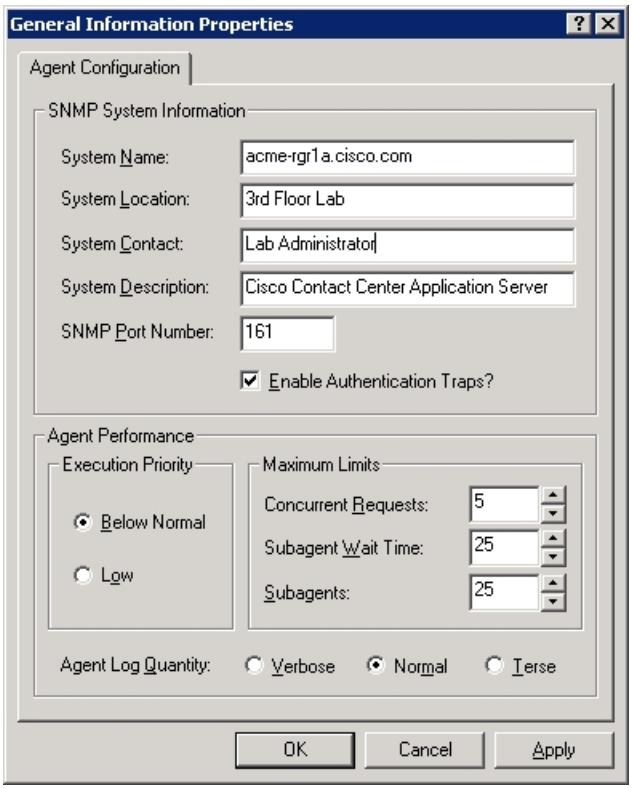
You can configure general information properties for Cisco SNMP within the Cisco SNMP Agent Management Snap-in.

Procedure

Step 1 Highlight **General Information** in the left pane under Cisco SNMP Agent Management. Attribute, Value, and Description columns appear in the right pane.

Step 2 Right-click the white space in the right pane and choose **Properties**.
General Information Properties dialog appears.

Figure 8: SNMP General Information Configuration Dialog Box



Step 3 Change the following properties in the **SNMP System Information** section of the General Information Properties dialog box.

Table 1: SNMP General Information Properties

Property	Description
System Name	The fully qualified domain name of the system. If empty, this automatically fills.
System Location	The physical location of the server itself, for example, Building 5, Floor 3, Room 310.
System Contact	The name, email address and/or telephone number of the system administrator or point of contact that should be notified to help resolve a problem with the server.
System Description	A brief description of this server, to include the primary application running on the server.
Number	The port number to be used to access/poll the device. The default port for SNMP polling is UDP 161; if you NMS uses a different port, enter the desired port number here.
Enable Authentication Traps	Check if you wish to enable Authentication Traps. When an NMS attempts to poll this device with inappropriate authentication credentials (for example, wrong community name), the device generates a failed authentication trap.

Note
The notifications are explained in `<INSTALL_DRIVE>/icm/snmp/CCA-Notifications.txt`.

Step 4 (Optional) Change Windows Execution Priority of Cisco SNMP agents in **Agent Performance** section under **Execution Priority**.

The default is Below Normal. You can further lower it by setting it to Low. Keep the settings at the default levels unless you are seeing a significant performance impact.

Step 5 (Optional) Modify SNMP Agent Performance by changing the number of Concurrent Requests, Subagent Wait Time (in seconds), and Subagents.

The default values are 5, 25, and 25 respectively. Keep the settings at the default levels unless you are seeing a significant performance impact.

Maximum Limits Settings for Agent Performance

Specify maximum limits for agent performance in the **General Information Properties** dialog.

Concurrent Requests

The maximum number of SNMP requests that a subagent can currently process. Any pending requests above this value are queued.

Subagent Wait Time

The maximum number of seconds that the primary agent waits for a subagent response.

Subagents

The maximum allowable subagents that the primary agent loads.

Change Agent Log Quantity Setting

Important Change this value only under direction from Cisco Technical Assistance (TAC).

Procedure

Step 1 Change **Agent Log Quantity** setting in **General Information Properties** dialog.

- **Verbose** (most information),
- **Normal** (default), or
- **Terse** (least information)

Note
You can retrieve logs using the Analysis Manager.

Step 2 Click **OK** to save changes.

Configure SNMP Trap Destinations

You can configure SNMP Trap Destinations for SNMP v1, SNMP v2c, and SNMP v3. A trap is a notification used by the SNMP agent to inform the NMS of a certain event.

Procedure

Step 1 Expand **Cisco SNMP Agent Management** in left pane of MMC snap-in.

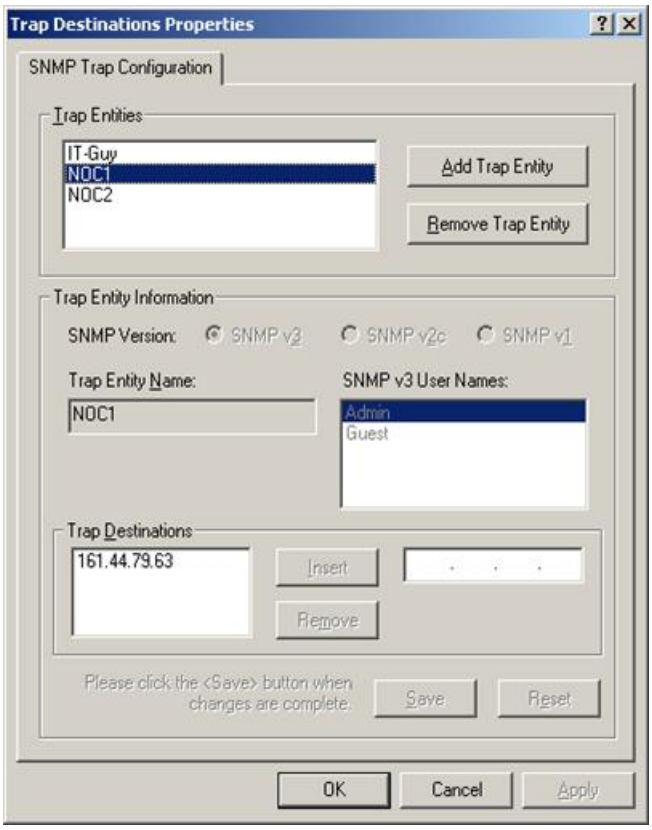
Step 2 Highlight **Trap Destinations** in left pane under Cisco SNMP Agent Management.

Trap Entity Name and SNMP Version columns appear in the right pane.

Step 3 Right-click white space in right pane and select **Properties**.

A dialog box appears:

Figure 9: SNMP Trap Destination Configuration Dialog Box



Step 4 Click **Add Trap Entity**.

Step 5 Select SNMP version in **Trap Entity Information** for version of SNMP used by your NMS.

Step 6 Provide name for trap entity in **Trap Entity Name** field.

Step 7 Select user or community name to associate with trap.

This list is auto-populated with existing users/community names that have been configured.

Step 8 Enter IP addresses in IP address entry field (containing “dots”) and click **Insert** to define one or more destinations for traps.

Step 9 Click **Save** to save trap destination.

The Trap Entity Name appears in the **Trap Entities** section at the top of the dialog box.

Note
You can remove the Trap Entity by highlighting the name in the **Trap Entities** section and clicking **Remove Trap Entity**.

Step 10 Click **OK**.

Multihomed Windows Server

A multihomed server is a server that connects to multiple network interfaces or IP addresses. If your Microsoft Windows Server uses several IP addresses, you cannot guarantee that the SNMP primary agent binds to the appropriate IP address.

For example, on Unified CCE Routers or Router/Logger (Rogger) systems, **SNMP get** requests may arrive on the public nonhigh priority interface although responses are sent on the public high interface. Also, outbound SNMP traps may be sent using the public high interface when the network management station expects the SNMP traps sent using the public nonhigh priority interface.

There are two ways to bind the SNMP primary agent to the appropriate IP address:

- Continue to use the public high-priority network interface. Configure the network management station to poll the public high interface.
- Use Web Setup to change the IP addresses for the Routers. Use the public high network interface for public (normal priority) traffic. Use the public nonhigh interface for high priority traffic. Ensure that you specify the appropriate priority (high or normal) for all nodes.