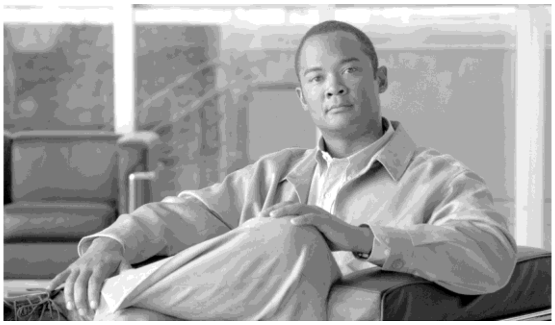


# Cisco Unified Intelligent Contact Management / Unified Contact Center, Enterprise and Hosted

# Serviceability Best Practices Guide

**Release 7.x**

**Last Updated:** August 9, 2012

**Corporate Headquarters**

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

THE INFORMATION IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

*Cisco Intelligent Contact Management / Contact Center Enterprise and Hosted, Serviceability Best Practices Guide for Version 7.x*

# Table of Contents

# List of Tables

# List of Figures

# 1 Introduction

The Serviceability Best Practices Guide is intended to provide information to effectively monitor and manage Cisco Unified Contact Center Enterprise (Unified CCE) / Hosted (Unified CCH) and Cisco Unified Intelligent Contact Management Enterprise (Unified ICME) and Hosted (Unified ICMH).

## 1.1 Target Audience

The target audience for this document is system administrators who will be monitoring and managing Unified CCE/Unified CCH and Unified ICME/Unified ICMH.

## 1.2 Document Intent and Focus

The intent of this document is to provide the reader (presumably one who does not necessarily possess extensive, detailed knowledge of the use of Unified ICM/Unified CC) with sufficient information to understand the product from a management perspective, and to describe in detail the capabilities of the management interfaces and features. The hope is that the reader can then formulate a management and monitoring strategy or easily integrate the management of Unified ICM/Unified CC into an existing network management infrastructure.

The focus of this document is Unified CCE. The vast majority of the content and serviceability features are supported by (and the vast majority of the content applies to) Unified ICME management as well. Where certain content is specific only to one product or the other, such a notation will be made.

## 1.3 Product Names

Some of the product names and other terminology have been changed over time. Some of the supporting documentation has not been updated to reflect the new names. In some cases, even user interfaces and splash screens have yet to be modified to reflect current release product names.

**Table 1-1: Product Names**

| Current Name | Previous Name | AKA(s) | Notes |
|---|---|---|---|
| Cisco Unified Contact Center Enterprise (Unified CCE) | IP Contact Center Enterprise Edition (IPCC/E) | Classic IPCC | |
| Cisco Unified Contact Center Hosted (Unified CCH) | IP Contact Center Hosted Edition (IPCC/H) | Hosted IPCC | |
| Cisco Unified System Contact Center Enterprise (Unified SCCE) | System IPCC (SIPCC) | Simplified IPCC | This term refers to deployments that use the System PG and the web-based configuration interface. |
| Outbound Option | Blended Agent | | User Interface and some documentation may still refer to this as Blended Agent. |

| | | | |
|---|---|---|---|
| Cisco Unified Intelligent Contact Management Enterprise (Unified ICME) | Intelligent Contact Management Enterprise Edition (ICM/E) | Intelligent Call Router (ICR) | |
| Cisco Unified Intelligent Contact Management Hosted (Unified ICMH) | Intelligent Contact Management Hosted Edition (ICM/H) | | |

# 2 Product Architecture

## 2.1 Overview – Cisco Unified Contact Center

Unified CCE delivers intelligent contact routing, call treatment, network-to-desktop computer telephony integration (CTI), and multi-channel contact management over an IP infrastructure. It combines multi-channel automatic call distributor (ACD) functionality with IP telephony in a unified solution, enabling customers to rapidly deploy a distributed contact center infrastructure.

Unified Contact Center provides:

- Segmentation of customers and monitoring of resource availability
- Delivery of each contact to the most appropriate resource anywhere in the enterprise
- Comprehensive customer profiles using contact-related data, such as dialed number and calling line ID
- Routing to the most appropriate resource to meet customer needs based on real-time conditions (such as agent skills, availability, and queue lengths) continuously gathered from various contact center components

Unified Contact Center enables customers to smoothly integrate inbound and outbound voice applications with internet applications such as real-time chat, web collaboration, and e-mail. This integration enables a single agent to support multiple interactions simultaneously regardless of which communications channel the customer has chosen.

Unified Contact Center is a distributed solution; there is no single-server implementation but rather Unified Contact Center Enterprise employs multiple servers each with multiple software components. Deployment options are extremely flexible with performance, capacity and network topology driving the deployment design.

Unified Contact Center was derived from Unified ICME with the primary difference being that Contact Center integrates only with the Cisco Unified Communications Manager IP PBX. All other major components of the Unified Contact Center solution are the same as a Unified ICM solution.

The Unified ICM platform was originally designed to route calls between various nodes in the Time Division Multiplexing (TDM) telephone network. It is designed with an emphasis on reliability and flexibility. All processing in these components is message based. The processing of each message is determined entirely by the content of the message, and the current state of the process. The messages are delivered to these components using Unified ICM Message Delivery Service (MDS). MDS ensures that both processes are fed the exact same set of messages in the same order.

One of the most important concepts to understand about Unified Contact Center is its redundancy strategy. The components that contain centralized state are run in duplex, in that there are two of these components that work in lock step to ensure redundancy and immediate recovery from a (single point of failure) fault.

From a device standpoint, a typical Unified CCE deployment looks as follows:

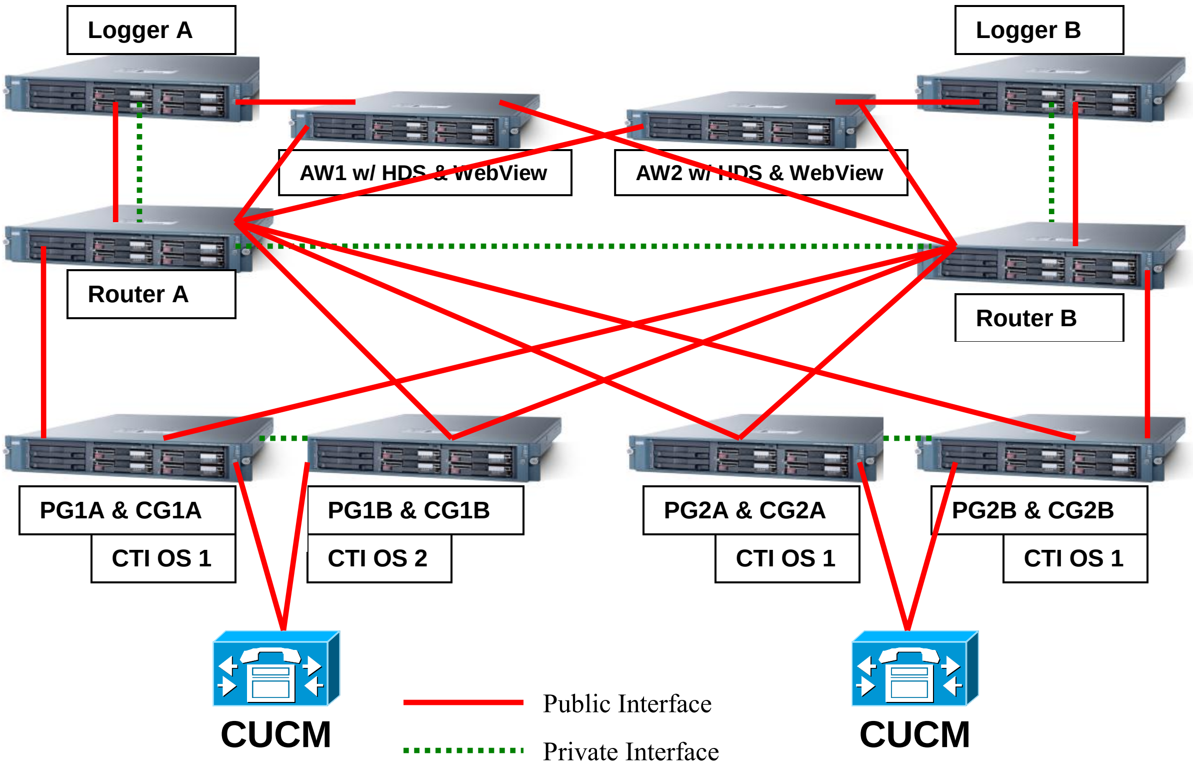


**Figure 1: Typical Unified CCE Architecture**

There are four major components of a Unified CCE deployment: The Router, the Logger, the Peripheral Gateway and the Admin Workstation. The basic function of each is:

1. Router: Make the routing decisions – select a peripheral or agent to receive an inbound contact (voice call, email, and chat).
2. Logger: Store (and replicate) all configuration, real-time and historical data.
3. Peripheral Gateway (PG): Act as a gateway to a peripheral device -- an IP PBX or an Interactive Voice Response (IVR) unit -- as well as a CTI gateway linking agent desktops.
4. Admin Workstation (AW): A server implementation which provides a copy of configuration data (from the logger), an interface for real-time data and a platform for the historical data server (HDS). The Admin Workstation also offers an interface for administrators to generate reports (WebView) and alter configuration and routing scripts (Script Editor, Internet Script Editor).

## 2.2 Router

The Router is the brain of Unified CCE. It is capable of running user defined scripts to make decisions on what should happen with calls, and it has the ability to figure out how to get a call from one place to another. The Router talks to several other components, including the Logger, the Peripheral Gateways (PG), and [Distributor] Admin Workstations (AW) [to include those with a Historical Data Server (HDS)].

The Router receives notification from routing clients (PGs) that a call is in need of some form of routing. It then executes a user-defined script to determine what to tell the routing client to do with the call.

In addition, the Router receives status events and reporting events from PGs. These messages are used to update its current representation of the agents and resources in the system, which is used by the scripts to determine where to send calls. It also sends these messages to the Logger for storage and some of the messages to the Admin Workstations for real-time reporting.

Routers, Loggers and PGs are fault tolerant, having two instances of each component whereby a failure of one provides for “bump-less” continuation of function via the remaining half of a duplex pair. Routers are “duplex” entities, whereby two separate, distributed instances (identified as Side A and Side B) use the Message Delivery Service (MDS) to keep in lock-step with its other side, ensuring that any outage of one side guarantees that the system continues operating without failures or impairments – the opposite side assumes sole responsibility for making routing decisions. All data as well as call control messaging is shared between sides to ensure that both sides have the same data by which to make (the same) routing decisions. Both router sides are “in service” concurrently.

### 2.2.1 Network Interface Controller

[Unified ICME-Unified ICMH only]

Like a Peripheral Gateway, a Network Interface Controller (NIC) is a type of routing client. A NIC is more limited than a PG, however. The purpose of a NIC is to interface with a telephony network, usually the TDM. A NIC is typically co-resident with the Router and used for Unified ICM deployments.

## 2.3 Logger

The Logger is used by Unified CCE to store historical data and configuration data about the call center. It is the place where historical data is first stored, and from which it is later distributed. The logger receives messages from the Router. These messages include detail messages about the calls as well as summary messages that have been computed by the PGs and sent through the Router. Examples of these are ½ hour summaries (how many calls were received during a given period).

The Logger uses a synchronization process that is a little different than the Router. The messages coming to the Logger are only sent from the corresponding Router. Side A Router only sends messages to the Side A Logger. Side B Router only sends messages to the Side B Logger. Because the routers are running in lock-step, it is guaranteed that while messages are flowing they are the same messages; however, recovery happens directly from Logger to Logger, using bulk database copy algorithms for efficiency.

The Loggers also distribute historical data to Historical Data Servers (HDS) and configuration and real time data to the Admin Workstations through MDS. Loggers are duplex as well and are tightly coupled with their respective Router. In many deployments, a side of the Router and Logger are co-located on the same physical server; a Router/Logger combination is often referred to as the “Central Controller”.

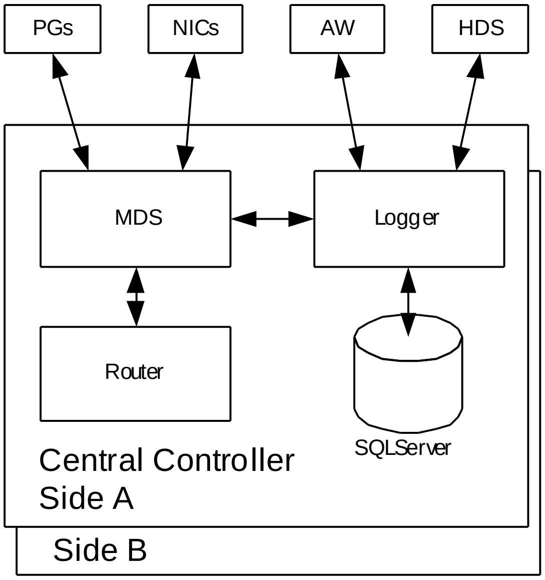


**Figure 2: Central Controller Architecture**

## 2.4 Peripheral Gateway

The Peripheral Gateway (PG) is the component that talks to the telephony devices through their own proprietary CTI interface in a Unified CCE system. These devices can be Automatic Call Distributors (ACDs), Interactive Voice Response (IVR) devices or, in cases such as with Unified CCE, an IP PBX. The PG normalizes whatever protocol the telephony device speaks, and keeps track of the state of agents and calls that are on that device. The PG sends this status to the Router, as well as forwards requests requiring customer logic to the Router.

The PG also exposes a normalized CTI interface to clients. These clients can be traditional CTI clients (wallboards, agent/supervisor desktop clients, and so on), or they can be another instance of Unified CCE, as is the case in a parent/child deployment.

The component of the PG that does the normalization is called a Peripheral Interface Manager (PIM). This component is responsible for actually talking to the peripheral and translating whatever proprietary language it speaks into the normalized one that the Open Peripheral Controller (OPC) and the rest of the PG understands.

There are several groups that PGs fall into. The first classification of PG includes those that talk to an ACD or a Unified Communications Manager (Unified CM) that has agents on it. This is the typical case for a PG. It talks a proprietary CTI protocol to the switch, and maintains the state of agents and calls in queue on the device. While all of these PGs report agent state to the Central Controller, they do it in a different way. In the case of a PG talking to an ACD, the PG mirrors the state of the agents on the ACD; it is keeping a copy of the master state of the agents tracked by the ACD. In the case of a PG attached to a Unified CM, the Communications Manager does not know about agents or agent states, it only knows about phone lines. In this case the PG is actually the master for the agent state.

The second classification of PG is a Voice Response Unit (VRU) or Media Routing (MR) PG. These PGs expose an interface that is client-neutral. In the case of the VRU PG, this interface is tailored to voice calls; in the case of the MR PG, it is more generic task routing that is exposed. These PGs do not maintain agent state, but only maintain the state of calls (or tasks) and expose an interface for the devices to get instructions from the Router.

The third classification of PG is the group PG. There are two types of PGs that talk to groups of peripherals. The first is the Generic PG. This PG allows multiple PIMs of different types to reside inside of the same PG. Each peripheral on this PG behaves completely independently. Currently the Generic PG is only supported for Unified CCE, where it contains a Communications Manager PIM and a VRU PIM talking to an IP-IVR or Customer Voice Portal (CVP). The second type of group PG is a Unified CCE System PG. This PG, like the generic PG, has one Call Manager PIM and one or more VRU PIMs. The System PG ties these multiple PIMs together. In traditional Unified CCE, a call that comes into the Communications Manager then gets transferred to the IP-IVR and then back to an agent looks like three separate calls to Unified CCE. The new PG coordinates these calls and makes that call look like a single call. This is more like what happens on a traditional TDM ACD, where the ACD also has a queue point.

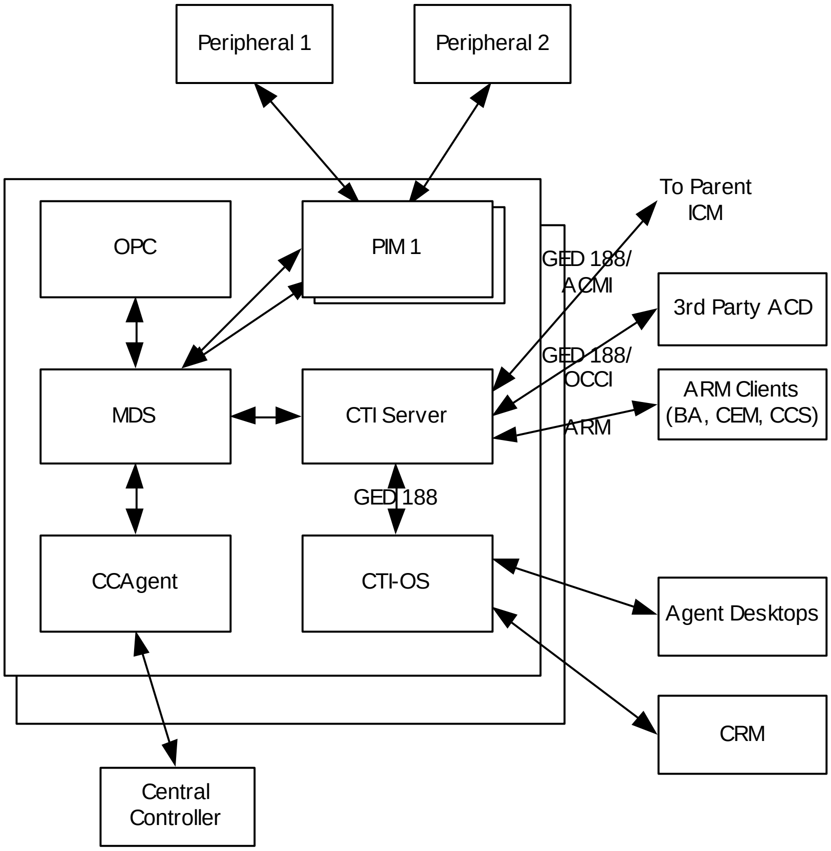


**Figure 3: Peripheral Gateway Architecture**

The PG is duplexed using the same technology as the Central Controller, MDS. This means that there are two PGs operating at any time. All of the messages to the critical process on the PG (OPC) go through the MDS queue, to keep the two operating in lock-step. However, the PG operates slightly different from the Router – from a fault tolerance standpoint – in that while both sides share the same data, for many PG components, only one side is “active”. Should a fault occur, the opposite side activates and continues functioning, having the context of the other side without losing calls.

PGs use the Device Management Protocol (DMP) to communicate between themselves and the central controller. The following depicts the components involved in this communication and the communication links employed:

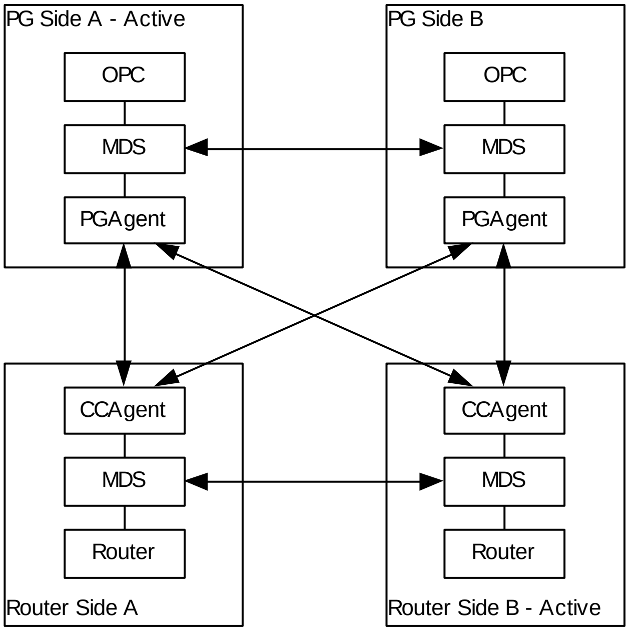


**Figure 4: DMP Flows**

Co-resident with the PG is the CTI Gateway (CG - CTI Server component) and the CTI Object Server (CTI OS).

### 2.4.1 Open Peripheral Controller

OPC is responsible for computing and maintaining the state of agents on the PG, reporting that state to the Router, knowing when a call needs to request instructions from the Router, and performing the CTI operations on the telephony device as necessary. OPC is the critical process on the PG. It is kept in lock-step with its sibling on the other side.

### 2.4.2 Peripheral Interface Manager

The Peripheral Interface Manager (PIM), as previously mentioned, is responsible for the actual connection to the peripheral (ACD, PBX, IVR). This process is not a lock-step process nor is data shared between the two sides. Instead either the Side A or Side B PIM is active for each peripheral. If one side loses its connection, the other side activates.

#### *2.4.2.1 Unified Communications Manager PIM*

[Unified CCE-Unified CCH Only]

The Communications Manager PIM provides the interface between the Cisco Unified CM and the Unified Contact Center Enterprise OPC process. This PIM communicates with Unified CM through the JTAPI Gateway.

#### *2.4.2.2 VRU PIM*

The VRU PIM provides an interface between a voice response unit (VRU) (or IVR). The communication protocol used between the PIM and the VRU is GED-125.

#### *2.4.2.3 Media Routing PIM*

The MR PIM provides the integration point for multimedia contacts such as emails or collaboration (chat) sessions. It is also a necessary component for integration of the Outbound Option Dialer.

#### *2.4.2.4 TDM ACD PIMs*

[Unified ICME-Unified ICMH only]

The TDM ACD PIMs provide interfaces to various manufacturers' Automatic Call Distributors. The communication protocol between the PIM and the ACD is typically proprietary.

### 2.4.3 JTAPI Gateway

[Unified CCE-Unified CCH only]

The JTAPI Gateway is a process that connects to the Unified CM CTI Manager and provides the link between the peripheral gateway and the Unified Communications Manager cluster. The Unified CM CTI Manager communicates CTI messages to/from other nodes in the Unified CM cluster. The JTAPI Gateway provides an added level of translation between the (Java) JTAPI interface and the (C++) Unified Communications Manager PIM.

### 2.4.4 CTI Gateway (CTI Server)

The CTI Server is the interface from OPC to CTI clients. It provides an interface (protocol) specified as GED-188. This interface actually has many flavors and message sets. It has in the past been used as a direct CTI connection to agent desktops or 3rd party desktops. This use has been deprecated.

GED-188 helps to make the details of individual peripherals hidden, but does not fully complete the job. The messages sent from a CTI Server connected to an Aspect PG are different than the messages sent from a CTI Server connected to a Unified CCE PG.

Today CTI Server connects to several types of clients:

- CTI Object Server (CTI OS) – this is the client of choice for agent and supervisor desktops, as well as CRM integration.
- Agent Reporting and Monitoring (ARM) clients – this flavor of GED-188 allows reporting agent status and receiving information about the status of agents. It is one of the integration points for multi-channel (e-mail and web collaboration) applications as well as for the outbound dialing options.
- Parent ICM – a single connection is allowed to a CTI Server attached to a Unified CCE System PG. This connection allows the parent ICM to receive status about agents and calls on this PG, as well as to take control of certain incoming calls and route them itself. This flavor of GED-188 is known as ACMI.

At any given time, only the Side A or Side B CTI Server is active, not both. Clients must connect to one or the other.

### 2.4.5 CTI Object Server

The CTI Object Server (CTI OS) is the connection from the PG to desktop clients and is used for CRM integration. CTI OS completes the abstraction of peripheral type. The set of messages and commands are the same no matter what type of peripheral the PG is connected to.

CTI OS is also used as the per-agent connection to the Cisco Agent Desktop (CAD). CTI OS can connect to both Side A and Side B CTI Servers to provide for a reliable connection.

### 2.4.6 Cisco Agent Desktop

The Cisco Agent Desktop (CAD) base services consist of a set of services that run as Windows Server services. The base services include:

- Chat Service
- Directory Services
- Enterprise Service
- Browser and IP Phone Agent Service
- LDAP Monitor Service
- Licensing and Resource Manager Service
- Recording and Statistics Service
- Sync Service
- Tomcat web Service

The Enterprise Service and BIPPA Service interface with the CTI service, typically running on a peripheral gateway (PG). There are other services that can be placed on the same or separate computer as the base services. These include:

- Voice over IP Monitor Service
- Recording & Playback Service

A set of the base services plus the additional services is a logical contact center, or LCC. The maximum number of agents that can be supported by a single LCC is 2,000 (approximately 15,000 Busy Hour Call Completion [BHCC] with a call volume of 20 calls per agent per hour).

CAD services typically reside co-resident on the same server with PG and CTI OS services.

#### Service Names/Executables

To check if a service is running, use the following table to match what is shown in the Services window (accessed through the Windows Control Panel) with a particular executable.

**Table 2-1: CAD Services and Executables**

| Service Name | Executable Name |
|---|---|
| Cisco Browser and IP Phone Agent Service | IPPASvr.exe |
| Cisco Chat Service | FCCServer.exe |
| Cisco Enterprise Service | CTI Storage Server.exe |
| Cisco LDAP Monitor Service | LDAPmonSvr.exe |
| Cisco Licensing and Resource Manager Service | LRMServer.exe |
| Cisco Recording & Playback Service | RPServer.exe |
| Cisco Recording and Statistics Service | FCRasSvr.exe |
| Cisco Sync Service | DirAccessSynSvr.exe |
| Cisco VoIP Monitor Service | FCVoIPMonSvr.exe |
| Directory Replication Service | slurpd.exe |
| Directory Services | slapd.exe |
| Tomcat Service | tomcat5.exe |

For more details on administering CAD services, please refer to the *Cisco CAD Service Information Manual*.

## 2.5 Configuration System

The Unified CCE configuration system is also based around the concept of reliability and scalability. There can be multiple configuration database copies, which are kept in sync using MDS and a synchronization process from the central controller. Each of these can send updates to the Router, but only the Logger configuration database is authoritative.

The configuration system consists of the DBAgent process on the Router, which accepts connections from the Admin Workstations, and distributes configuration updates to those AWs. The AWs have a copy of the configuration and expose a GUI for browsing and making changes; the AW also exposes an API (ConAPI) for accessing the configuration information and for making changes.

### 2.5.1 Admin Workstation

The Admin Workstation is the main interface to the Unified ICM/Unified CC configuration. On the AW resides a database which contains a copy of the configuration information contained in the Logger. A Distributor process, which receives updates from the central controller, writes to the database to keep everything in sync. Multiple clients read the configuration from the database and send update messages to the central controller DBAgent process.

The two main clients in the AW are the configuration tools which are used to provide a GUI to update the configuration, and the Configuration Management Server (CMS) process which is used to provide the Configuration API (ConAPI).

Processes that connect to ConAPI are the multi-channel components for agent and skill group management and the Unified System CCE Administration process to provide a GUI.

The AW does not have a dependent twin but rather provides fault tolerance in numbers (N+1 model). A typical Unified ICM/CC deployment often has two or more AWs. AWs connect to each central controller side – a primary and a secondary – so that if a failure occurs on its primary link, the secondary is utilized to recover from the failure and restore connectivity.

### 2.5.2 Configuration Updates

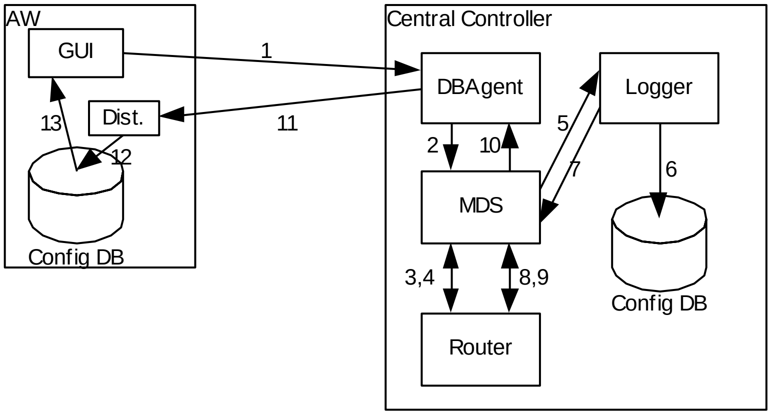


**Figure 5: Configuration System Message Flow**

The message flow for a configuration update is shown in **Error! Reference source not found.**. This is shown to illustrate how a configuration update may happen in Unified CCE.

- The first step (not shown) is that an AW client reads configuration from the AW database, and realizes that it wants to make a change.
- When this happens, the GUI connects to the DBAgent process on the central controller and sends the update (Step 1).
- DBAgent sends the message to the Router, through MDS (Steps 2-3).
- The Router validates the configuration message and sends it to the Logger to be executed (Steps 4, 5).
- The Logger updates its configuration (Step 6)
- The Logger sends confirmation that the update happened to the Router (Steps 7-8).
- The Router then sends the update to all of its clients (DBAgent, PGs, and so on) (Step 9, 10).
- DBAgent sends this message to each of its AW Distributors (Step 11). The AW Distributors update their database (Step 12).
- The Configuration GUI sees the change happen (Step 13).

## 2.6 Reporting System

The reporting system for Unified ICM and Unified CC is similar to its configuration system; they use the same distribution channel.

- Reporting messages are generated by PGs (this includes both detail messages and summary messages) and then are sent to the Central Controller, which consists of the Router and the Logger.
- The Router feeds real-time data to the Administration Server and Real-time Data Servers.

- The Logger stores historical data and replicates it to the Historical Database.
- Administration Server and Real-time Data Servers write those records into the real-time reporting database. Those Administration Server and Real-time Data Servers that are configured to have Historical Data Servers also write the appropriate records to the historical database. Cisco Unified Intelligence Suite (Unified IS) are web applications that uses Java Servlets to build reports to be viewed from thin (web browser) clients.

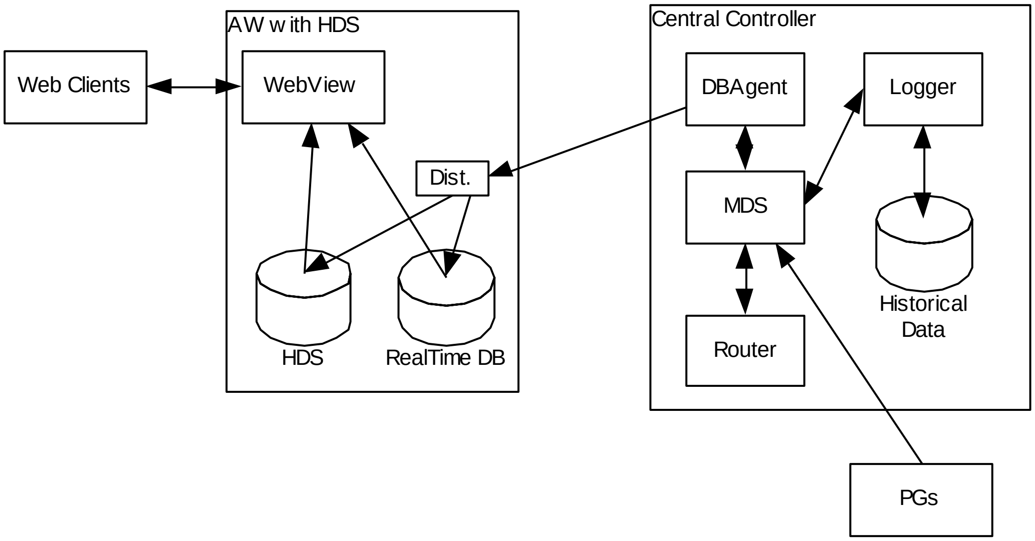


**Figure 6: Reporting Architecture**

**Note**: In the above diagram, the "WebView" component shown may be either the WebView server component (which may be co-resident on the AW/HDS or standalone on its own server) or the Unified Intelligence Center component.

### 2.6.1 Historical Data Server

The Historical Data Server (HDS) is an option to be installed with an Admin Workstation. It uses the same distributor technology used to keep the AW configuration database up to date. The HDS provides a long-term repository for historical data and also offloads historical reporting from the Logger. Historical data is replicated from the Logger to one or more HDSs.

### 2.6.2 WebView (Enterprise Reporting)

WebView is a web application that allows clients to access real-time and historical reporting from the Unified ICM/CC databases. WebView is configured to know where to access its historical and real-time databases. Clients connect to a servlet engine; the New Atlanta ServletExec is what is currently used. The servlet engine connects to the Sybase Jaguar server. The Jaguar server uses PowerBuilder templates to create queries and format results, which are returned to the servlet engine and sent to the web client.

Real-time reporting goes through WebView using the same path as historical reporting, with the exception that the pages refresh themselves on a regular basis, allowing users to see changes as they happen.

WebView requires some source for historical and real-time data. It gets its real-time data from an AW database. Its historical data can either come from a Historical Data Server or, at the very low end, directly from the Logger. WebView can be run without a source for historical data and only run real-time reports.

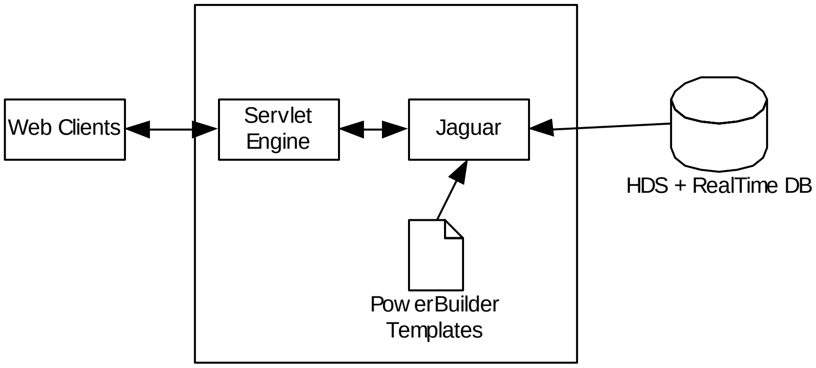


**Figure 7: WebView Architecture**

WebView does not currently expose SNMP instrumentation or generate SNMP notifications (or syslog messages).

## 2.6.3 Unified Intelligence Suite

Unified Intelligence Suite (Unified IS) is a web-based reporting platform for the Cisco Unified Communications products and can be used with Releases 7.5 and 7.2 of Unified ICM, Unified CCE and Unified CCH.

Unified IS consists of two components: the *Unified Intelligence Center (Unified IC)* and the *Archiver*. Each component requires a separate and dedicated server. Unified IC is the user interface for reporting. The Unified IC component, in turn, has two sub-components—a database and a web server. Unified IC:

- Is installed with stock Cisco reporting templates and with tools for modifying those templates.
- Is the interface for creating and maintaining users and user groups.
- Has a Unified IC database that stores metadata and configuration settings and provides the data that is displayed in Error Reports.
- Depending on the deployment model, this database can be configured to reside on the Unified IC server or on the Archiver server.
- Can be deployed with or without the Archiver.

The Archiver is an MS SQL Server data repository. It contains a normalized data schema and a set of stored procedures that pull data from defined data sources for use in reporting. The Archiver is configured to pull data from the Unified ICM/CC AW/HDS.

### *2.6.3.1 Unified IS "Simple" Deployment Model*

In the simple deployment model, the Unified IC web server application and the Unified IC database are installed and configured on a single, dedicated Unified IC server. A simple deployment has no Archiver server.

Unified IC is configured to connect to the Unified ICM/CC Admin Workstation that houses the AW database (_awdb) and the Historical Data Server (_hds). The AW is the data source for real time reports. The HDS is the data source for historical reports.

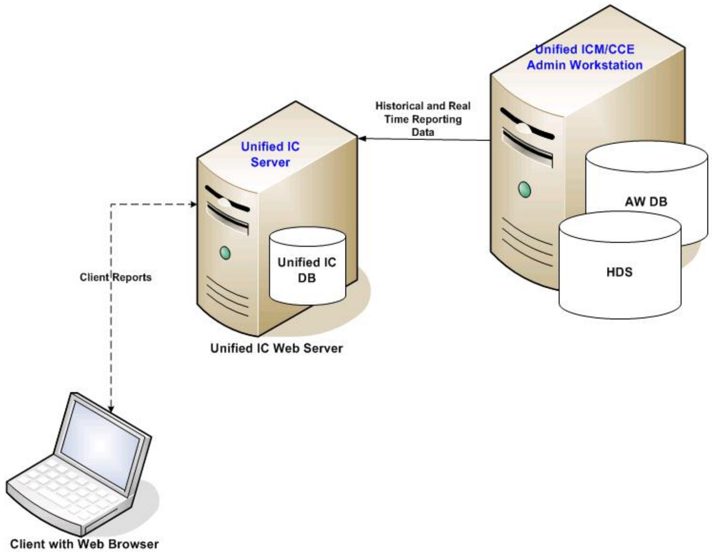


**Figure 8: CUIS Simple Deployment**

### 2.6.3.2 Unified IS "Standard" Deployment Model

In the standard deployment model, the Unified IC connects to the Unified ICM/CC Admin Workstation (AW) *and to* the databases on the Unified IS Archiver. All Unified IS databases—the Unified IC database and the Archiver databases—are configured on the Archiver server. Microsoft SQL Server is installed on the Archiver server.

As in the simple deployment model, Unified IC builds real time reports directly from the AW database on the Admin Workstation. By default, Unified IC also builds most historical reports directly from the HDS. It is the responsibility of the Archiver to collect and aggregate historical data from the Unified ICM/CC AW/ HDS. Unified IC queries are run against the historical data that the Archiver has extracted from the HDS and are not run against the HDS directly. Building historical reports from the Archiver instead of forcing the HDS on the AW to perform potentially complex queries on-demand removes some performance load from the HDS and provides an environment for reporting on historical and aggregated data.

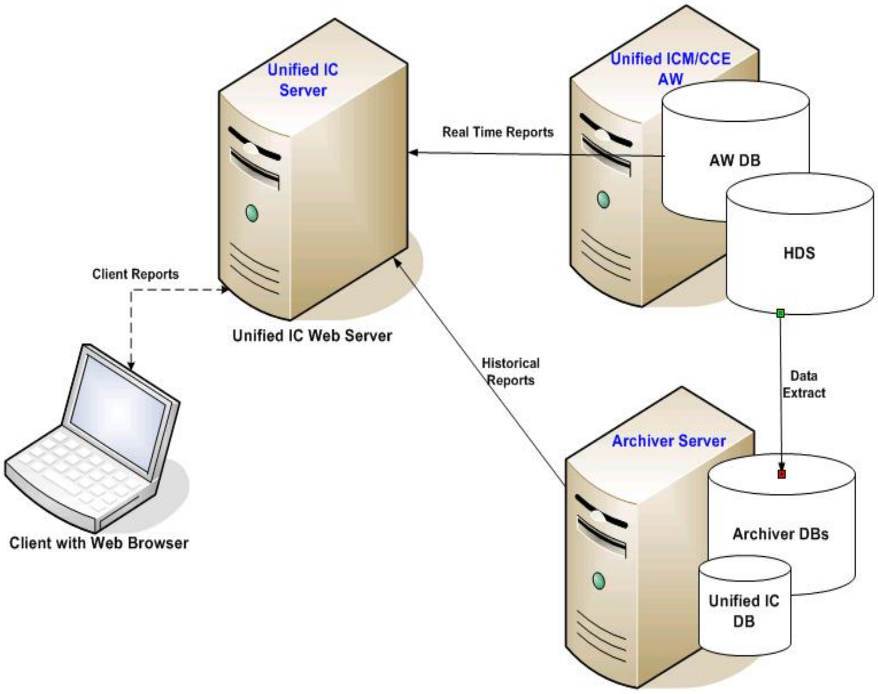


Figure 9: CUIS Standard Deployment

### 2.6.3.3 Unified IS "Scaled" Deployment Model

The scaled deployment is a variation of the standard deployment. In a scaled deployment, there is one Archiver server and there can be multiple Unified IC servers. The Unified IC servers can share SQL Server with the Archiver database, but they must have their own Unified IC databases.

You can deploy a maximum of two Unified IC servers per AW/HDS.

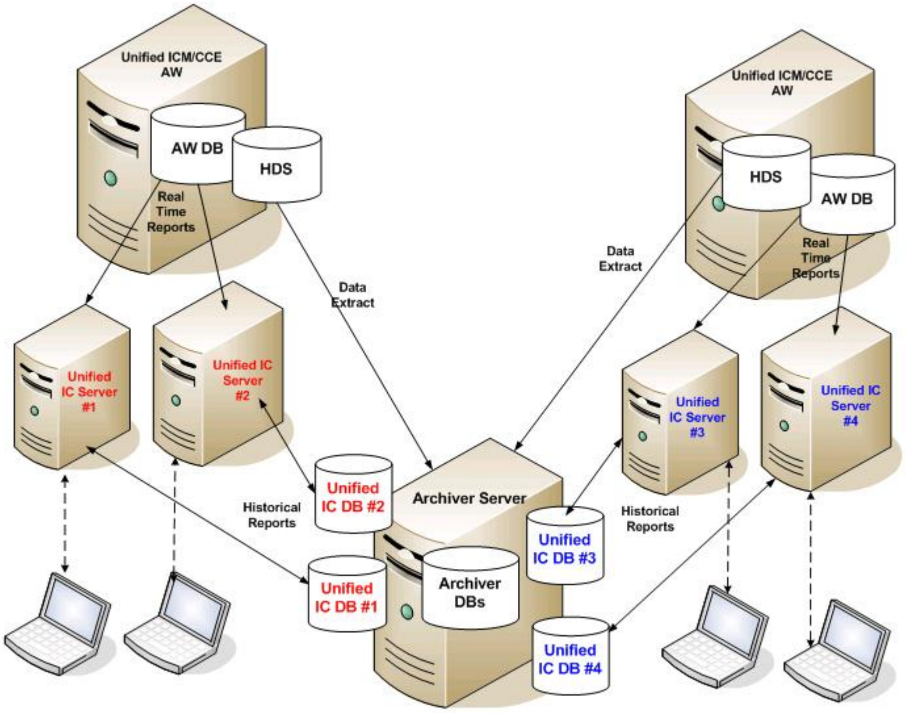


Figure 10: CUIS Scaled Deployment

Unified IS does not currently expose SNMP instrumentation or generate SNMP notifications (or syslog messages).

### 2.6.4 Unified Contact Center Management Portal

Unified CCMP is a suite of server components that simplify the operations and procedures for performing basic administrative functions such as managing agents and equipment, and provide a common, web-based user interface within the entire Unified Contact Center Enterprise and Hosted product set. Unified CCMP consists of four components:

- The **Database Server** component, which utilizes an application called the **Importer** to import enterprise data from different data sources into a Microsoft SQL Server management information database. The database consists of separate database elements that sit on top of SQL Server and which provide data to different reporting elements:
  - **RDBMS Database** (known as the d*atamart*) holds the imported enterprise data
  - **Reporting Services Database** imports and processes data from the datamart so that SQL Server Reporting Services can use it to populate reports
- The **Application Server** component manages security and failover. It manages security by ensuring that users can only view specific folders and folder content as defined by their security login credentials. It verifies that a user is valid and then loads the system configuration that applies to that user. It also manages failover, so if one database server fails, the application can automatically retrieve the required data via an alternative database server
- The **Web Server** component provides a user interface to the platform that allows users to interact with report data, as well as performing administrative functions

- The **Data Import Server** component is an Extract, Transform and Load (ETL) server for data warehouses. The Data Import component imports the data used to build reports. It is designed to handle high volume data (*facts*) such as call detail records as well as data that is rarely changed (*dimensions*) such as agents, peripherals and skill groups

If these components are installed on more than one server, the Data Import and Database components are normally installed on the Database Server. The Application and Web components are usually installed on the Web Application Server.

The Unified CCMP maintains a complete data model of the contact center equipment to which it is connected and periodically synchronized. In addition to configuration information, for example agents or skill-groups, the Unified CCMP can optionally record the events logged by the equipment, such as call records for management information and reporting purposes. The Unified CCMP data model and synchronization activity allows for items to be provisioned either through the Unified CCMP Web interface or from the standard equipment specific user interfaces.

The Unified CCMP system architecture is shown below. The top half of the diagram is a traditional three tier application. This includes a presentation layer (an ASP.NET web application), a business logic application server and a SQL Server database. The lower half of the system architecture is a process orchestration and systems integration layer called the Data Import Server.

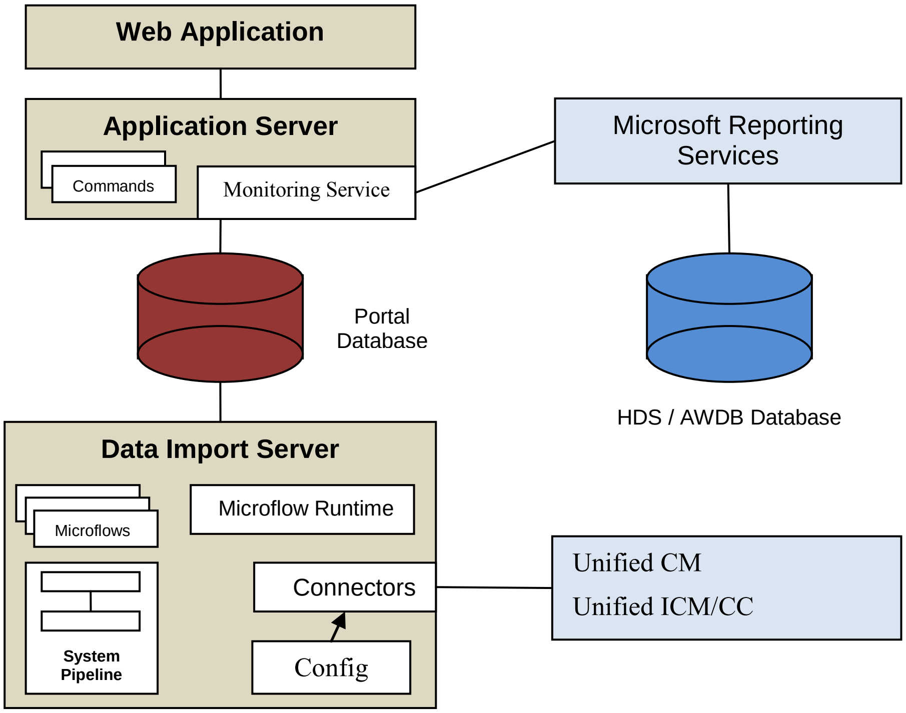


Figure 11: Unified CCMP Architecture

## Web Application

The user interface to Unified CCMP is via a web application that is accessed by a web browser (Microsoft Internet Explorer). Access to the Unified CCMP application is gained through a secure login screen. Every user has a unique user name. This user is assigned privileges by the system administrator. The privileges define the system functions the user can access and perform.

The user interface is time-zone aware and connections to it are secured through HTTPS. The web application is hosted on the server by Microsoft Internet Information Services (IIS) and so is suitable for lockdown in secure environments.

### Application Server

The Unified CCMP Application Server component provides a secure layer in which all business logic is implemented. The application server component runs in a separate service and is always hosted with the web server component. The application server component also includes caching to improve performance and audits all actions taken by logged in users.

### Reporting Services

The Unified CCMP utilizes Microsoft Reporting Services technology for generating reports. Microsoft Reporting Services is an integral part of SQL Server Enterprise Edition. The Unified CCMP provides a flexible reporting system in which reports are authored in the industry standard Report Definition Language (RDL).

### Data Import Server

The Data Import Server component is an Extract, Transform and Load application for the Unified CCMP. The Data Import Server component imports the data used in the Unified CCMP. It is designed to handle high volume data (facts), such as call detail records as well as data which is changed irregularly (resources), such as agents, peripherals and skill groups. The Data Import Server component is also responsible for monitoring changes in the Unified CCMP system and ensuring that those changes are updated onto the Unified ICM/CC and Unified Communications Manager. The Data Import Server component orchestrates the creation, deletion and update of resources to the Unified ICM/CC and Unified Communications Manager. The Microflow Runtime is the heart of the Data Import Server component. It orchestrates systems without resorting to low level programming languages. The Microflow Runtime is a general purpose scripting environment and can be applied to a wide range of problems. The term microflow describes any modular, reusable and independent unit of business logic. An example microflow might update an agent on the Unified ICM/CC when changes are made in the Unified Communications Manager web server component.

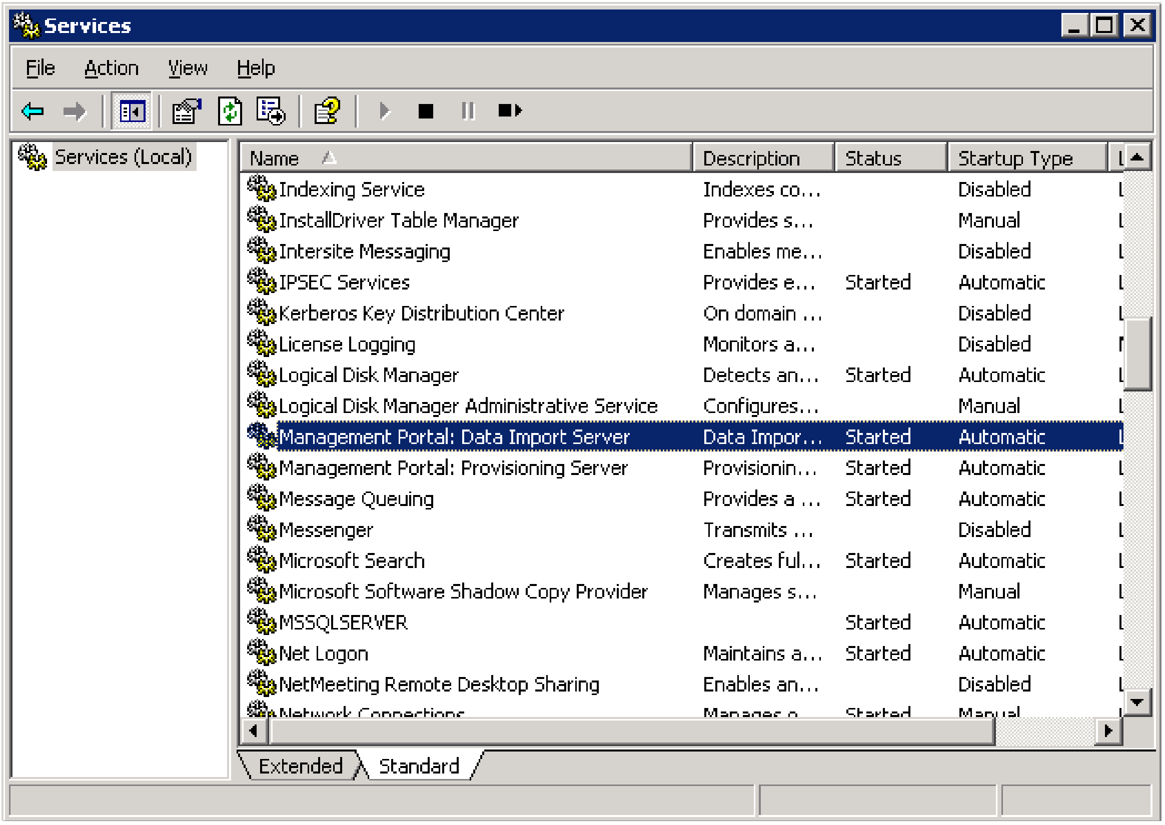
### Unified CCMP Services

- Management Portal: Data Import Server:

The Data Import Server is responsible for importing new dimensions and changes to dimensions such as Agents, Skill Groups, Call Types and Dialed Numbers from Cisco UCCE.  The Data Import Server periodically checks if there are any new dimensions to import or whether there have been any changes made to dimensions that have already been imported.  This allows for closed-loop management of changes made to dimensions provisioned by CCMP.

- Management Portal: Provisioning Server:

The Provisioning Server is responsible for sending provisioning requests from CCMP to Cisco UCCE.  The requests are MACD (move, add change and delete) operations for the resource types that can be managed by CCMP such as creation of new resources, for example a new Agent, or new memberships, such as an Agent to Skill Group membership.  These updates are applied via the ConAPI interface.



| Name | Description | Status | Startup Type | L |
|---|---|---|---|---|
| Indexing Service | Indexes co... | | Disabled | L |
| InstallDriver Table Manager | Provides s... | | Manual | L |
| Intersite Messaging | Enables me... | | Disabled | L |
| IPSEC Services | Provides e... | Started | Automatic | L |
| Kerberos Key Distribution Center | On domain ... | | Disabled | L |
| License Logging | Monitors a... | | Disabled | N |
| Logical Disk Manager | Detects an... | Started | Automatic | L |
| Logical Disk Manager Administrative Service | Configures... | | Manual | L |
| Management Portal: Data Import Server | Data Impor... | Started | Automatic | L |
| Management Portal: Provisioning Server | Provisionin... | Started | Automatic | L |
| Message Queuing | Provides a ... | Started | Automatic | L |
| Messenger | Transmits ... | | Disabled | L |
| Microsoft Search | Creates ful... | Started | Automatic | L |
| Microsoft Software Shadow Copy Provider | Manages s... | | Manual | L |
| MSSQLSERVER | | Started | Automatic | L |
| Net Logon | Maintains a... | Started | Automatic | L |
| NetMeeting Remote Desktop Sharing | Enables an... | | Disabled | L |
| Network Connections | Manages o... | Started | Manual | L |

**Figure 12: Unified CCMP Services**

Unified CCMP exposes a rich set of performance (PerfMon) counters that can be monitored in real-time to gauge status, performance and health.

## 2.7 Outbound Option

Unified ICM and Unified CCE support outbound campaign dialing through its Outbound Dialing subsystem (also known as Blended Agent or BA). The outbound dialing subsystem consists of three major components: The Campaign Manager, the Import Process and the Dialers.

Outbound campaigns start with the Import process. The Import process is used by the customer to import a set of outbound calls into the BA database. This data defines what calls are made and how they are made.

The Campaign Manager is responsible for actually running the outbound dialing campaigns. It reads the campaigns from the BA DB. It then distributes the calls to be made to the dialers. It takes the results of calls and sends reporting information to the Unified ICM/CC central controller where it is recorded in the Unified ICM/CC reporting database.

The dialers actually make the calls, performing the two tasks of agent reservation and dialing. The IP dialer uses the MR PG to reserve an agent to handle the call and it talks to Unified Communications

Manager directly using SCCP (Communications Manager phone protocol) to perform the dialing. Once everything is connected it uses the Unified Communications Manager to connect the call.

The Outbound Option Dialer maximizes the resources in a contact center by dialing several customers per agent. This component resides on the PG server.

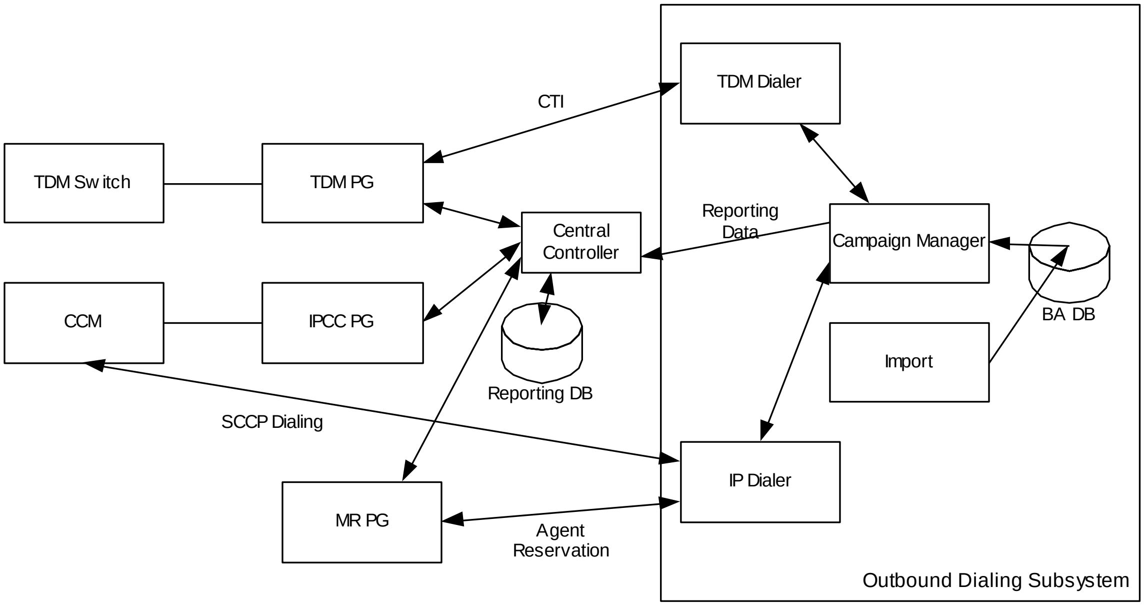


Figure 13: Outbound Option Architecture

# 3 Monitoring SNMP Health

## 3.1 SNMP Overview

### 3.1.1 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 example. to an SNMP network management station (NMS) via SNMP and/or syslog.

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) will be 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 will become 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 contained in section 4.1.

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

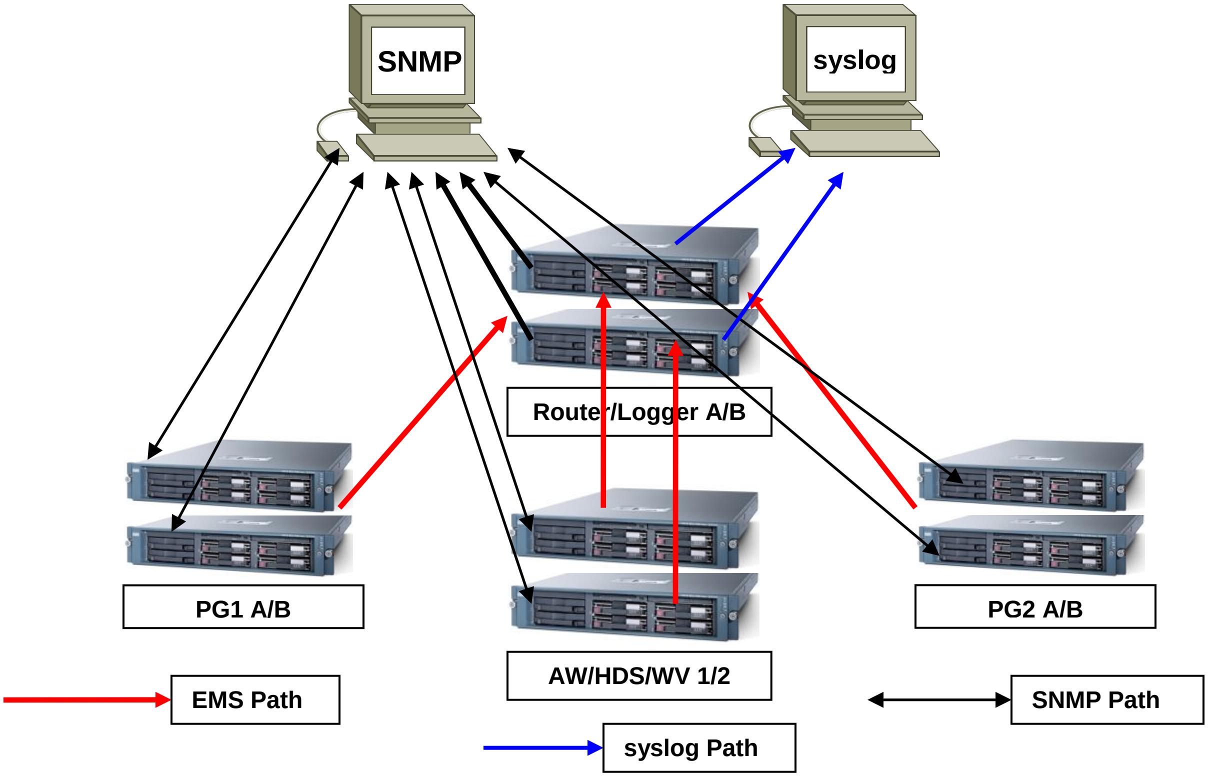


Figure 14: ICM/CC Event Message Flow

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 configured SNMP management station(s). 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 execute 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 only sent to configured collectors. If no syslog collector is configured, the CW2KFeed process will not run and thus no syslog events will be 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 413 configured SNMP notifications for Unified CCE version 7.x.

### 3.1.2 Instrumentation

All Unified CCE nodes 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 will seamlessly integrate with SNMP agent implementation installed by Unified ICM/CC.

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

## 3.2 Base-Level SNMP MIB Support

### 3.2.1 SNMP Master Agent

Unified CCE uses the SNMP Research International EMANATE SNMP agent infrastructure. The agent infrastructure employs typical master/subagent architecture; the master agent supports industry-standard MIB-II instrumentation. Subagents service polls for instrumentation from the MIBs listed herein. There is also a native subagent adapter process which integrates Microsoft Windows extension agents that operate using the native Windows master/subagent interface. Thus, existing extension agents (such as the platform MIB subagents noted above) are seamlessly integrated into the infrastructure.

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

The master agent listens for polls on port 161 (gets/sets) and by default, sends traps to the network management station on port 162. Either port may be configured other than the well-known ports via the Unified CCE Microsoft Management Console (MMC) snap-in configuration tool.

### 3.2.2 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 master agent.

#### *3.2.2.1 Platform MIB Support*

A platform MIB/subagent is provided by the hardware vendor – in case of the Cisco Media Convergence Server (MCS) platform, HP or IBM. This subagent provides instrumentation for low-level attributes of the specific hardware.

For IBM hardware platforms, the following MIBs are supported:

- IBM-SYSTEM-AGENT-MIB
- IBM-SYSTEM-ASSETID-MIB

- IBM-SYSTEM-HEALTH-MIB
- IBM-SYSTEM-LMSENSOR-MIB
- IBM-SYSTEM-MEMORY-MIB
- IBM-SYSTEM-MIB
- IBM-SYSTEM-NETWORK-MIB
- IBM-SYSTEM-POWER-MIB
- IBM-SYSTEM-PROCESSOR-MIB
- IBM-SYSTEM-RAID-MIB
- IBM-SYSTEM-TRAP-MIB

For HP hardware platforms, the following MIBs are supported:

- CPQHLTH
- CPQHOST
- CPQNIC
- CPQSINFO
- CPQSTDEQ
- CPQTHRSH
- CPQSM2
- CPQIDE
- CPQIDA
- CPQSTSYS
- CPQSCSI

#### 3.2.2.2 Host Resources MIB Subagent

The Host Resources MIB is an implementation of RFC-2790. The Host Resources MIB is a standard MIB which instruments 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, processor(s), 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 an Unified ICM/Unified CC managed process. The cccaComponentElmtName field contains the process executable name without the .exe extension. The cccaComponentElmtRunID field contains the process id, which can be used 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 has been 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.

#### *3.2.2.3 Cisco Discovery Protocol (CDP) MIB Subagent*

The CDP is a Cisco-proprietary network protocol used (for our purposes) to broadcast device discovery information to routers and/or switches on the network. Cisco Unified Operations Manager can use this device discovery data to build a network topology and to identify devices within that topology. This means that a network administrator could then click on the device icon for a product node and quickly identify it.

Installation of the CDP driver and CDP subagent is optional on Unified CCE because installation on Cisco MCS servers is not guaranteed. However, Unified System CCE requires Cisco MCS servers and thus the CDP driver and subagent will be installed by default.

**Note:** The CDP driver may cause low-level system halts (for example, blue screen) if installed on servers with an unsupported NIC chipset. This is the reason that the CDP driver and subagent are optionally installed for Unified CCE.)

#### 3.2.2.4 MIB2

The MIB2 is defined in RFC 1213. It contains objects such as interfaces, IP, icmp, and so on.

This MIB is fully supported on Unified CCE deployments.

#### 3.2.2.5 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 executing 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. See the sysApplInstallPkgTable.

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 compose an application, and currently running and previously run applications.

## 3.3 CISCO-CONTACT-CENTER-APPS-MIB

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

- Router
- Logger
- Peripheral Gateways (PGs)
- Distributor Administrator Workstations (AWs)
- CTI Gateways (CGs)
- CTI Object Servers (CTI OS)

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 provides an example of the data provided by an actual Unified ICM/Unified CC installation.

### 3.3.1 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 Distributor Admin Workstation. (Note: In prior versions, the CTI Gateway and the CTI Object Server components were supported installed on separate servers however are only supported co-located on the Peripheral Gateway as of version 7.0.) The SNMP agent infrastructure is installed on all of these components with a subagent that serves 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 distributor AW, the PG (with a sub-table of PIMs), the CG and CTI OS. The component-specific tables are only populated if that component has been installed on the server.

### 3.3.2 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 hosted deployment, there may be up to 25 instances of a particular component installed on a single server (such as a router – one for each “customer” instance in a service provider solution). This is why the Unified CCE instance is the primary index – it would be the only way to distinguish one router from another. However, in a typical Unified CCE deployment, there will only be a single instance.

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 shall communicate with or which Logger is the primary for a particular Distributor AW).

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

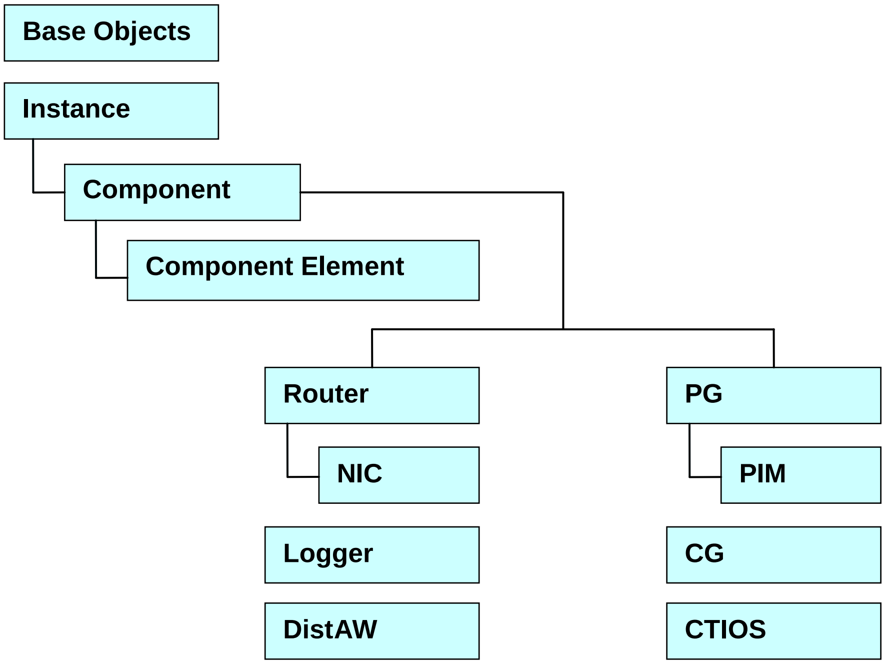


**Figure 15: CISCO-CONTACT-CENTER-APPS-MIB Structure**

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 (arbitrarily assigned by the agent; the value could change from one run period to another).

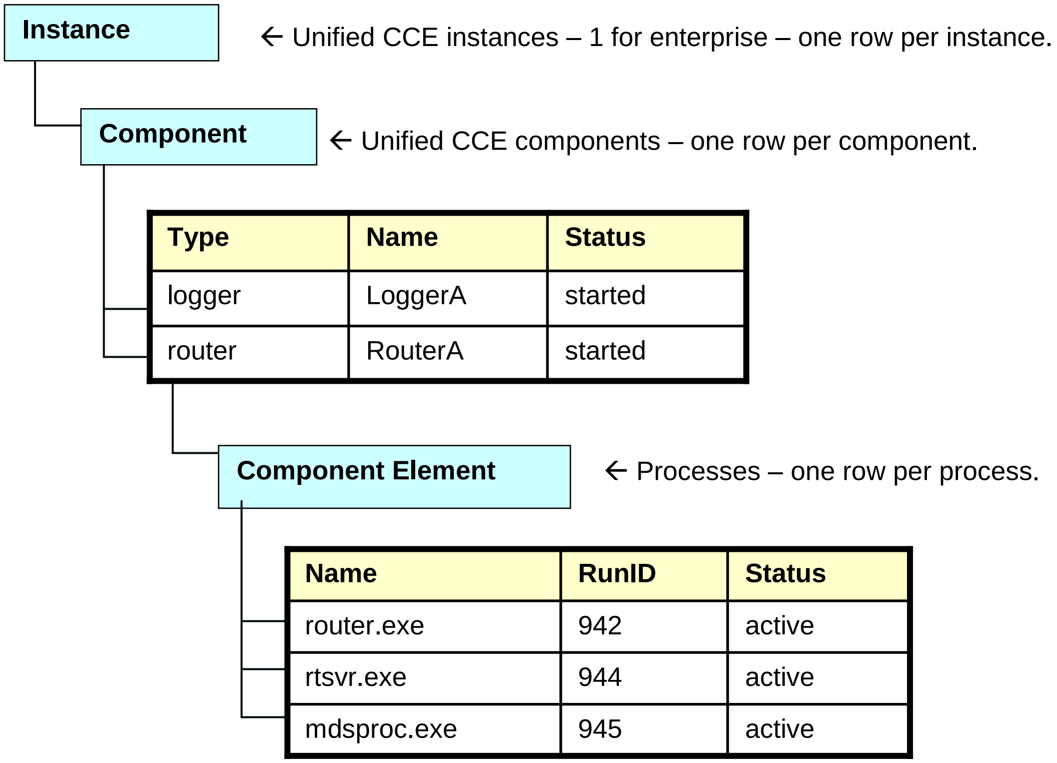
The Component Element table is indexed by Instance, Component number and Component Element number (arbitrarily assigned by the agent; the value could change from one run period to another).

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

So, from an inventory standpoint (a network management station taking inventory of the server itself), the NMS would first poll the Instance table. Typically, for Unified CCE, there will only be one instance. From that, the NMS would poll all components that are part of this instance. Now the NMS knows what is been installed on this server and can see what is actually running. For example, 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 would then poll 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 wishes 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 which will return a list of PIMs installed.

The following figure illustrates content for the application components installed:


Instance
← Unified CCE instances – 1 for enterprise – one row per instance.
Component
← Unified CCE components – one row per component.
Component Element
← Processes – one row per process.

| Type | Name | Status |
|---|---|---|
| logger | LoggerA | started |
| router | RouterA | started |

| Name | RunID | Status |
|---|---|---|
| router.exe | 942 | active |
| rtsvr.exe | 944 | active |
| mdsproc.exe | 945 | active |

**Figure 16: CCCA MIB – Component Inventory Example**

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

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

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

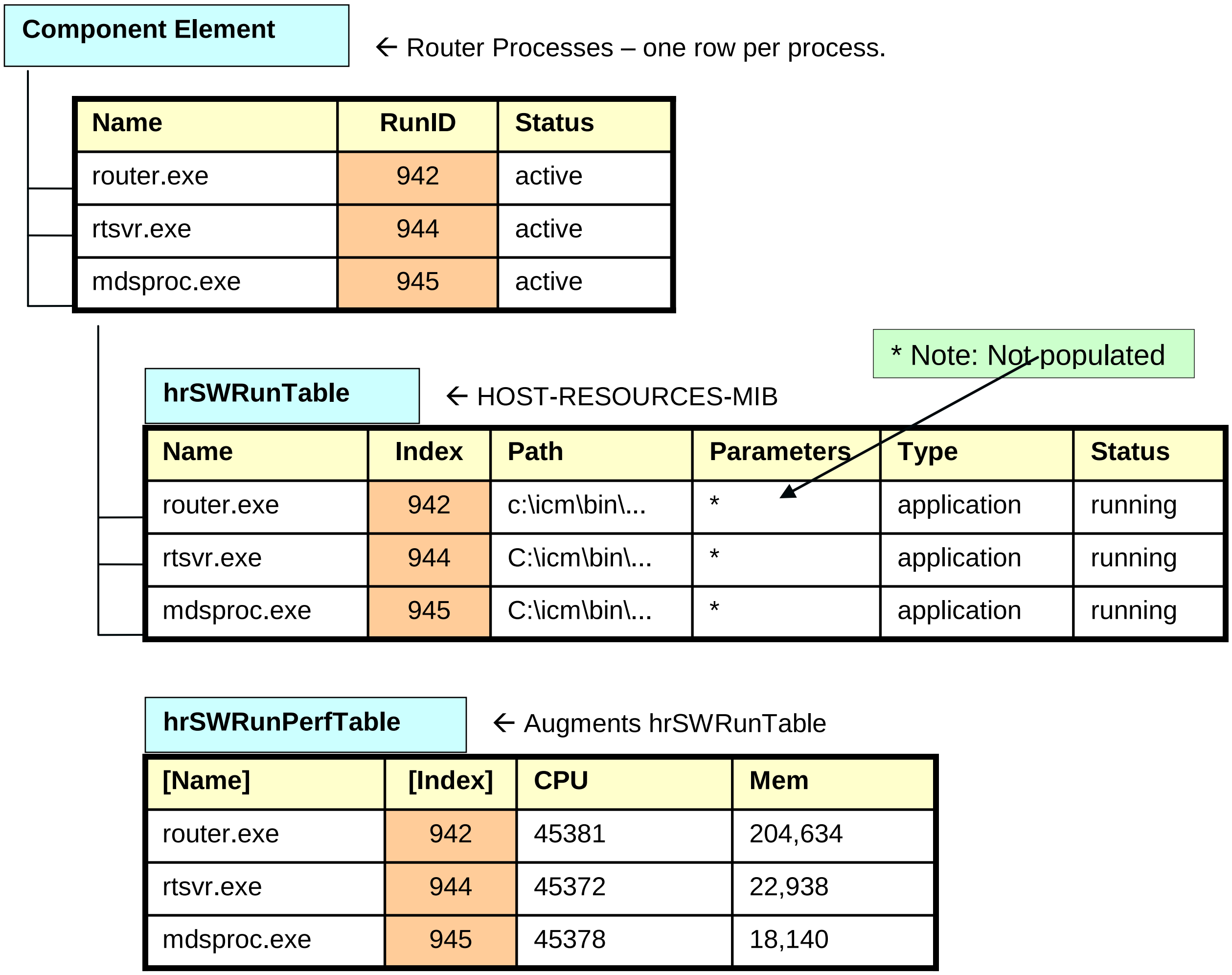
Real-time status of each component may be monitored by polling the cccaComponentTable. The status of the component is derived by analyzing the collective status of each component element (AKA the processes) as best it can.

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

**Note**: The information in Figure 16 is an example, only; there would be many more processes listed in the Component Element table.

### 3.3.3 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.



| Name | RunID | Status |
|---|---|---|
| router.exe | 942 | active |
| rtsvr.exe | 944 | active |
| mdsproc.exe | 945 | active |

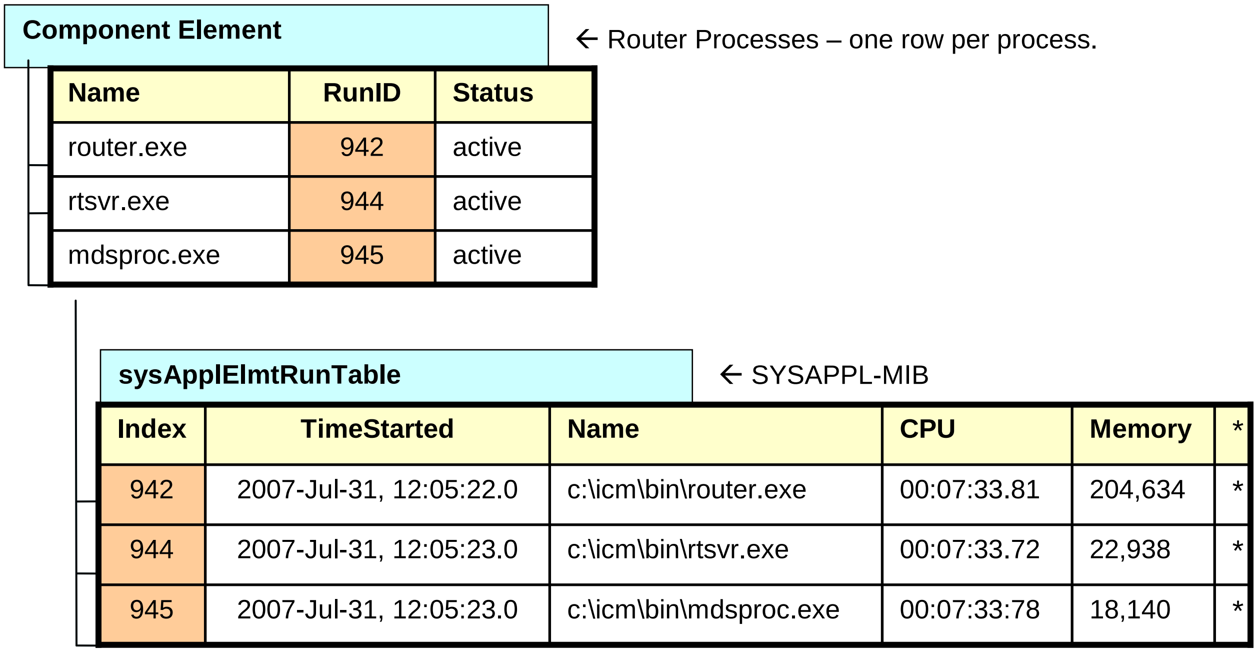
| Name | Index | Path | Parameters | Type | Status |
|---|---|---|---|---|---|
| router.exe | 942 | c:\icm\bin\... | * | application | running |
| rtsvr.exe | 944 | C:\icm\bin\... | * | application | running |
| mdsproc.exe | 945 | C:\icm\bin\... | * | application | running |

| [Name] | [Index] | CPU | Mem |
|---|---|---|---|
| router.exe | 942 | 45381 | 204,634 |
| rtsvr.exe | 944 | 45372 | 22,938 |
| mdsproc.exe | 945 | 45378 | 18,140 |

**Figure 17: 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.

It is important to 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 are either not able to be implemented or simply are not necessary. There are no cut-and-dried rules. A case of where this can be problematic is the “Status” object where the implementation provided on Unified CCE servers interprets the status in such a way that is confusing to some. In most cases, the run status of a process will be reported as “notRunnable(3)”, indicating that the process is in a wait state – waiting for CPU resources. Do not assume that this means the process is not running – it is indeed loaded and running, albeit waiting for its CPU cycles. Any process that is listed in a row of the hrSWRunTable is a running process; if the process terminates, it will be deleted from the table.



**Component Element** ← Router Processes – one row per process.

| Name | RunID | Status |
|---|---|---|
| router.exe | 942 | active |
| rtsvr.exe | 944 | active |
| mdsproc.exe | 945 | active |

**sysApplElmtRunTable** ← SYSAPPL-MIB

| Index | TimeStarted | Name | CPU | Memory | * |
|---|---|---|---|---|---|
| 942 | 2007-Jul-31, 12:05:22.0 | c:\icm\bin\router.exe | 00:07:33.81 | 204,634 | * |
| 944 | 2007-Jul-31, 12:05:23.0 | c:\icm\bin\rtsvr.exe | 00:07:33.72 | 22,938 | * |
| 945 | 2007-Jul-31, 12:05:23.0 | c:\icm\bin\mdsproc.exe | 00:07:33:78 | 18,140 | * |

* Note: Not all entries in this table are implemented. Those shown are typically populated.

**Figure 18: 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) – since 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) – since always installed on the PG, this is a subcomponent of the PG
- CTI Gateway Table (cccaCgTable)
- CTI Object Server Table (cccaCtiOsTable)

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 CC. See section 4.1 for more details on the notification type object.

### 3.3.4 CISCO-CONTACT-CENTER-APPS-MIB Object Descriptions

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

**Table 3-1: CCCA MIB Base Objects**

| Object Name | Description |
|---|---|
| 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 cccaTimeZoneOffsetHours object to represent the local time zone's total offset from GMT. |
| cccaSupportToolsURL | 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 Distributor AW host. This object offers a navigation point from the management station (assuming a web interface) can quickly access the Support Tools application server. |

**Table 3-2: CCCA MIB Instance Table Objects**

| Object Name | Description |
|---|---|
| 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. |

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.

**Table 3-3: CCCA MIB Component Table Objects**

| Object Name | Description |
|---|---|
| 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.<br>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.<br>unknown(1): The status of the functional component cannot be determined.<br>disabled(2): The functional component has been explicitly disabled by an administrator.<br>stopped(3): The functional component is stopped. The component may be dysfunctional or impaired.<br>started(4): The functional component has been started.<br>active(5): The functional component has been started, is currently running and is the active side of a fault tolerant component duplex pair.<br>standby(6): The functional component has been started, is currently running and is the 'hot-standby' side of a fault tolerant duplex pair. |

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; there will be one or many entries in this table that relate to a single entry in the instance table.

**Table 3-4: CCCA MIB Component Element Table Objects**

| Object Name | Description |
|---|---|
| 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 file name, 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.<br>unknown(1): The status of the component element cannot be determined.<br>disabled(2): The component element has been explicitly disabled by an administrator.<br>stopped(3): The component element is stopped; it may be dysfunctional or impaired.<br>started(4): The component element has been started.<br>active(5): The component element is currently running. |

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 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.

**Table 3-5: CCCA MIB Router Table Objects**

| **Object Name** | **Description** |
|---|---|
| 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 will be in various states of treatment. |
| cccaRouterDuplexPairName | The host name 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 will be 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. |

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.

The router table has a sparse dependent relationship with the component table. The instance number acts as the primary index for the router table in order 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.

**Table 3-6: CCCA MIB NIC Table Objects**

| Object Name | Description |
|---|---|
| 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. |

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.

**Table 3-7: CCCA MIB Logger Table Objects**

| Object Name | Description |
|---|---|
| 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 host name 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 host name 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 host name 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 will be the null string.<br>The logger connects to its duplex pair via a 'private' interface -- a closed subnet that guarantees a quality of service level that will 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 will be replicating data to a distributor AW Historical Data Server (HDS). 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. |

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 in order 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.

**Table 3-8: CCCA MIB Distributor AW Table Objects**

| **Object Name** | **Description** |
|---|---|
| cccaDistAwSide | Which of the duplex pair this entry represents, of an enterprise contact center application fault tolerant distributor administrator workstation functional component. The distributor AW side value is either 'A' or 'B'. For simplex configurations, the distributor AW side value defaults to 'A'. |
| cccaDistAwType | Which type of enterprise contact center application distributor administrator workstation, is installed on this server. The distributor AW 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 distributor AW component. |
| cccaDistAwRouterSideAName | The host name of the side 'A' router that this enterprise contact center application distributor AW functional component is associated. The distributor AW 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 distributor AW will lose its connection to the side B router and thus use this object |

| | |
|---|---|
| | value to connect to the side A router. |
| cccaDistAwRouterSideBName | The host name of the side 'B' router that this enterprise contact center application distributor AW functional component is associated. The distributor AW 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 distributor AW will lose its connection to the side A router and thus use this object value to connect to the side B router. |
| cccaDistAwLoggerSideAName | The host name of the side 'A' logger that this enterprise contact center application distributor AW functional component is associated. The distributor AW 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 distributor AW will lose its connection to the side B logger and thus use this object value to connect to the side A logger. |
| cccaDistAwLoggerSideBName | The host name of the side 'B' logger that this enterprise contact center application distributor AW functional component is associated. The distributor AW 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 will lose its connection to the side A logger and thus use this object value to connect to the side B logger. |
| cccaDistAwDuplexPairName | The host name of the duplex pair (for example, the other side) server of an enterprise contact center application fault tolerant distributor AW component. If this component is not part of a duplex pair (for example, simplex), the object value will be 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 distributor AW will receive replicated data from the logger at periodic intervals and add the data to the HDS. Client administrator workstations will generate reports based on the data in this HDS. |
| cccaDistAwWebViewEnabled | 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. |
| cccaDistAwWebViewServer<br>Name | 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. The WebView server accesses this database to serve WebView client reports. |

The distributor AW table lists the enterprise contact center application Distributor Administrator Workstation functional components installed and enabled on this server.

The distributor AW table has a sparse dependent relationship with the component table. The instance number acts as the primary or the distributor AW table in order to properly relate a distributor AW 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.

**Table 3-9: CCCA MIB Peripheral Gateway Table Objects**

| Object Name | Description |
|---|---|
| cccaPgNumber | A user-defined numeric identifier for this enterprise contact center application peripheral gateway. The value is limited by the contact center application to a value between 1 and 80; 80 is the maximum number of peripheral gateways supported by the architecture. |
| 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 host name 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 will lose its connection to the side B router and thus use this object value to connect to the side A router. |
| cccaPgRouterSideBName | The host name 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 will lose its connection to the side A router and thus use this object value to connect to the side B router. |
| cccaPgDuplexPairName | The host name 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 will be the null string. |
| cccaPgPimCount | The number of peripheral interface managers configured and enabled for this enterprise contact center application peripheral gateway functional component. This value is limited to 32 - this is the maximum number of PIMs supported on a single peripheral gateway. |

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

The peripheral gateway table has a sparse dependent relationship with the component table. The instance number acts as the primary index for the peripheral gateway table in order to properly relate a peripheral 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.

Table 3-10: CCCA MIB Peripheral Interface Manager Table Objects

| Object Name | Description |
|---|---|
| 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 since 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/or 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 host name or IP address of the peripheral (the PBX, ACD or VRU) that the enterprise contact center application PIM will be connected. If there are multiple interfaces to the peripheral, each host name or IP address will be separated by a comma. |

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

The PIM table is dependent upon 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 peripheral gateway and to the appropriate instance.

The PIM table has an expansion dependent relationship with the peripheral gateway table. There may be one or more PIM entries associated with a single peripheral gateway 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 peripheral gateway and to the appropriate instance. The SNMP agent assigns the PIM number, based upon the configuration, when each PIM table entry is created.

Table 3-11: CCCA MIB CTI Gateway Table Objects

| Object Name | Description |
|---|---|
| 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. The value is limited by the contact center application to a value between 1 and 80 as this is the maximum number of CTI gateways supported by the architecture. |
| 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 host name of the side ‘A’ 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 ‘B’ PG is the active PG and a failure occurs, the side ‘A’ PG then immediately assumes the role. In this case, the CG will lose its connection to the side ‘B’ PG and thus use this object value to connect to the side ‘A’ |

| | |
|---|---|
| | PG. |
| cccaCgPgSideBName | The host name 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 will lose its connection to the side ‘A’ PG and thus use this object value to connect to the side ‘B’ PG. |
| cccaCgDuplexPairName | The host name 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 will be the null string. |

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.

Table 3-12: CCCA MIB CTI OS Table Objects

| Object Name | Description |
|---|---|
| cccaCtiOsServerName | The user-defined textual name assigned to this enterprise contact center application CTIOS component to uniquely identify it. |
| cccaCtiOsPeripheralName | The unique identifier for the peripheral that the enterprise contact center application CTIOS component is associated. This association links the CTI desktop clients with a particular peripheral PBX. |
| cccaCtiOsPeripheralType | The peripheral type that the enterprise contact center application CTIOS is associated. This also then identifies the peripheral PBX type that the CTI desktop clients are associated. |
| cccaCtiOsCgSideAName | The host name of the side 'A' CTI gateway (CG) that this enterprise contact center application CTI object server (CTIOS) functional component is associated. The CTIOS component must be connected to a CG that is part of the same instance. If the side 'B' CG is the active CG and a failure occurs, the side 'A' CG then immediately assumes the role. In this case, CTIOS will lose its connection to the side 'B' CG and thus use this object value to connect to the side 'A' CG. |
| cccaCtiOsCgSideBName | The host name of the side 'B' CTI gateway (CG) that this enterprise contact center application CTIOS functional component is associated. The CTIOS component must be connected to a CG that is part of the same instance. If the side 'A' CG is the active CG and a failure occurs, the side 'B' CG then immediately assumes the role. In this case, CTIOS will lose its connection to the side 'A' CG and thus use this object value to connect to the side 'B' CG. |
| cccaCtiOsPeerName | The host name of the peer server of an enterprise contact center application CTI object server functional component. If this component does not have a peer, the object value will be the null string. Note that the CTIOS component implements fault tolerance slightly differently than other components of the contact center solution. CTIOS maintains two active peer object servers to serve client desktop CTI applications. If a failure occurs on one of the two servers, its clients will connect to the peer server. |

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

The CTIOS table has a sparse dependent relationship with the component table. The instance number acts as the primary index for the CTIOS table in order to properly relate a CTIOS 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.

## 3.4 Configuring the SNMP Agents

### 3.4.1 Installation Prerequisites for SNMP Support

Unified ICM/CC SNMP support is automatically installed during the course of normal setup. No extra steps need be taken *during* setup for SNMP support to be enabled. However, Microsoft Windows SNMP optional components must be installed on Unified ICM/CC servers for any SNMP agents to function.

**Note:** Install the appropriate Microsoft Windows SNMP component(s) before installing any Unified ICM/CC components that require SNMP monitoring. Instructions for installing the Microsoft Windows SNMP component are below.

The Microsoft SNMP component(s) are required for Cisco SNMP support. However, the Microsoft Windows SNMP service is disabled as part of 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.

### 3.4.2 Installing the Windows SNMP Component on Windows 2000 Server

Complete the steps below to install the Windows SNMP component on Windows 2000 Server.

**Note:** You will need to have the Windows 2000 Server CD available to complete this task.

1. Click **Start** > **Settings** > **Control Panel** > **Add/Remove Program Files.**
2. Click **Add/Remove Windows Components** on the left-hand side of the window.
3. In the **Windows Components Wizard** window, scroll down and highlight **Management and Monitoring Tools**
4. Click **Details**
5. Check the box next to **Simple Network Management Protocol**
6. Click **OK** and follow the directions on screen. You might be asked to insert your Windows2000 Server CD. Do so if prompted.

### 3.4.3 Installing the Windows SNMP Components on Windows Server 2003

Complete the steps below to install the Windows SNMP components on Windows 2003 Server.

**Note:** You will need to have the Windows Server 2003 CD available to complete this task.

1. Click **Start** > **Settings** > **Control Panel** > **Add/Remove Program Files**

   **Note:** You might only need to click **Start > Control Panel > Add or Remove Programs**, depending on which Windows Theme you are using.
2. Click **Add/Remove Windows Components** on the left-hand side of the window
3. In the **Windows Components Wizard** window, scroll down and highlight **Management and Monitoring Tools**
4. Click **Details**
5. Check the box next to **Simple Network Management Protocol**
6. Check the box next to **WMI Windows Installer Provider**
7. Click **OK** and follow the directions on screen. You might be asked to insert your Windows2003 CD. Do so if prompted.

### 3.4.4 SNMP Agent Configuration

While all SNMP components are installed and enabled by default, the device is not manageable via an NMS until the solution is properly configured. The Cisco Contact Center SNMP solution is configured 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 will be quite familiar.

#### Adding the Cisco SNMP Agent Management Snap-in

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

1. From the Start menu select **Run...**
2. In the Start box type in **mmc** and press <Enter>
3. From the Console, select **File > Add/Remove Snap-in**
   A new window appears.
4. From the **Standalone** tab, verify **Console Root** is selected in the **Snap-ins added to:** field and click **Add**
5. In the Add Snap-in window scroll down and select **Cisco SNMP Agent Management**
6. In the Add Snap-in window click **Add**
7. In the Add Snap-in window click **Close**
8. Click **OK** in the **Add/Remove Snap-in** window

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

#### Saving the Snap-in View

Once you have loaded the Cisco SNMP Agent Management MMC Snap-in, you can save that console view to a file (with a .MSC file extension) that can be launched directly instead of repeatedly adding the Snap-in to a new MMC console view. To do so, select the **Console >Save As…** menu; a **Save As** dialog will appear.

Select a memorable file name such as **Cisco SNMP Agent Management.msc** (retain the .msc file extension) and 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.

The system administrator must configure the following to grant access to the agents and enable the receipt of SNMP notifications:

SNMP v1/v2c Community Name

OR

SNMP v3 User Names

AND

SNMP Trap Destination(s)

If using SNMP version 1 or version 2c, at least one community string must be configured on each Unified ICM/CC server to be managed, OR

If using SNMP version 3, at least one user name must be configured on each Unified ICM/CC server to be managed.

In order to receive SNMP notifications at a network management station, an SNMP trap destination must be configured on each Unified ICM/CC server. You can also optionally add a syslog destination on a Unified ICM/CC Logger server.  Please note that Unified ICM/CC notifications are only sent from the Unified ICM/CC Logger, however, in order to receive standard SNMP notifications (for example, Link Up or Link Down notifications) as well, a trap destination must be configured on all Unified ICM/CC servers.

**Note:** Some diagnostic tools may use SNMP locally to gather information about the system using one of the community strings configured for Windows SNMP. These community strings are not added to the Contact Center SNMP configuration, which will cause SNMP requests from these diagnostic tools to fail. All communities configured for Windows SNMP should be added to the Contact Center SNMP configuration. It is not necessary for the Windows SNMP service to be started or enabled. The Windows SNMP communities can be found in the "Security" tab by selecting "properties" for the Windows SNMP service from the list of Windows services.

## Configuring 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:

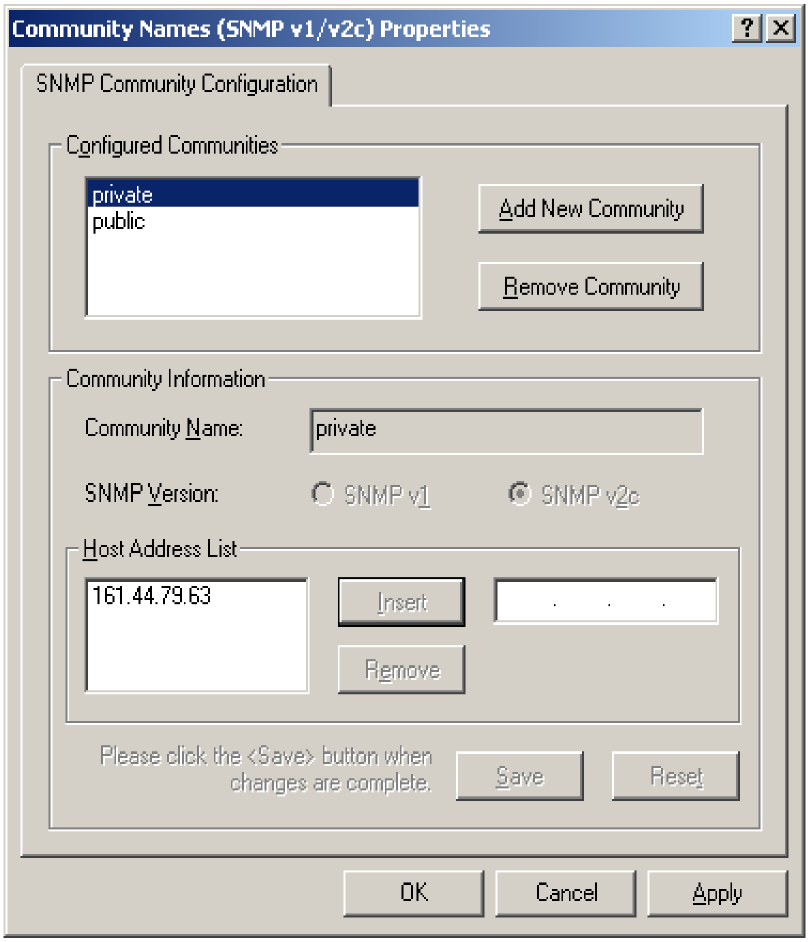
1. Expand **Cisco SNMP Agent Management** in the left pane of the MMC snap-in.
2. Highlight **Community Names (SNMP v1/v2c)** in the left pane under Cisco SNMP Agent Management Community Name, SNMP Version, and Restricted Access columns appear in the right pane.
3. Right click on the white space in the right pane and choose **Properties**

   A dialog box appears.
4. Click **Add new Community**
5. In the dialog box, under **Community Information**, provide a community name.
6. Select the **SNMP Version** by selecting the radio box for SNMP v1 or SNMP V2c.
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.
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**.

Changes become effective when you click **OK**.



Community Names (SNMP v1/v2c) Properties

SNMP Community Configuration

Configured Communities

private
public

Community Information

Community Name: private

SNMP Version: ○ SNMP v1 ● SNMP v2c

Host Address List

161.44.79.63

Please click the <Save> button when changes are complete.

Figure 19: SNMP Community Name Configuration Dialog

## Configuring 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.

To configure a User Name for SNMP v3:

1. Expand **Cisco SNMP Agent Management** in the left pane of the MMC snap-in.
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.
3. Right click on the white space in the right pane and choose **Properties**

   A dialog box appears.
4. Click **Add User**
5. In the **User Configuration** text box enter a user name.
6. If you wish to use SNMP v3 authentication, check **Required?** under Authentication and choose an authentication protocol; then enter and confirm a password.

   **Note:** This setting encrypts the password information as it is sent over the network. These settings must also be used on your NMS to access SNMP data from this server.

7. If you wish to use SNMP v3 privacy, check **Required?** under Privacy and choose an encryption type; then 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 authentication can be configured without configuring privacy. These settings must also be used on your NMS to access SNMP data from this server.

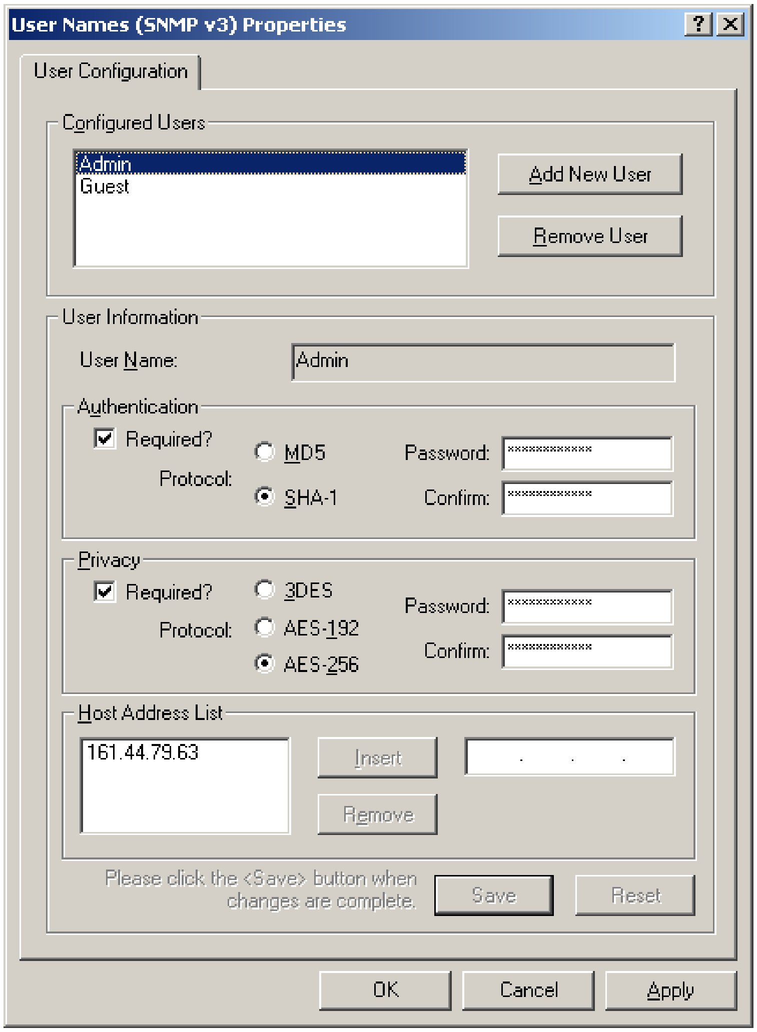
8. Optionally, enter one or more IP addresses in the IP Address entry field (containing "dots") and click **Insert** to enable access solely from the NMS with the IP Address provided.

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 Name by highlighting the name in the **Configured Users** section and clicking **Remove User**.

Changes become effective when you click **OK**.



User Names (SNMP v3) Properties

User Configuration

Configured Users

Admin
Guest

User Information

User Name: Admin

Authentication

☑ Required?

Protocol: ○ MD5 ● SHA-1

Password: *************

Confirm: *************

Privacy

☑ Required?

Protocol: ○ 3DES ○ AES-192 ● AES-256

Password: *************

Confirm: *************

Host Address List

161.44.79.63

Please click the <Save> button when changes are complete.

Figure 20: SNMP User Name Configuration Dialog

## Configuring General Information Properties

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

1. Highlight **General Information** in the left pane under Cisco SNMP Agent Management. Attribute, Value, and Description columns appear in the right pane.
2. Right click on the white space in the right pane and choose **Properties**.

   A dialog box appears.
3. You can change the following properties in the **SNMP System Information** section of the General Information Properties dialog box.

**Table 3-13: SNMP General Information Properties**

| Property | Description |
|---|---|
| System Name | The fully qualified domain name of the system. If empty, this will be automatically filled in. |
| 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. |
| SNMP Port 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 will generate a failed authentication trap. |

Check **Send CISCO-ICM-ALARMEX-MIB Traps** if you wish to send the deprecated notifications defined by this (legacy) MIB. Otherwise, the notifications found in CISCO-CONTACT-CENTER-APPS-MIB.my are used (this is the default for release 7.x).

The notifications are explained in **<INSTALL_DRIVE>/icm/snmp/ccca-Notifications.txt**.

You can change the Windows Execution Priority of the Cisco SNMP agents in the **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.

You can also further 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.

**Definitions:**

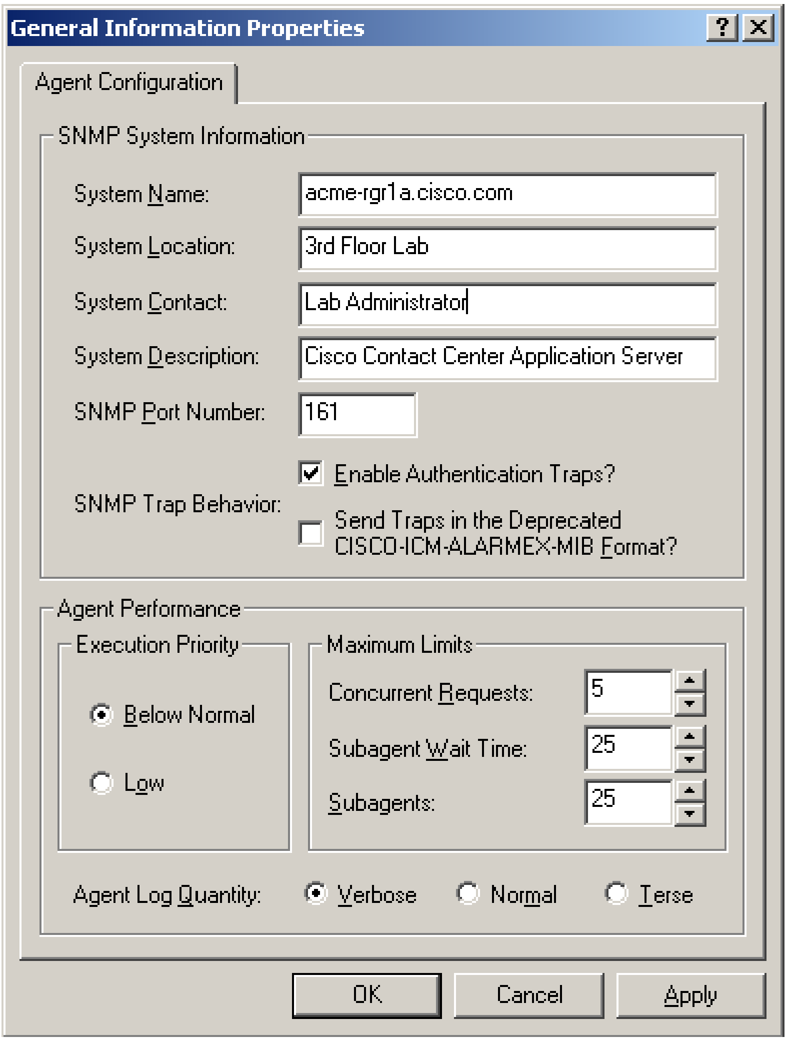
| Definition | Description |
|---|---|
| **Concurrent requests** | The maximum number of SNMP requests that can be concurrently processed by a subagent. Any pending requests above this value are |

| | |
|---|---|
| | queued. |
| **Subagent Wait Time**: | The maximum number of seconds that the master agent waits for a subagent response. |
| **Subagents** | The maximum allowable subagents that the master agent loads. |

You can change the amount of information written to the SNMP logs by choosing Verbose(most information), Normal (Default), or Terse (least information). This value should only be changed under direction from Cisco Technical Assistance (TAC).

**Note:** Logs can be retrieved using Cisco Support Tools.

Click **OK** to save any changes you have made.



General Information Properties

Agent Configuration

SNMP System Information

System Name: acme-rgr1a.cisco.com

System Location: 3rd Floor Lab

System Contact: Lab Administrator

System Description: Cisco Contact Center Application Server

SNMP Port Number: 161

SNMP Trap Behavior:
☑ Enable Authentication Traps?
☐ Send Traps in the Deprecated CISCO-ICM-ALARMEX-MIB Format?

Agent Performance

Execution Priority: ● Below Normal ○ Low

Maximum Limits
Concurrent Requests: 5
Subagent Wait Time: 25
Subagents: 25

Agent Log Quantity: ● Verbose ○ Normal ○ Terse

**Figure 21: SNMP General Information Configuration Dialog**

## Configuring 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. To configure the trap destinations:

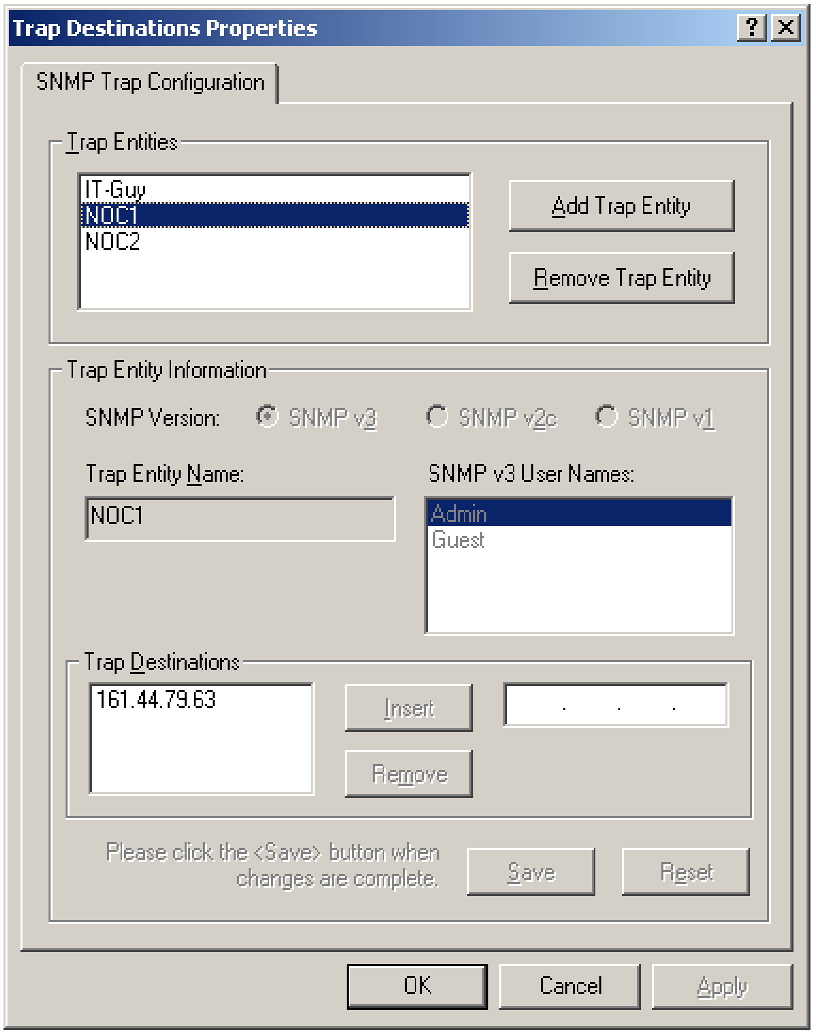
1. Expand **Cisco SNMP Agent Management** in the left pane of the MMC snap-in.
2. Highlight **Trap Destinations** in the left pane under Cisco SNMP Agent Management. Trap Entity Name and SNMP Version columns appear in the right pane.
3. Right click on the white space in the right pane and choose **Properties**

   A dialog box appears.
4. Click **Add Trap Entity**
5. Under **Trap Entity Information** select the SNMP version radio box for the version of SNMP used by your NMS.
6. Provide a name for the trap entity in the **Trap Entity Name** field.
7. Select the SNMP Version Number.
8. Select the User Name/Community Name that you want to associate with this trap. This list is auto-populated with existing users/community names that have already been configured.
9. Enter one or more IP addresses in the IP Address entry field (containing "dots") and click **Insert** to define the destination(s) for the trap(s).
10. Click **Save** to save the new 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**.

Changes become effective when you click **OK**.



Trap Destinations Properties

SNMP Trap Configuration

Trap Entities

IT-Guy
NOC1
NOC2

Trap Entity Information

SNMP Version: ● SNMP v3 ○ SNMP v2c ○ SNMP v1

Trap Entity Name: NOC1

SNMP v3 User Names:

Admin
Guest

Trap Destinations

161.44.79.63

Please click the <Save> button when changes are complete.

Figure 22: SNMP Trap Destination Configuration Dialog

# 4 Understanding ICM/CC SNMP Notifications

Most Unified ICM/Unified CC SNMP notifications are "stateful" events; each event correlates to a managed object. An object is defined as having dual state or single state.

## 4.1 ICM/CC Notification Type

### cccaIcmEvent

An ICM event is a notification that is sent by a functional component of the Cisco Unified Intelligent Contact Management (ICM) and the Cisco Unified Contact Center, Enterprise and Hosted Edition, contact center applications.

The following table details the objects which comprise the notification type:

**Table 4-1: ICM/CC Notification Type Objects**

| Object Name | Description |
|---|---|
| cccaEventComponentId | A unique identifier used to correlate multiple notifications generated by a single enterprise contact center application functional component or subcomponent. A functional component constructs its unique identifier based upon configured parameters; all notifications by that component will include this event component ID. |
| cccaEventState | The state (not to be confused with severity) of the notification and potentially the current state of the functional component that generated the notification. The possible states are:<br>'*clear*' (0): The clear state indicates that the condition which generated a previous raise notification has been resolved.<br>'*applicationError*' (2): The application error state alerts the recipient that an error exists in the enterprise contact center application but that the error does not affect the operational status of the functional component.<br>'*raise*' (4): 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 will follow when the error condition is resolved.<br>'*singleStateRaise*' (9): The single state raise state indicates that a health-impacting error has occurred and that a subsequent clear state notification will not be forthcoming. An example of a single state raise condition is an application configuration error that requires the system to be stopped and the problem resolved by an administrator before the affected component will function properly. |
| cccaEventMessageId | The unique notification message identifier (value) that was assigned by the enterprise contact center application. This identifier is unique for each different notification but consistent for each instance of the same notification. |
| cccaEventOriginatingNode | The application-defined name of the enterprise contact center application functional component that generated this notification. This name will vary, both in content and in format, based on the component that generated the notification. For example, the name for a router component may be 'RouterA', a combination of the component |

| | |
|---|---|
| | identification and the 'side' identifier, while the name 'PG1A' is a combination of the peripheral gateway acronym followed by the peripheral gateway number and the 'side' identifier. |
| cccaEventOriginatingNodeType | The type of enterprise contact center application functional component or subcomponent that generated this notification. The node types are:<br>'*unknown*' (0): The notification originates from an unknown source.<br>'*router*' (1): The notification was generated by the router functional component.<br>'*pg*' (2): The notification was generated by the peripheral gateway functional component.<br>'*nic*' (3): The notification was generated by the network interface controller functional component.<br>'*aw*' (4): The notification was generated by the administrator workstation functional component.<br>'*logger*' (5): The notification was generated by the logger functional component.<br>'*listener*' (6): The notification was generated by the listener functional component. The listener is an enterprise contact center application process that collects event messages from the logger for display in a Cisco proprietary event management application that is part of the Remote Management Suite (RMS).<br>'*cg*' (7): The notification was generated by the CTI gateway functional component.<br>'*ba*' (8): The notification was generated by the Blended Agent functional component. Blended Agent is an enterprise contact center 'outbound option' functional component that manages campaigns of outbound dialing. |
| cccaEventOriginatingProcessName | Each enterprise contact center application functional component includes one or more operating system processes, each of which performs a specific function. The event originating process object identifies the name of the application process that generated this notification. |
| cccaEventOriginatingSide | The enterprise contact center application functional component fault tolerant side (either 'A' or 'B') that generated this notification. |
| cccaEventDmpId | The Device Management Protocol (DMP) is a session layer protocol used for network communication between enterprise contact center application functional components. The DMP ID uniquely identifies the session layer addresses of an application functional component. A single component may have multiple DMP IDs since a functional component will communicate with other functional components (or its duplex pair) via multiple physical network interfaces and maintain multiple DMP session connections on each interface. Should a communications failure occur, the event DMP ID identifies the physical and logical address that the error occurred. |
| cccaEventSeverity | The severity level of<br>this notification. The severity levels are:<br>'*informational*' (1): The notification contains important health or operational state information that is valuable to an administrator, however, the event itself does not indicate a failure or impairment condition.<br>'*warning*' (2): The notification contains serious health or operational |

| | |
|---|---|
| | state information that could be a precursor to system impairment or eventual failure.<br>'*error*' (3): The notification contains critical health or operational state information and indicates that the system has experienced an impairment and/or a functional failure. |
| cccaEventTimestamp | The date and time that the notification was generated on the originating node. |
| cccaEventText | The full text of the notification. This text includes a description of the event that was generated, 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. |

## 4.2 Dual State Objects

Most objects are defined as dual state; they have either a *raise* or *clear* state. The raise state indicates that there is a problem or fault associated with the object. The clear state indicates the object is operating normally.

A dual state Unified ICM/CC SNMP notification contains a raise(4) or clear(0) value in the cccaEventState field. In some cases, multiple raise notifications can correlate to the same object. For example, an object can go offline for a variety of reasons: process termination, network failure, software fault, et cetera. The SNMP notification cccaEventComponentId field specifies a unique identifier that can be used to correlate common raise and clear notifications to a single managed object.

The following example shows a pair of raise and clear notifications with the same cccaEventComponentId. Note that the first notification has a raise state; the notification that follows has a clear state.

```
snmpTrapOID.0 = cccaIcmEvent
cccaEventComponentId = 4_1_CC-RGR1A_ICM\acme\RouterA
cccaEventState = raise(4
cccaEventMessageId = 2701295877
cccaEventOriginatingNode = CC-RGR1A\acme
cccaEventOriginatingNodeType = router(1)
cccaEventOriginatingProcessName = nm
cccaEventOriginatingSide = sideA(1)
cccaEventDmpId = 0
cccaEventSeverity = warning(2)
cccaEventTimestamp = 2006-03-31,14:19:42.0
cccaEventText = The operator/administrator has shutdown the ICM software on ICM\acme\RouterA

snmpTrapOID.0 = cccaIcmEvent
cccaEventComponentId = 4_1_CC-RGR1A_ICM\acme\RouterA
cccaEventState = clear(0)
cccaEventMessageId = 1627554051
cccaEventOriginatingNode = CC-RGR1A\acme
cccaEventOriginatingNodeType = router(1
cccaEventOriginatingProcessName = nm
cccaEventOriginatingSide = sideA(1)
cccaEventDmpId = 0
cccaEventSeverity = informational(1)
cccaEventTimestamp = 2006-03-31,13:54:12.0
cccaEventText = ICM\acme\RouterA Node Manager started. Last shutdown was by operator request.
```

The CCCA-Notifications.txt file is installed in the icm\snmp directory as part of Unified ICM/CC installation. It contains the complete set of SNMP notifications, which can be used to identify grouped events. The Correlation Id is the data used to generate the cccaEventComponentId, which is determined at run time. The following entries correspond to the SNMP notifications in the preceding example.

**Table 4-2: Example "Raise" Notification**

| Field | Value / Description |
|---|---|
| NOTIFICATION | 1028105 |
| cccaEventMessageId | 2701295877 (0xA1028105) |
| DESCRIPTION | Node Manager on the ICM node has been given the command to stop ICM services. This occurs when an operator/administrator stops ICM services using ICM Service Control, 'nmstop', 'netstop', Control Panel Services, or shuts down the node. |
| cccaEventState | Raise |
| SUBSTITUTION STRING | The operator/administrator has shut down the ICM software on %1. |
| ACTION | Contact the operator/administrator to determine the reason for the shutdown. |
| cccaEventComponentId | {cccaEventOriginatingNode %1} |
| CorrelationId | { CLASS_NM_INITIALIZING cccaEventOrginatingNode %1 } |

**Table 4-3: Example "Clear" Notification**

| Field | Value / Description |
|---|---|
| NOTIFICATION | 1028103 |
| cccaEventMessageId | 1627554051 (0x61028103) |
| DESCRIPTION | The Node Manager successfully started. The last reason the Node Manager stopped was because a clean shutdown of the ICM code was requested by the operator. |
| cccaEventState | Clear |
| SUBSTITUTION STRING | %1 Node Manager started. Last shutdown was by operator request. |
| ACTION | No action is required. |
| cccaEventComponentId | { cccaEventOriginatingNode %1 } |
| CorrelationId | { CLASS_NM_INITIALIZING cccaEventOrginatingNode %1 } |

## 4.3 Correlating Dual State Notifications

The cccaEventComponentId is the primary means of matching a clear event to a raise event. When a clear event is received, all pending raise events with the same alarm class and with a matching cccaEventComponentId should be cleared.

- **“Raise” Event:**

| | |
|---|---|
| cccaEventComponentId: | **“4_1_acme-rgr_ICM\acme\RouterA”** |
| Event Class: | **CLASS_NM_INITIALIZING** |
| cccaEventState: | **raise(4)** |
| cccaEventMessageId: | **2701295877** |
| cccaEventSeverity: | **warning(2)** |

| | |
|---|---|
| cccaEventText: | The operator/administrator has shutdown the ICM software on ICM\acme\RouterA. |

- **“Clear” Event**

| | |
|---|---|
| cccaEventComponentId: | **“4_1_acme-rgr_ICM\acme\RouterA”** |
| Event Class: | **CLASS_NM_INITIALIZING** |
| cccaEventState: | **clear(0)** |
| cccaEventMessageId: | **1627554051** |
| cccaEventSeverity: | **informational(1)** |
| cccaEventText: | ICM\acme\RouterA Node Manager started. Last shutdown was by operator request. |

- ✓ Upon receipt of “Raise” event, categorize by severity
- ✓ Upon receipt of “Clear” event, match to “Raise” using ‘cccaEventComponentId’

In the above example notifications, a simple string comparison of “” will suffice in matching the clear to the raise. cccaEventComponentId has the event class built into this value and the rest of the string has been crafted to be sufficiently unique to ensure that the appropriate raise(s) will be cleared by the clear notification. (Remember: Multiple raise notifications can be cleared by a single clear notification.)

Sample logic:

```
If (cccaEventState == “clear”)
    set ID = cccaEventComponentId;
    for (all “raise” events where cccaEventComponentId == ID)
        Acknowledge();
```

There is no one-to-one mapping of alarms by event message ID.

**Note**: SNMP Notifications do not have a unique OID assigned to each alarm. The static assignment of an OID to a notification requires that that notification be explicitly documented (in Cisco customer-facing documents) and maintained following an established deprecation schedule. With so many Cisco devices in service, maintaining such a list is simply impossible. The event definition method in the CISCO-CONTACT-CENTER-APPS-MIB is consistent with the Unified Communications Manager (CISCO-CCM-MIB) and Unified Contact Center Express (CISCO-VOICE-APPS-MIB) product MIBs.

## 4.4 Single State Objects

A single state object has only a *raise* state. Since there is no corresponding clear event, the administrator must clear the object manually. Single state objects are typically used when a corresponding clear event cannot be tracked, for example the database is corrupt. Single state Unified ICM/CC SNMP notifications contain raise(9) value in the cccaEventState field.

The following example shows a value of Single-state Raise in the cccaEventState field to identify a single state object.

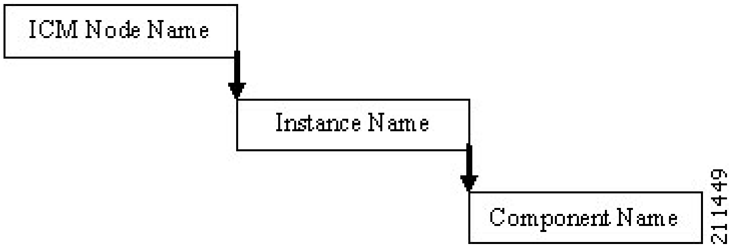
**Table 4-4: Example "Single-State Raise" Notification**

| Field | Value / Description |
|---|---|
| NOTIFICATION | 105023C |
| cccaEventMessageId | 3775201852 (0xE105023C) |
| DESCRIPTION | The router has detected that it is no longer synchronized with its partner. One result of this is that the router might be routing some calls incorrectly. |
| cccaEventState | **Single-state Raise** |
| SUBSTITUTION STRING | The router has detected that it is no longer synchronized with its partner. |
| ACTION | Recommended action: Stop the router on both sides. After both sides are completely stopped, restart both routers.<br>Alternate Action: Restart the router on one side. After doing this, the routers might still route some calls incorrectly, but they will be in sync. Other actions: Collect all rtr, mds, ccag process logs from both routers from the entire day. Collect all sync*.sod files (where * is some number) that exist in the icm\<instance>\ra directory of router A and in the icm\<instance>\rb directory of router B. Contact the Support Center. |
| cccaEventComponentId | { cccaEventOriginatingNode cccaEventOriginatingProcessName cccaEventOriginatingSide } |
| CorrelationId | { CLASS_RTR_SYNC_CHECK cccaEventOriginatingNode cccaEventOriginatingProcessName cccaEventOriginatingSide } |

## 4.5 Organizing SNMP Notifications

Using the contents of the following Unified ICM/CC SNMP notification fields, an SNMP Monitoring tool can group Unified ICM/CC SNMP notifications in an organized, hierarchical manner.

```
cccaEventOriginatingNode = CC-RGR1A\acme
cccaEventOriginatingNodeType = router(1)
cccaEventOriginatingSide = sideA(1)
```



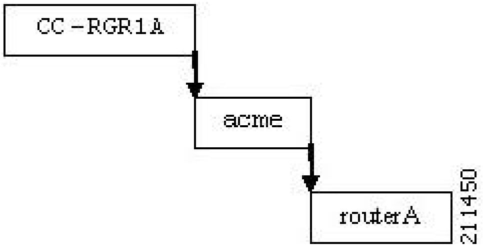
where:

Unified ICM/CC Node Name = left side of cccaEventOriginatingNode

Instance Name = right side of cccaEventOriginatingNode

Component name = cccaEventOriginatingNodeType + cccaEventOriginatingSide letter

For example:



Within this node, raise and clear events with the same **cccaEventComponentId** can be grouped as a single object.

## 4.6 CSFS Heartbeat Notification

The Customer Support Forwarding Service (CSFS) heartbeat notification should be monitored specifically as it is a critical SNMP notification.

**Table 4-5: CSFS Heartbeat Notification**

| Field | Value / Description |
|---|---|
| NOTIFICATION | 12A0003 |
| cccaEventMessageId | 1630142467 (0x612A0003) |
| DESCRIPTION | Periodic message to indicate MDS is in service and that the event stream is active. |
| cccaEventState | |
| SUBSTITUTION STRING | HeartBeat Event for %1 |
| ACTION | No action is required. |
| cccaEventComponentId | { cccaEventOriginatingNode %1 } |
| CorrelationId | n/a |

**Note:** The CCCA-Notifications.txt file defines the decimal value of cccaEventMessageId for this event incorrectly as 19529731.

The heartbeat notification is sent periodically by the Logger CSFS process to indicate a healthy connection exists between the Router and the Logger, and that the Logger SNMP notification feed is active. The heartbeat interval is set to 720 minutes (12 hours) by default. The reason the interval is set this high is to accommodate using a modem to communicate notifications.

You can modify the interval via the Windows Registry value: **`heartbeatIntervalMinutes`**, in:

**`HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\<instance>\Logger<A or B>\CSFS\CurrentVersion`**

The actual interval can be as much as one minute longer than the configured interval, so the logic that reacts to these events should employ a certain "deadband" – in other words, allow for at least 60 seconds beyond the scheduled interval before assuming the worst.

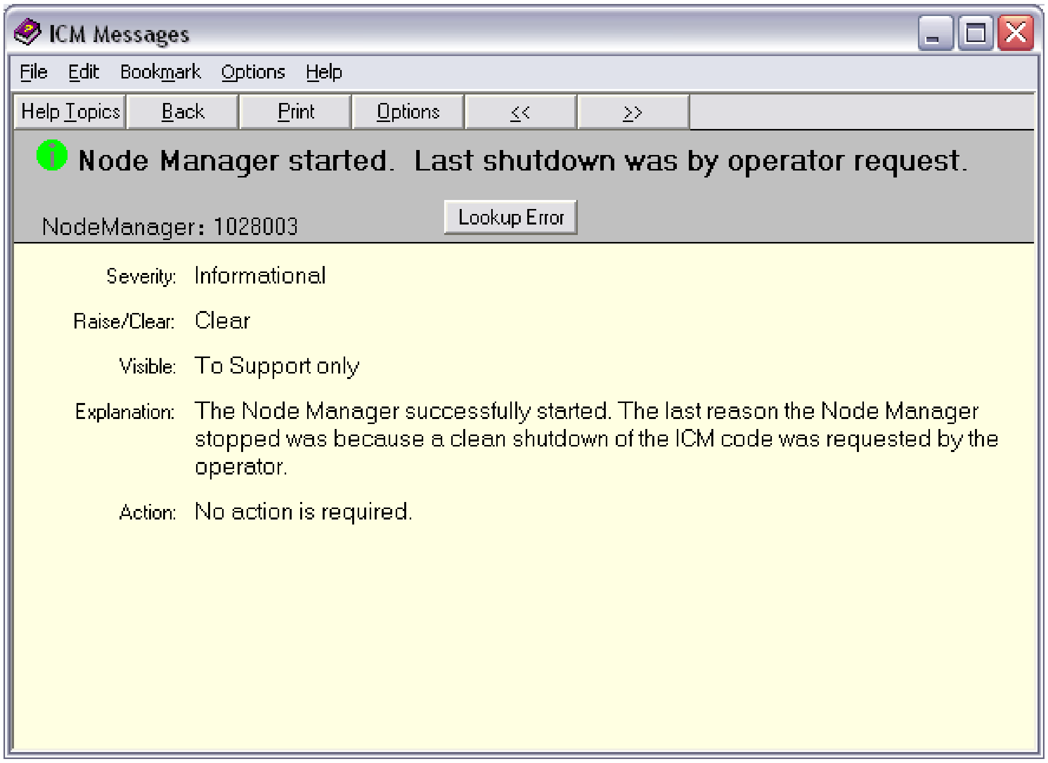
**IMPORTANT**: Monitoring this heartbeat notification provides an additional measure of safety; if the communication infrastructure that sends notifications were to fail, one might assume that the system is operating normally when in fact, it is not.  If this heartbeat event ceases to arrive at the management station, this indicates that that communication infrastructure is impaired and immediately attention is necessary.

## 4.7 More Information on Notifications

Detailed information on each notification is available via a help file installed on every Unified ICM/CC server. The file is named ICRMESS.HLP and is located in the \ICM\BIN directory.

```
C:\icm\bin\ICRMESS.HLP
```

From Windows Explorer, double-click on this file then click on “Help Topics”. If looking for specific message and know the message ID, you may search for it (Lookup Error); else, select an event, and then navigate via Next / Previous buttons


ICM Messages
File Edit Bookmark Options Help
Help Topics Back Print Options << >>
Lookup Error

Node Manager started. Last shutdown was by operator request.

NodeManager: 1028003

Severity: Informational

Raise/Clear: Clear

Visible: To Support only

Explanation: The Node Manager successfully started. The last reason the Node Manager stopped was because a clean shutdown of the ICM code was requested by the operator.

Action: No action is required.

Figure 23: Message Help File

# 5 The syslog Messaging Interface

All versions (since release 4.6(2)) of Unified ICM/CC have provided a syslog (The BSD syslog Protocol, RFC-3164) event feed; this feed was originally designed for the CiscoWorks family of network management products. As a result, the Logger process that provides the syslog feed is named CW2KFeed (CiscoWorks 2000 Feed) however, it is indeed an RFC3164 compliance event feed.

The syslog feed provides a more verbose set of notifications than the SNMP notifications – there are many more events sent via syslog than SNMP and the content matches that which is stored in the Unified ICM/CC database and the Windows Event Log.

Configuration of the syslog feed is done using the Microsoft Management Console snap-in – the same MMC snap-in used to configure the SNMP agents. See below for more details on configuring the syslog feed.

The syslog event feed changed with release 7.2(1) of Unified ICM/CC to format all events in Cisco Log message format. The Cisco Log message format provides the following key benefits:

- Precisely documented message format for wide interoperability.
- Compatible with IOS message format.
- Precise message source identification with host, IP address, application, process, et cetera.
- Message ordering with sequence numbers and timestamp with millisecond precision.
- Support for tagging of messages for correlation or external filtering.
- Support for internationalization of host, tags, and message text.

## 5.1 The Cisco Log Message Format

The Cisco Log message format is:

```
<PRI>SEQNUM: HOST: MONTH DAY YEAR HOUR:MINUTES:SECONDS.MILLISECONDS TIMEZONE: %APPNAME-
SEVERITY-MSGID: %TAGS: MESSAGE
```

Below is an example of a CiscoLog formatted syslog event. An actual entry displays on a single line.

```
<134>25: host-w3k: Feb 13 2007 18:23:21.408 +0000: %ICM_Router_CallRouter-6-10500FF:
[comp=Router-A][pname=rtr][iid=ipcc1][mid=10500FF][sev=info]: Side A rtr process is OK.
```

The following table describes the Cisco Log message fields:

**Table 5-1: Cisco Log Message Fields**

| Field | Description |
|---|---|
| PRI | Encodes syslog message severity and syslog facility. Messages are generally sent to a single syslog facility (that is, RFC-3164 facilities local0 through local7). Refer to RFC-3164 for additional information. |
| SEQNUM | Number used to order messages in the time sequence order when multiple messages occur with the same time stamp by the same process. Sequence number begins at zero for the first message fired by a process since the last startup. |
| HOST | Fully qualified domain name (FQDN), hostname, or IP address of the originating system. |
| MONTH | Current month represented in MMM format (for example, “Jan” for January) |

| | |
|---|---|
| DAY | Current day represented in DD format. Range is 01 to 31. |
| YEAR | Current year represented in YYYY format. |
| HOUR | Hour of the timestamp as two digits in 24-hour format; range is 00 to 23. |
| MINUTE | Minute of the timestamp as two digits; range is 00 to 59. |
| SECOND | Second of the timestamp as two digits; range is 00 to 59. |
| MILLISECONDS | Milliseconds of the timestamp as three digits; range is 000 to 999. |
| TIMEZONE | Abbreviated time zone offset, set to +/-#### (+/- HHMM from GMT). |
| APPNAME | Name of the application that generated the event. APPNAME field values are:<br>PRODUCT_COMPONENT_SUBCOMPONENT<br>PRODUCT – such as ICM<br>COMPONENT – such as Router<br>SUBCOMPONENT – such as Router |
| SEVERITY | Supported severity values are:<br>3 (Error)<br>4 (Warning)<br>6 (Informational)<br>7 (Debug) |
| MSGID | Hexadecimal message id that uniquely identifies the message, such as 10500FF. |
| TAGS | (Optional) Supported tags are:<br>[comp=%s] - component name including side, such as Router-A<br>[pname=%s] - process name, such as rtr<br>[iid=%s] - instance name, such as ipcc1<br>[mid=%d] - message id, such as 10500FF<br>[sev=%s] – severity, such as info |
| MESSAGE | A descriptive message about the event. |

## 5.2 Configuring syslog Destinations

You can configure syslog destinations using the Cisco SNMP Agent Management Snap-in. The syslog feed is only available on the Unified ICM/CC Logger Node.

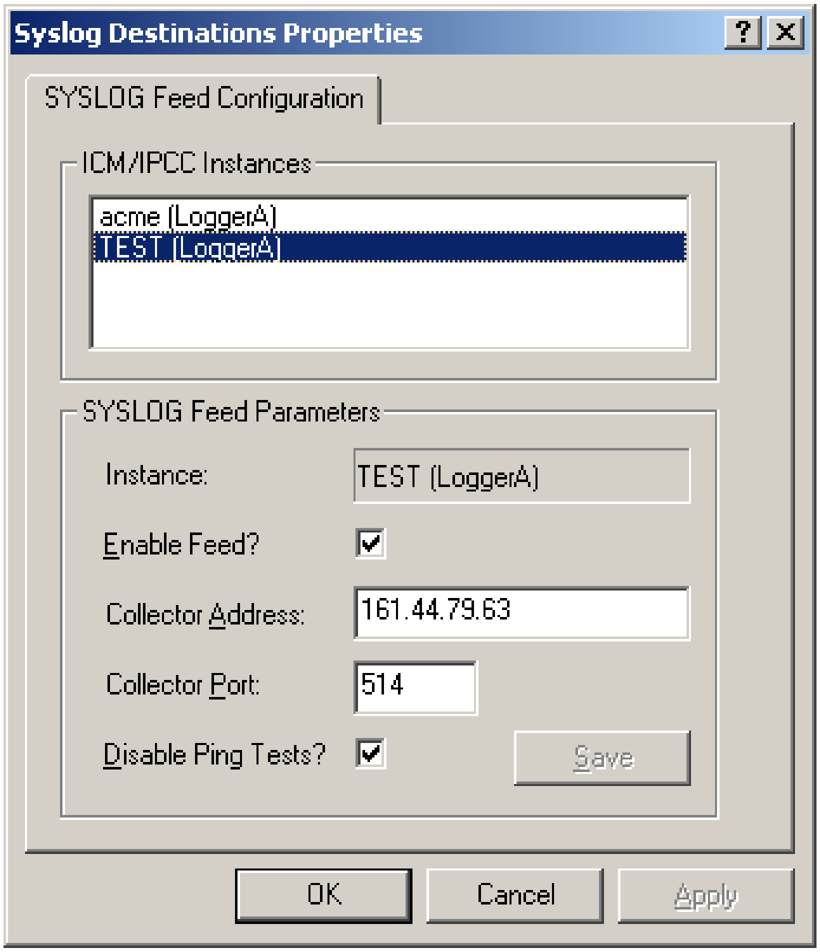
To configure syslog destinations:

1. Expand **Cisco SNMP Agent Management** in the left pane of the MMC snap-in.
2. Highlight **Syslog Destinations** in the left pane under Cisco SNMP Agent Management. ICM Instance Name, Feed Enabled, Collector Address, Port, and Ping Disabled columns appear in the right pane.
3. Right click on the white space in the right pane and choose **Properties**.

   A dialog box appears.
4. Select an ICM/CC Instance from the list box.
5. Check the **Enable Feed?** checkbox.
6. Enter an IP Address or Host Name in the **Collector Address** field.
7. Optionally, enter the collector port number on which the syslog collector is listening in the **Collector Port** field. The default port is 514.

8. Optionally, check the **Disable Ping Tests?** checkbox.
9. Click **Save**

Changes become effective when you click **OK**.



acme (LoggerA)
TEST (LoggerA)

Instance: TEST (LoggerA)

Enable Feed? ☑

Collector Address: 161.44.79.63

Collector Port: 514

Disable Ping Tests? ☑

**Figure 24: syslog Feed Configuration Dialog**

**IMPORTANT**: The Logger service MUST be cycled to start the flow of events from the syslog feed. The Node Manager picks up the configuration parameters from the registry and passes them to the CW2KFEED process when it invokes it. Changing the syslog parameters and killing the CW2KFEED process will not suffice because the Node Manager will simply restart it with the parameters it previously read from the registry. Unfortunately, a service recycle is required.

# 6 Unified ICM/CC Services and Processes

Each Unified ICM/CC component consists of one or more processes, which are enabled and managed by Node Manager. Each component has a separate Node Manager that is installed as a Windows service. All Node Manager services have the same process name, *nodeman.exe*.

## Services

The following table lists the processes running on a particular server. Note that in the Description column, the criticality of a process is denoted within brackets []. The key definitions are as follows:

| Key Definition | Description |
|---|---|
| **Critical**: | This process is critical to the operation of the component. Failure of the process renders the Contact Center application either dysfunctional or impaired. |
| **Critical/Optional**: | This process is optional (needed for a feature often enabled via configuration or during product installation). However, if the feature is enabled, the process is critical and failure of the process is likely to render the Contact Center application either dysfunctional or impaired. |
| **Optional**: | This process is optional (needed for a feature often enabled via configuration or during product installation). Failure of the process is unfortunate but will not impair the Contact Center application. |
| **Important**: | While failure of this process will not impair the Contact Center application, it will disable an important capability. |
| **Non-Critical**: | This process will be running on the server under normal operating conditions but its failure has little or no impact on the Contact Center application. |

Also note that an asterisk preceding the process name denotes that this process will appear in the SNMP CISCO-CONTACT-CENTER-APPS-MIB cccaComponentElmtTable.

**Table 6-1: Unified ICM/CC Processes**

| Component | Process | Description |
|---|---|---|
| Router | * router.exe | [Critical]: This is the primary Router process. |
| | * rtsvr.exe | [Critical]: Provides real-time data feed from the Router to the AW |
| | * mdsproc.exe | [Critical]: Message Delivery Service |
| | * ccagent.exe | [Critical]: Router component that manages communication links between the router and peripheral gateways. |
| | * dbagent.exe | [Critical]: Manages connections and transactions (configuration updates) from configuration tool(s). |
| | * testsync.exe | [Non critical] Provides interface for component test tools. |
| | * appgw.exe | [Optional/Critical]: The process that provides an interface for the Router to communicate with external applications. |
| | * dbworker.exe | [Optional/Critical]: The process that provides the interface for the Router to query external databases. |
| | * [NIC].exe | [Optional/Critical]: A separate process will be active for each Network Interface Controller (NIC) enabled during SETUP. The NIC process manages the interface to a |

| | | |
|---|---|---|
| | | telephony network.<br>The presence of a NIC process in a CCE deployment is very rare.<br>NIC process names: attnic.exe, cainnic.exe, netwrkcic.exe, crspnic.exe, cwcnic.exe, gktmpnic.exe, incrpnic.exe, mcinic.exe, gennic.exe, ntnic.exe, ntlnic.exe, sprnic.exe, ss7innic.exe, stentornic.exe, timnic.exe, unisourcenic.exe |
| Logger | * configlogger.exe | [Critical]: The process that manipulates configuration data. |
| | * histlogger.exe | [Critical]: The process that inserts historical data into TMP historical tables in the logger database. |
| | * recovery.exe | [Critical]: This process bulk copies historical data from the TMP historical tables to the actual historical tables. Recovers and synchronizes historical data with its partner logger during failover if loggers are running duplex. In addition, it is responsible for historical data purges in the logger database based on configured retention parameters. |
| | * replication.exe | [Critical]: The process that replicates data from the Logger to the Historical Data Server on an AW. |
| | * csfs.exe | [Critical]: The alarm/event processor. CSFS distributes alarms/events send via EMS to supported alarm/event feeds, for example, SNMP, syslog. CSFS stands for Customer Support Forwarding Service, which in ICM's infancy, forwarded events to a central monitoring location. |
| | * cw2kfeed.exe | [Optional]: The syslog event feed. This process acquires events from the CSFS process, formats them appropriately in accordance with the Berkley syslog protocol and sends the events to the configured collector.<br>If a syslog collector is not configured, this process will not be executing. |
| | * campaignmanager.exe | [Optional/Critical]: Outbound Option Campaign Manager. This process manages customer lists: provides customer records for every dialer in the enterprise; determines when customers should be called again; maintains the "Do Not Call" list in memory. The Campaign Manager also sends real time and historical data to the router and distributes configuration information to Dialer and Import processes.<br>This process is installed and executes on the "A" side Logger only. |
| | * baimport.exe | [Optional/Critical]: Outbound Option Import process. This process imports contact lists into the Outbound Option database; applies query rules to the contact table to build dialing lists; determines the GMT value for each phone based on the region prefix configuration.<br>This process is installed and executes on the "A" side Logger only. |
| | sqlservr.exe | [Critical]: Microsoft SQL server process |
| | sqlmangr.exe | [Critical]: Microsoft SQL server process |
| | sqlagent.exe | [Critical]: Microsoft SQL server process |
| Peripheral Gateway (PG) | * opc.exe | [Critical]: Open Peripheral Controller. This process acts as the brain for the peripheral gateway, including acting as a |

| | | |
|---|---|---|
| | | central collection and distribution point for all interaction with peripherals. OPC also ensures that all synchronization is accomplished with the other side. It also prepares and sends termination call detail (TCD) records as well as 5 minute and 30 minute reports. |
| | * mdsproc.exe | [Critical]: Message Delivery Service |
| | * pgagent.exe | [Critical]: MDS Peripheral Gateway component that manages the interface between the peripheral gateway and the central controller. |
| | * testsync.exe | [Non critical] Provides interface for component test tools. |
| | * eagtpim.exe | [Optional/Critical]: The CUCM peripheral interface manager process. This process manages the interface between OPC and the JTAPI Gateway. Multiple PIMs of the same type can be enabled for a PG. VRU PIMs and CUCM PIMs may be co-resident on a PG as well.<br>This is very common in CCE deployments but may not be present on all PGs.<br>There may be multiple instances of this process running. |
| | * acmipim.exe | [Optional/Critical]: The process is expected on the SCCE Gateway PG – this Peripheral Interface Manager is responsible for the communication interface between the parent instance and the child instance. |
| | * vrupim.exe | [Optional/Critical]: Peripheral Interface Manager process between OPC and a Voice Response Unit (VRU) or Interactive Voice Response (IVR).<br>There may be multiple instances of this process running. |
| | * mrpim.exe | [Optional/Critical]: The Media Routing Peripheral Interface Manager is the integration point for the Outbound Option Dialer, Cisco Email Manager (CEM), Cisco Collaboration Server (CCS) as well as the Email Interaction Manager (EIM) and Web Interaction Manager (WIM).<br>There may be multiple instances of this process running. |
| | * msgis.exe | [Optional/Critical]: Message Integration Service (MIS) which provides a mechanism to share call context data with a VRU. This process will only be present on a PG with a VRU PIM. |
| | * ctiosservernode.exe | [Critical]: The CTI OS Server process which manages connections from CTI clients (agent desktops), retains (real-time) data about agents and acts as the conduit for events and control messaging between CTI Server and CTI clients. |
| | * jtapigw.exe | [Critical]: JTAPI Gateway which manages the interface to the Unified Communications Manager IP PBX via the JTAPI client to the CTI Manager on the CM. On the other side, the JTAPI Gateway connects to the CUCM PIM and translates JTAPI messages and events into a format expected by the PIM. |
| | * ctisvr.exe | [Critical]: CTI Gateway (CTI Server) process that processes (GED-188) messages between CTI OS and |

| | | |
|---|---|---|
| | | OPC. (Note: in legacy implementations, CTI Server manages connections to CTI desktops.) |
| | IPPASvr.exe | [Optional/Critical] CAD: Cisco Browser and IP Phone Agent Service |
| | FCCServer.exe | [Optional] CAD: Cisco Chat Service |
| | CTI Storage Server.exe | [Optional/Critical] CAD: Cisco Enterprise Service |
| | LDAPmonSvr.exe | [Optional/Critical] CAD: Cisco LDAP Monitor Service |
| | LRMServer.exe | [Optional/Critical] CAD: Cisco Licensing and Resource Manager Service |
| | RPServer.exe | [Optional/Critical] CAD: Cisco Recording & Playback Service |
| | FCRasSvr.exe | [Optional/Critical] CAD: Cisco Recording and Statistics Service |
| | DirAccessSynSvr.exe | [Optional/Critical] CAD: Cisco Sync Service |
| | FCVoIPMonSvr.exe | [Optional/Critical] CAD: Cisco VoIP Monitor Service |
| | slurpd.exe | [Optional/Critical] CAD: Directory Replication Service |
| | slapd.exe | [Optional/Critical] CAD: Directory Services |
| | tomcat5.exe | [Optional/Critical] CAD: Tomcat Service |
| | badialer_ip.exe | [Optional/Critical]: Outbound Option: This is the Dialer process which implements a dialing algorithm and places calls to customers during an outbound campaign. The dialer monitors skill groups for agent availability and reserves agents via the MR PG. The dialer then informs the Campaign Manager of the result of each attempt to contact a customer. |
| Admin Workstation<br>(AW – Distributor) | * configlogger.exe | [Critical]: Processes inbound configuration data. |
| | * updateaw.exe | [Critical]: Updates the local AW configuration database with configuration data from the central controller. |
| | * rtclient.exe | [Critical]: Receives a real-time data feed (from a real-time distributor) and updates the local AW database. |
| | * rtdist.exe | [Critical]: Manages inbound real-time data from the real time server on the Router and distributes it to real-time clients. |
| | * replication.exe | [Critical]: Manages replicated historical data received from the Logger (HDS only) and inserts historical data in the HDS database. In addition, it is responsible for historical data purges in the HDS database based upon configured retention parameters. |
| | * cmsnode.exe | [Optional]: Configuration Management System (CMS). Manages configuration data for the ConAPI interface. This is a necessary interface (process) for the System CCE web configuration and the Agent Reskilling Tool. Thus, for System CCE, this is an important process. Also, if the customer has purchased Contact Center Management Portal (CCMP), CONAPI is also used. However, for a CCE deployment without CCMP, this process is not critical.<br>In recent version of CCE, cmsnode.exe will be running by |

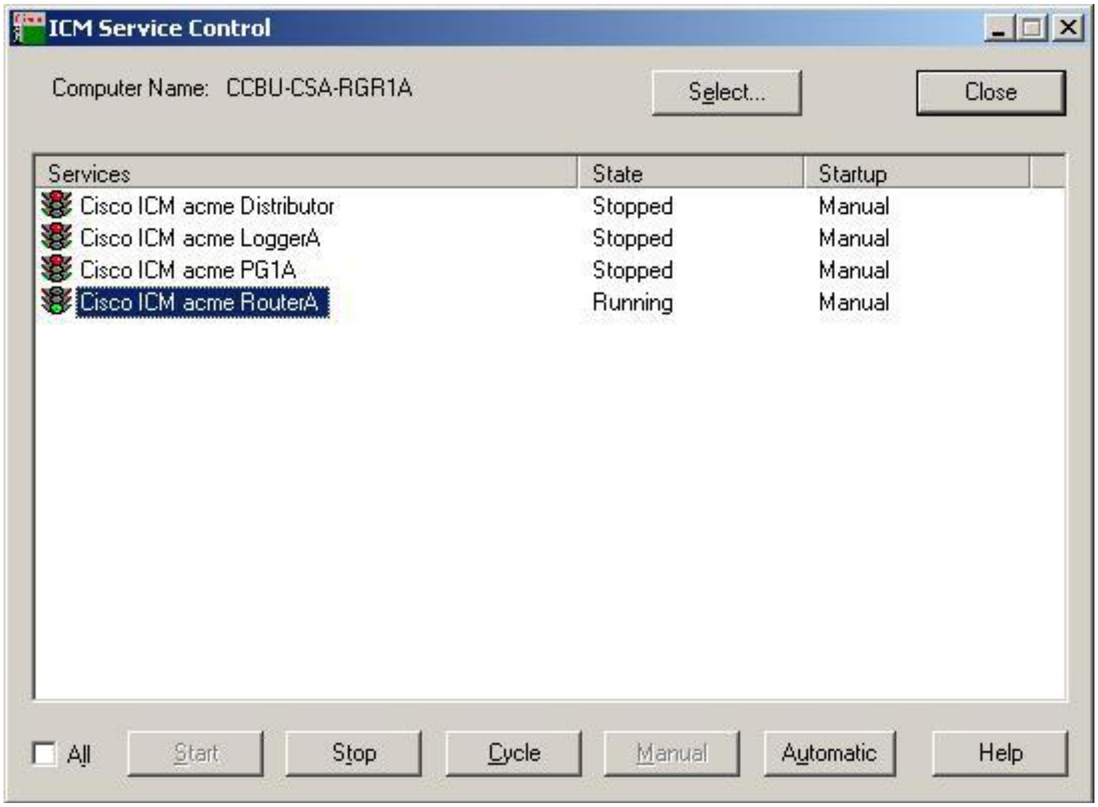
| | | |
|---|---|---|
| | | default but it is difficult for a management station to know whether it is necessary. Therefore, this is listed as Optional. |
| | * cms_jserver.exe | [Optional]: Configuration Management System (CMS) Jaguar Server. This process works with cmsnode.exe for CMS to provide Java interfaces for ConAPI.<br>In recent version of CCE, cms_jserver.exe will be running by default but it is difficult for a management station to know whether it is necessary. Therefore, this is listed as Optional. |
| | tomcat5.exe | [Optional/Critical]: Apache Tomcat servlet engine for SCCE web config. |
| | * iseman.exe | [Optional]: Internet Script Editor |
| | [Webview] | (If WebView server is co-resident on AW/HDS): WebView is a Java application that runs within the Java Virtual Machine. |
| | sqlservr.exe | [Critical]: Microsoft SQL server process |
| | sqlmangr.exe | [Critical]: Microsoft SQL server process |
| | sqlagent.exe | [Optional]: Microsoft SQL server process |
| All Nodes | nodeman.exe | [Critical]: Node Manager. This process monitors the status of all ICM/CC processes on the server; should a process terminate unexpectedly, the Node Manager automatically restarts that process. |
| | nmm.exe | [Critical]: Node Manager Manager. This process monitors the primary Node Manager (nodeman.exe) process; should the primary Node Manager (nodeman.exe) process terminate unexpectedly, the Node Manager Manager will restart it. |
| | snmpdm.exe | [Important]: SNMP master agent |
| | cccsnmpmgmt.exe | [Important]: SNMP agent management service – this service manages the SNMP agent infrastructure and restarts any agents that may terminate unexpectedly. It also ensures that the agent processes run at a reduced priority so as to not adversely impact server performance. |
| | msnsaagt.exe | [Important]: Microsoft native subagent adapter |
| | hostagt.exe | [Important]: HOST-RESOURCES-MIB subagent |
| | sappagt.exe | [Important]: SYSAPPL-MIB subagent |
| | cccaagent.exe | [Important]: CISCO-CONTACT-CENTER-APPS-MIB subagent |

## 6.1 Using the Local Desktop

Use ICM Service Control and the local registry to monitor Unified ICM/CC components and their processes.

## 6.2 ICM Service Control and Windows Task Manager

ICM Service Control displays the Node Manager service for each Unified ICM/CC component as well as its state and startup settings. Each Node Manager service appears in the following format: **Cisco ICM <instance> <component>**. As an example, the ICM Service Control window shown below lists information about the Node Manager services running on the local machine. The Router component Node Manager service is identified as **Cisco ICM acme RouterA**.



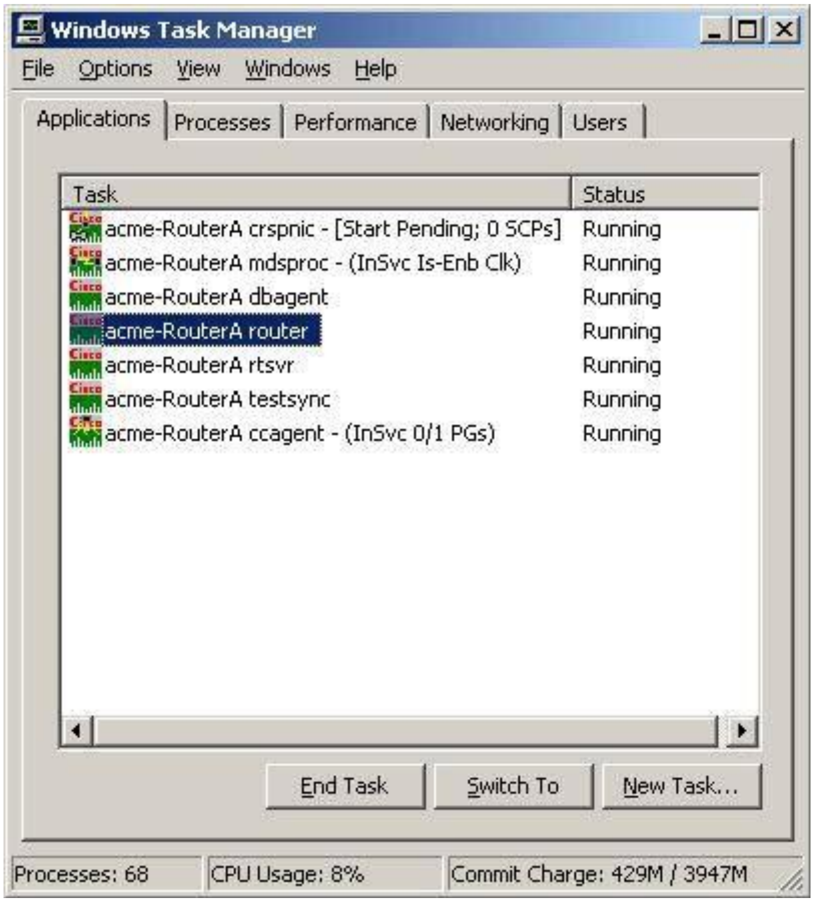
Computer Name: CCBU-CSA-RGR1A

| Services | State | Startup |
|---|---|---|
| Cisco ICM acme Distributor | Stopped | Manual |
| Cisco ICM acme LoggerA | Stopped | Manual |
| Cisco ICM acme PG1A | Stopped | Manual |
| Cisco ICM acme RouterA | Running | Manual |

Figure 25: ICM Service Control

Each running Unified ICM/CC process has an associated window on the desktop. The title bar in the window uniquely identifies each process in the following format:**<instance>-<component> <process>**. Note that some processes might display additional status information.

The Windows Task Manager Application tab corresponds to the Windows title bars for the Unified ICM/CC processes. The following illustration shows all the running processes for the RouterA component.

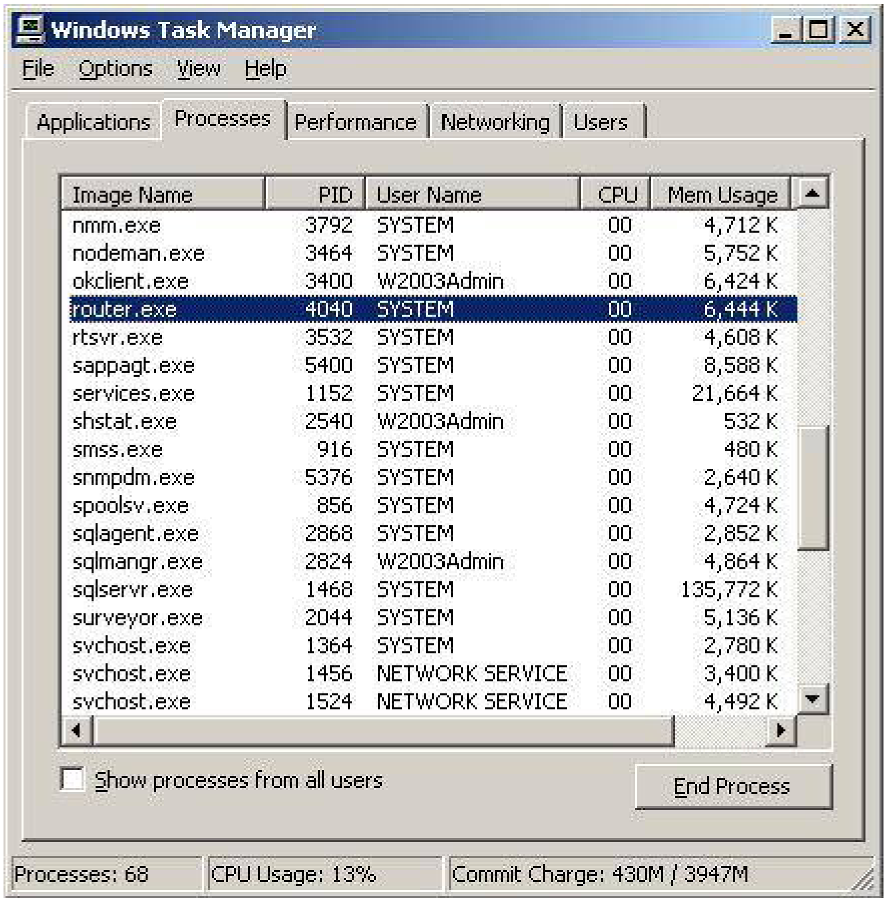

Windows Task Manager
File Options View Windows Help
Applications Processes Performance Networking Users
End Task
Switch To
New Task...

| Task | Status |
|---|---|
| acme-RouterA crspnic - [Start Pending; 0 SCPs] | Running |
| acme-RouterA mdsproc - (InSvc Is-Enb Clk) | Running |
| acme-RouterA dbagent | Running |
| acme-RouterA router | Running |
| acme-RouterA rtsvr | Running |
| acme-RouterA testsync | Running |
| acme-RouterA ccagent - (InSvc 0/1 PGs) | Running |

Processes: 68 CPU Usage: 8% Commit Charge: 429M / 3947M

**Figure 26: Windows Task Manager – Applications List**

From the **Applications** tab, right-click on a process and select the **Go To Process** option. Selecting this option causes the corresponding process entry to display in the Window Task Manager **Processes** tab. The following illustration is an example of the entry for the router.exe process that corresponds to acme-RouterA router shown in the Applications tab.


Windows Task Manager
File Options View Help
Applications Processes Performance Networking Users
Show processes from all users
End Process

| Image Name | PID | User Name | CPU | Mem Usage |
|---|---|---|---|---|
| nmm.exe | 3792 | SYSTEM | 00 | 4,712 K |
| nodeman.exe | 3464 | SYSTEM | 00 | 5,752 K |
| okclient.exe | 3400 | W2003Admin | 00 | 6,424 K |
| router.exe | 4040 | SYSTEM | 00 | 6,444 K |
| rtsvr.exe | 3532 | SYSTEM | 00 | 4,608 K |
| sappagt.exe | 5400 | SYSTEM | 00 | 8,588 K |
| services.exe | 1152 | SYSTEM | 00 | 21,664 K |
| shstat.exe | 2540 | W2003Admin | 00 | 532 K |
| smss.exe | 916 | SYSTEM | 00 | 480 K |
| snmpdm.exe | 5376 | SYSTEM | 00 | 2,640 K |
| spoolsv.exe | 856 | SYSTEM | 00 | 4,724 K |
| sqlagent.exe | 2868 | SYSTEM | 00 | 2,852 K |
| sqlmangr.exe | 2824 | W2003Admin | 00 | 4,864 K |
| sqlservr.exe | 1468 | SYSTEM | 00 | 135,772 K |
| surveyor.exe | 2044 | SYSTEM | 00 | 5,136 K |
| svchost.exe | 1364 | SYSTEM | 00 | 2,780 K |
| svchost.exe | 1456 | NETWORK SERVICE | 00 | 3,400 K |
| svchost.exe | 1524 | NETWORK SERVICE | 00 | 4,492 K |

Processes: 68 CPU Usage: 13% Commit Charge: 430M / 3947M

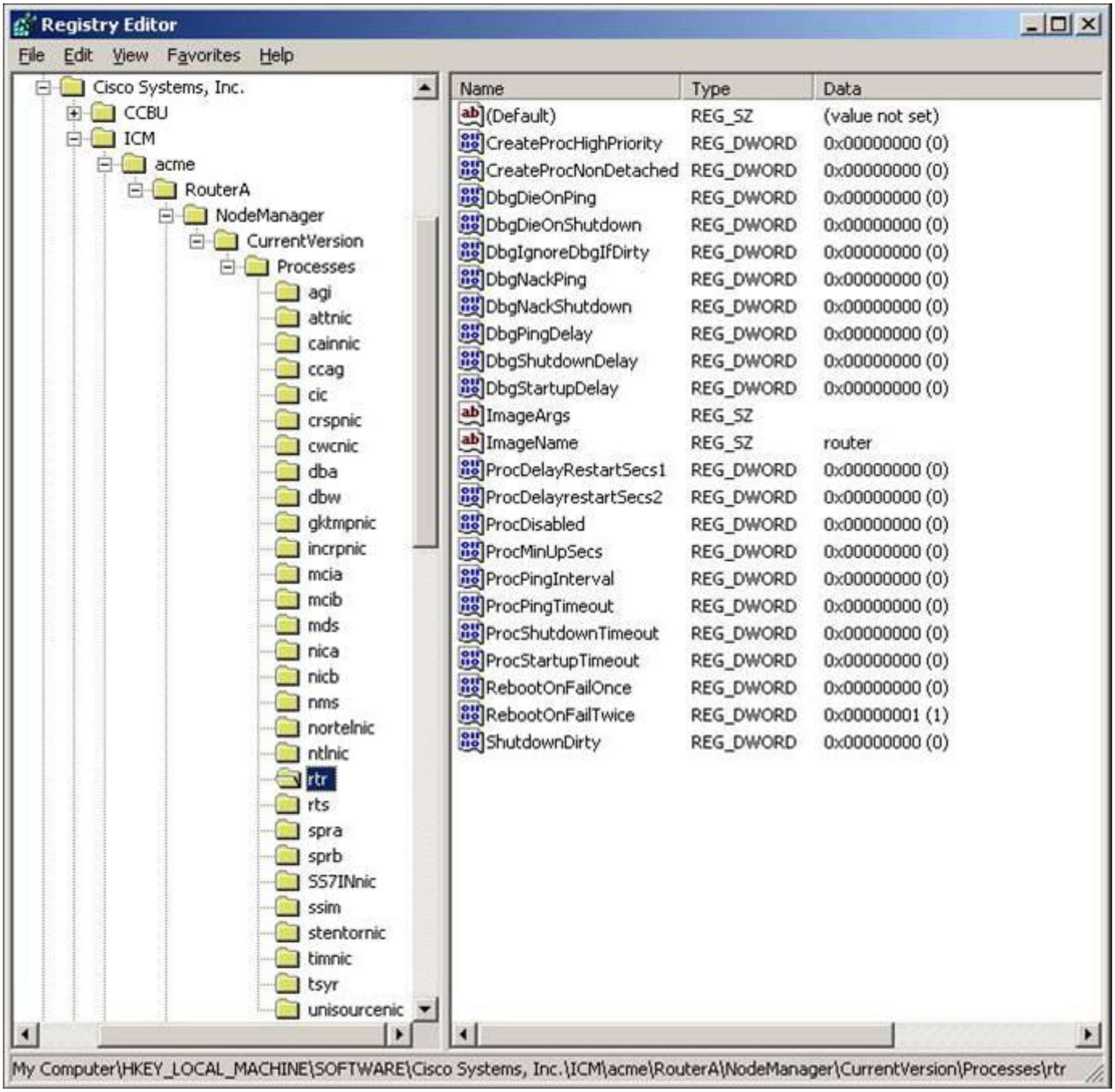
**Figure 27: Windows Task Manager - Process List**

## 6.3 Using the Local Registry

The Unified ICM/CC Windows registry hive contains the set of all installed components and their processes. However, to determine which processes are being managed, you need to traverse the Node Manager registry key for each component.

The following illustration shows the set of processes associated with the Cisco ICM acme RouterA component. Note that the key name is typically not the same as the process name. The key name for the router process is rtr; it appears highlighted in the navigation pane of the Registry Editor window. The process name, router, is contained in the ImageName value; it appears without the .exe file extension. If the ProcDisabled value is set to 0—as is the case for the router process—the process will be started and managed by the RouterA Node Manager process.

**Note:** The key name is typically not the same as the process name.



- Cisco Systems, Inc.
  - CCBU
  - ICM
    - acme
      - RouterA
        - NodeManager
          - CurrentVersion
            - Processes
              - agi
              - attnic
              - cainnic
              - ccag
              - cic
              - crspnic
              - cwcnic
              - dba
              - dbw
              - gktmpnic
              - incrpnic
              - mcia
              - mcib
              - mds
              - nica
              - nicb
              - nms
              - nortelnic
              - ntlnic
              - rtr
              - rts
              - spra
              - sprb
              - SS7INnic
              - ssim
              - stentornic
              - timnic
              - tsyr
              - unisourcenic

| Name | Type | Data |
|---|---|---|
| (Default) | REG_SZ | (value not set) |
| CreateProcHighPriority | REG_DWORD | 0x00000000 (0) |
| CreateProcNonDetached | REG_DWORD | 0x00000000 (0) |
| DbgDieOnPing | REG_DWORD | 0x00000000 (0) |
| DbgDieOnShutdown | REG_DWORD | 0x00000000 (0) |
| DbgIgnoreDbgIfDirty | REG_DWORD | 0x00000000 (0) |
| DbgNackPing | REG_DWORD | 0x00000000 (0) |
| DbgNackShutdown | REG_DWORD | 0x00000000 (0) |
| DbgPingDelay | REG_DWORD | 0x00000000 (0) |
| DbgShutdownDelay | REG_DWORD | 0x00000000 (0) |
| DbgStartupDelay | REG_DWORD | 0x00000000 (0) |
| ImageArgs | REG_SZ | |
| ImageName | REG_SZ | router |
| ProcDelayRestartSecs1 | REG_DWORD | 0x00000000 (0) |
| ProcDelayrestartSecs2 | REG_DWORD | 0x00000000 (0) |
| ProcDisabled | REG_DWORD | 0x00000000 (0) |
| ProcMinUpSecs | REG_DWORD | 0x00000000 (0) |
| ProcPingInterval | REG_DWORD | 0x00000000 (0) |
| ProcPingTimeout | REG_DWORD | 0x00000000 (0) |
| ProcShutdownTimeout | REG_DWORD | 0x00000000 (0) |
| ProcStartupTimeout | REG_DWORD | 0x00000000 (0) |
| RebootOnFailOnce | REG_DWORD | 0x00000000 (0) |
| RebootOnFailTwice | REG_DWORD | 0x00000001 (1) |
| ShutdownDirty | REG_DWORD | 0x00000000 (0) |

My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\ICM\acme\RouterA\NodeManager\CurrentVersion\Processes\rtr

**Figure 28: Registry Editor**

## 6.4 Using the Remote SNMP Management Station

In addition to the information available using the local desktop tools and registry, the Contact Center SNMP agent returns information about all Unified ICM/CC enabled processes regardless of whether they are running. This information is available from the *cccaInstanceTable*, *cccaComponentTable*, and *cccaComponentElmtTable*. The instance number and component index correlate a process to a specific instance and component.

The first example shows the entries for acme-RouterA router process. Note that the *cccaComponentElmtRunID* value, which is the process id, is valid if the *cccaComponentElmtStatus* is active, started, or standby.

```
cccaInstanceName.0 = acme
cccaComponentType.0.1 = router(1)
cccaComponentName.0.1 = RouterA
cccaComponentStatus.0.1 = started(4)
cccaComponentElmtName.0.1.5 = router
cccaComponentElmtRunID.0.1.5 = 4040
cccaComponentElmtStatus.0.1.5 = active(5)
```

The next example shows the entries for *acme-LoggerA*, the configlogger process. Note that the *cccaComponentElmtRunID* value, which is the process Id, is valid if the *cccaComponentElmtStatus* is not stopped (3).

```
cccaInstanceName.0 = acme
cccaComponentType.0.2 = logger(2)
cccaComponentName.0.2 = LoggerA
cccaComponentStatus.0.2 = stopped(3)
cccaComponentElmtName.0.2.8 = configlogger
cccaComponentElmtRunID.0.2.8 = 0
cccaComponentElmtStatus.0.2.5 = stopped(3)
```

# 7 Unified ICM/ Unified CC Trace Levels

Unified ICM and Unified CC application components write trace messages to trace log files on the local disk; these traces provide the following details about the operation of the component:

1. Error conditions (errors which may impair operation or performance are also reported in the Windows Event Log and sent via the syslog feed or, if sufficiently actionable, as SNMP notifications)
2. Debugging messages (to be used by troubleshooting engineers to diagnose problems)
3. Periodic performance metrics
4. Call state and/or call progress information
5. Configuration parameters or errors
6. Connectivity information (details about successful and failed connections)

The level of detail that is written to these trace logs can be controlled via numeric settings in the registry or via tools which interact directly with the application component to control tracing. The default settings (upon installation of the component) seek to balance performance with tracing detail with the scale tipped toward maximizing performance. Any increase in tracing levels will have a corresponding adverse impact on performance (for example, agent capacity, IVR port capacity, and inbound call load capacity) as additional computing resources will then be consumed by the resulting disk I/O.

The amount of tracing that is stored on the local disk is controlled by the tracing infrastructure; a sliding (fixed size) window of tracing is maintained whereby the oldest data is deleted to make room for the newest data. The size of this window can be controlled by carefully editing parameters in the Windows registry. The tracing window size is represented in bytes (disk consumption), not by a time duration.

Routine capacity utilization measurements will indicate the amount of computing resources that are available for added diagnostics (please see section 9 Capacity Planning for more details). If the deployment is already at high utilization, great care must be taken to understand the impact of enabling additional tracing to ensure that doing so does not adversely impact normal operation.

Before enabling additional tracing, it is highly recommended that the Health Monitoring Performance Counters be monitored while the tracing change is in effect to ensure that the server is not exceeding maximum thresholds. Please see section 8.1 Health Monitoring Counters for more details.

What follows is the recommended trace settings to be configured when initially engaged in diagnosing a problem. Note that TAC may suggest some differences based upon their initial impressions of the problem symptoms. These are proposed for those who wish to take a quick, proactive approach in getting the trace levels up as quickly as possible in order to gather as much useful information as possible as soon as possible.

Remember that TAC or BU engineers very likely may come back with additional settings based upon their initial log analysis.

Do not set what you believe to be maximum tracing – doing so could very well cause more problems than you had initially or even mask the problem by significantly changing timing.

## 7.1 EMS Log Compression

To collect logs that span a greater period of time, EMS log files from the CTI OS Server and the following PG components are zipped:

- CTI OS Server
- OPC-CCE

- OPC-TDM
- CTISVR
- EAGTPIM
- JTAPIGATEWAY
- VRUPIM

**Note:** These are the only components that currently support EMS log compression.

File compression is activated for the supported PG components when:

- You install one of the following patches:
  - 7.5(10)
  - 8.0(3)
  - 8.5(1)
  - 8.5(2)
- You run PG setup on PGs with the supported PG components.

File compression is activated on the CTI OS Server when:

- You install one of the following patches:
  - 7.5(10)
  - 8.0(3)
  - 8.5(1)
  - 8.5(2)
- You run CTI OS Server setup.

Note: These are the only components that currently support EMS log compression

### 7.1.1 Patch Installer - New Default Value for EMSAllLogFilesMax

For the components that support EMS file compression, EMSAllLogFilesMax is set to 2GB if the install drive has at least 25GB free disk space. The new value is set when you install the patch is or when run PG or CTI OS Sever setup on the supported components. The new default value of this registry key allows up to 2 GB of logs to be maintained (size taken post compression) on the system.

### 7.1.2 CTIOS Setup Information post patch

When you run the patch installer, EMSAllLogFilesMax is set to 2GB as mentioned above. When you run the CTI OS Sever setup, EMSAllLogFilesMax is unconditionally set to 2GB.

### 7.1.3 Dumplog

Dumplog has been updated to handle the compressed ems files and can be used in the normal way. Dumplog looks for gzip.exe in <Install Drive>\icm\bin to unzip compressed ems files before dumping logs. If one must dump logs from compressed EMS files (with .gz extension) outside of a PG or CTI OS Server, the EMS files can be unzipped prior to using dumplog.

### 7.1.4 EMS File Compression Control

To enable or disable compression of EMS log files, the EMSZipCompressionEnabled registry key in \EMS\CurrentVersion\Library\Processes\<node name> is used. It is recommended that you not modify this registry key. This key takes affect on components that support EMS file compression only.

### 7.1.5 Other registry keys

The following two other registry keys are also available in …\EMS\CurrentVersion\Library\Processes\<node name>

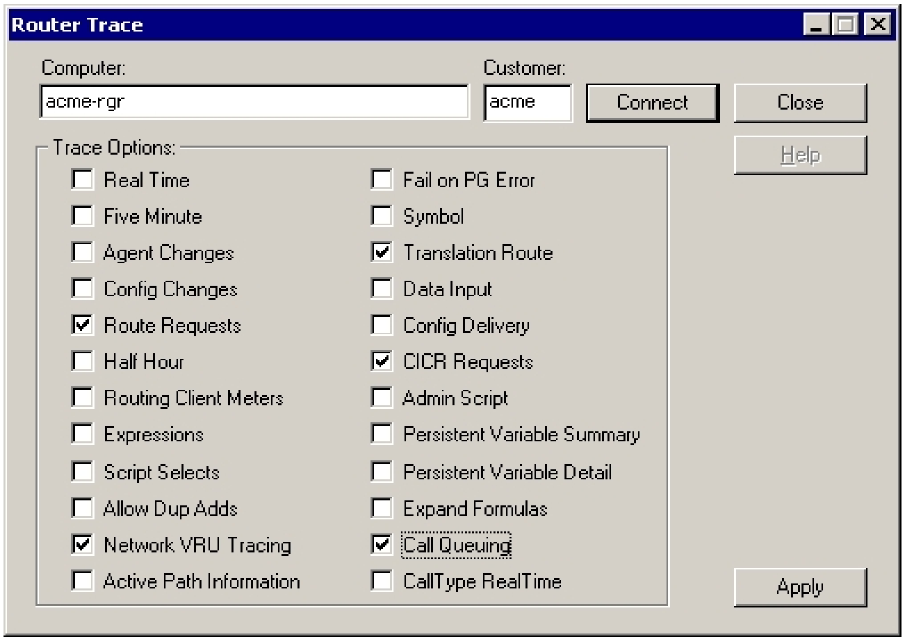
- EMSZipFormat
- EMSZipExtension

**Note:** You should **not** modify these registry keys.

## 7.2 Setting Router Tracing

Unified ICM/CC Router tracing is most easily set using the Router Trace utility. This is a single-form Windows GUI utility that is loaded on the Unified ICM/CC server. It is most easily launched by connecting to the server via remote desktop (or go to the local console); invoke RTRTRACE from ICM\BIN:

```
C:> \icm\bin\rtrtrace
```



Computer: acme-rgr
Customer: acme

Trace Options:

□ Real Time
□ Five Minute
□ Agent Changes
□ Config Changes
☑ Route Requests
□ Half Hour
□ Routing Client Meters
□ Expressions
□ Script Selects
□ Allow Dup Adds
☑ Network VRU Tracing
□ Active Path Information
□ Fail on PG Error
□ Symbol
☑ Translation Route
□ Data Input
□ Config Delivery
☑ CICR Requests
□ Admin Script
□ Persistent Variable Summary
□ Persistent Variable Detail
□ Expand Formulas
☑ Call Queuing
□ CallType RealTime

**Figure 29: Router Trace Utility**

Typically when a call routing failure occurs, the basic traces should at the minimum be "Route Requests" and "Translation Route" (if translation routing is used).

In addition, other tracing should be enabled depending on the specific problems seen.

- If using any type of VRU, enable "Network VRU Tracing"
- If NAM-CICM (Hosted), enable "CICR Requests"
- If queuing issues are suspected, enable "Call Queuing"
- For Call Type Reporting Problems, enable “Call Type Real Time”
- For Agent Issues, enable “Agent Changes”

All trace settings using "RTRTRACE" take effect immediately in the router.

Apart from this, it is possible to "observe" specific status of call routing, call type, skill group and schedule target variables using RTTEST command. This can be invoked as:

```
rttest /cust <instance>
```

Also, the RTTEST "watch" command is very useful.

## 7.3 Setting OPC Tracing

Unified ICM/CC OPC tracing is most easily set using the OPCTEST utility. This is a command-line utility so remote desktop or local console access is required.

Command Syntax (launch):

```
        C:> opctest /cust <instance> /node <node>
```

Where <instance> is the Unified ICM/CC instance name and <node> is the desired node name (for example, /cust cust1 /node PG1A).

Once invoked, you will be presented with an opctest: prompt where commands may be entered according to the syntax expected.  Entering a'?' at the opctest: prompt will display all possible commands, however, understand that OPCTEST is a very powerful utility that if used incorrectly, could have a very negative effect on a production system in operation. Please do not execute a command against a production system unless you are absolutely certain of the impact it can introduce.

The following commands are recommended for altering default trace levels.  Again, it is highly recommended that you first understand your current utilization to ensure there is sufficient capacity to accommodate the added tracing.

### 7.3.1 General Diagnostics

```
opctest:debug /on
```

### 7.3.2 Diagnosing Network Transfer Issues

```
opctest:debug /on

opctest:debug /NCT
```

### 7.3.3 Diagnosing Multi Media Issues

```
opctest:debug /on

opctest:debug /task /passthru
```

### 7.3.4 Diagnosing VRU PG Issues

```
opctest:debug /on

opctest:debug /passthru
```

The default is:

```
opctest:debug /routing /agent /closedcalls /cstacer /rcmsg /tpmsg /simplified
/inrcmsg
```

and

```
EMSTracemask = 0x40
```

EMSTracemask is reset in the Windows registry.

TAC will direct you to alter or add additional tracing based upon the analysis of collected logs.

### 7.3.5 Restoring Default Trace Levels

```
opctest:debug /on
```

This parameter turns on the /default tracing, modifes the EMSTracemask to 0x40, and turns off all other enabled tracing.

### 7.3.6 Displaying Trace Levels

```
opctest:debug /showtrace
```

This parameter displays current trace levels enabled on the peripheral. Modifying the trace levels in opctest echo the resulting tracing levels.

| | |
|---|---|
| Registry Key | **HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\<instance>\ARSGW\EMS\CurrentVersion\Library\Processes\arsgw1\EMSTraceMask** |
| Item | **EMSTraceMask** |
| Value | **0x80023fff**<br>The value of **0x80023fff** will provide sufficient tracing information to troubleshoot most issues |
| Registry Key | **HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\<instance>\ARSGW\EMS\CurrentVersion\Library\Processes\PG\CurrentVersion\ARS\ARSgw1\ARSData\Dynamic\EMSTraceMaskCollectMsg** |
| Item | **EMSTraceMaskCollectMsg** |
| Value | **0xffffffff**<br>The value of **0xffffffff** will provide sufficient tracing information to troubleshoot most issues |

### 7.3.7 OPC Capture File Compression

This feature is installed when you install the 7.5(10) patch.

OPC Capture File compression increases the capture message retention for the OPC component for Unified CCE and TDM.

To enable or disable compression of OPC Capture files use the new registry key, CaptureZipCompressionEnabled located in HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\ICM\<ICM-Instance>\<PG-Instance>\PG\CurrentVersion\OPC].

You must set the value of this registry to 1 to enable this feature.

> **Note:** Additional registry keys CaptureZipFormat and CaptureZipExtension are added under [HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\ICM\<ICM-Instance>\<PG-Instance>\PG\CurrentVersion\OPC] to support the compression mechanism.
>
> Do not modify these registry keys.

You must unzip the OPC Capture files before using it with the OPC Playback tool.

## 7.4 Setting Unified CCM PIM Tracing

To reset trace levels with the Unified Communications Manager Peripheral Interface Manager component (for example, “EAGTPIM”) use the PROCMON (process monitoring) utility. This is a command-line utility so remote desktop or local console access is required.

**Table 7-1: Setting Unified CCM PIM Tracing**

| | |
|---|---|
| Command Syntax (launch) | **C:> procmon <*instance*> <*node*> pim<*pim number*>** |
| Example | **C:> procmon acme PG1A pim1** |
| Commands | >>>**debug /on** |

### 7.4.1 ARS Gateway Registry Trace Settings

**Table 7-2: Setting ARS Gateway Registry Tracing**

| | |
|---|---|
| Regis try Key | **HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\<instance>\ARSGW\EMS\CurrentVersion\Library\Processes\arsgw1\EMSTraceMask** |
| Item | **EMSTraceMask** |
| Value | **0x80023fff**<br>The value of **0x80023fff** will provide sufficient tracing information to troubleshoot most issues |
| Registry Key | **HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\<instance>\ARSGW\EMS\CurrentVersion\Library\Processes\PG\CurrentVersion\ARS\ARSgw1\ARSData\Dynamic\EMSTraceMaskCollectMsg** |
| Item | **EMSTraceMaskCollectMsg** |
| Value | **0xffffffff**<br>The value of **0xffffffff** will provide sufficient tracing information to troubleshoot most issues |

### 7.4.2 ARS PIM Trace Settings

**Table 7-3: Setting ARS PIM Tracing**

| | |
|---|---|
| Command Syntax (launch) | **C:> procmon <*instance*> <*node*> pim<*pim number*>** |
| Example | **C:> procmon acme PG1A arspim1** |
| Commands | **debug /level 2** |

## 7.5 Setting JTAPI Gateway Tracing

As with the Unified Communications Manager PIM, resetting trace levels with the Unified CC JTAPI (Java Telephony Applications Programming Interface) Gateway component (for example, “JTAPIGW”) is most easily accomplishing using the PROCMON (process monitoring) utility. This is a command-line utility so remote desktop or local console access is required.

**Table 7-4: Setting JTAPI Gateway Tracing**

| | |
|---|---|
| Command Syntax (launch) | **C:> procmon <instance> <node> jgw<jtapigw number>** |
| Example | **C:> procmon acme PG1A jgw1** |
| Commands | >>>**trace * /off**<br>>>>**debug /on** |

### 7.5.1 Setting JTAPI Gateway Default Tracing

The default tracing for JTAPI gateway consists of a set of tracing levels that currently exist.

To enable the default tracing only, enter the following commands in PROCMON:

- **trace * /off**

**Note**: **debug /on** does not turn off non-default tracing so this is needed first.

- **debug /on** This enables default tracing only.

To turn off debug tracing, enter the following command in PROCMON:

- **debug /off** This turns off default tracing only. All other tracing is not affected.

## 7.6 Setting CTI Server Tracing

Resetting trace levels with the Unified ICM/CC CTI Server (for example, CTI Gateway or CG) is accomplishing by altering the trace mask saved in the Windows registry. Use the Windows REGEDIT utility to change this numeric value.

**Table 7-5: Setting CTI Server Tracing**

| | |
|---|---|
| Registry Key | **HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\<instance>\CG#A/B\EMS\CurrentVersion\ Library\Processes\ctisvr** |
| Example | **HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\acme\CG1A\EMS\CurrentVersion\Library\ Processes\ctisvr** |
| Item | **EMSTraceMask** |
| Value | **F0 (hex)**<br>This is the default value. The value of F0 provides sufficient tracing information to troubleshoot most issues |

### 7.6.1 Setting CTI Server Default Tracing

The default tracing level for CTI Server is EMSTraceMask = 0xF0. Do not enable any other tracing at the default trace level. EMSUserData should be NULL.

**Procmon debug commands:**

**debug /on** sets the EMSTraceMask to the default value of 0xF0 and NULL out EMSUserData. No other command is needed to set default tracing.

**debug /off** sets EMSTraceMask to 0x00 and NULL out EMSUserData.

## 7.7 Setting CTI OS Tracing

To reset trace levels with the Unified ICM/CC CTI Object Server (CTI OS) modify the trace mask saved in the Windows registry. Use the Windows REGEDIT utility to change this numeric value.

**Table 7-6: Setting CTI Server Tracing**

| | |
|---|---|
| Registry Key | **HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\<instance>\CTIOS\EMS\CurrentVersion\ Library\Processes\ctios** |
| Example | **HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\acme\CTIOS\EMS\CurrentVersion\Library\ Processes\ctios** |
| Item | **EMSTraceMask** |
| Value | **60A0F (hex)(recommended troubleshooting trace value)**<br>Increasing the trace levels (other than the Default (0x00060A0F)) will impact the CTIOS Server performance. High Tracemask needs to be reverted to the default trace levels after collecting the required logs. |

## 7.8 Setting VRU PIM Tracing

To reset trace levels with the Unified ICM/CC VRU Peripheral Interface Manager (PIM), modify the trace mask and user data values saved in the Windows registry. Use the Windows REGEDIT utility to change these numeric values.

**Table 7-7: Setting VRU PIM Tracing**

| | |
|---|---|
| Registry Key | **HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\<instance>\PG#A/B\EMS\CurrentVersion\ Library\Processes\pim#** |
| Example | **HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\acme\PG2A\EMS\CurrentVersion\Library\ Processes\pim1** |
| Item | **EMSUserData** |
| Value | **7F F7 E0 (hex)** |
| Item | **EMSTraceMask** |
| Value | **0 (zero)** |

When you collect the trace logs, collect both VRU PIM trace logs and the VRU trace capture file. VRU trace capture files can be obtained by running the VRUTRACE tool in the following directory:

`\icm\<inst>\pg#a/b\vrucap` (Ex: `\icm\acme\pg2a\vrucap`)

### 7.8.1 Setting VRU PIM Default Tracing

The default tracing for VRU PIM consists of a set of tracing levels that currently exist.

**Procmon debug commands:**

**debug /off** turns off all tracing

**debug /on** enables default tracing only and turns off any previously enabled tracing

## 7.9 Setting Trace File Retention Parameters

There are several Windows registry values that can be modified to adjust the trace log retention parameters, for example, increase the amount of trace data – extend the trace retention window. This is done by using the Windows REGEDIT utility.

Unified ICM/CC Event Management System (EMS) tracing is stored in a binary format in a set of files located in a directory on the local drive following a specific structure:

[Drive]:\icm\<instance>\<node>\logfiles

Example:

C:\icm\acme\pg1a\logfiles

Trace log files are formatted using a consistent format:

Process_YYMMDD_HHMMSS.ems

Example:

opc_090713_123025.ems

Which is an OPC trace log file created 13 July, 2009 at 12:30:25.

Under the control of the Event Management System, the following rules apply while traces are being written to the trace log files:

When the size of this file is greater-than or equal-to the maximum (configured) size that a single EMS trace log file is allowed, the file is closed and a new file is created.

If the maximum number of trace log files for this process is greater-than the maximum (configured) number of trace log files, then the oldest trace log file is deleted.

If the total combined size of all process trace log files is greater-than or equal-to the maximum (configured) total size of all of the trace log files for this process, then the oldest trace log files are deleted until the total is less-than the configured maximum.

The following registry item values can be changed to increase or decrease the amount of disk space allocated for a single process:

Registry Key:

```
HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\<instance>\<node>\EMS\CurrentVersion\
Library\Processes\<process>
```

Example:

```
HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\acme\PG1A\EMS\CurrentVersion\Library\
Processes\opc
```

Items:

**`EMSLogFileMax`**

The maximum size (in bytes) of a single trace log file for this process

**`EMSLogFileCountMax`**

The maximum number of trace log files permitted for this process

**`EMSAllLogFilesMax`**

The total space allowed for all trace log files (combined size) for this process

Note: **`EMSLogFileMax`** multiplied by **`EMSLogFileCountMax`** may be greater-than **`EMSAllLogFilesMax`** and it often is by default; this is to ensure trace log files created by frequent process restarts (where a number of small trace log files will be created) will not be lost when the max count is exceeded but very little disk space is used. **`EMSAllLogFilesMax`** is used to guarantee that under any circumstances, the maximum amount of disk space allocated is never exceeded.

The default values of these items are evaluated with every release of Unified ICM/CC to determine the optimal limits based on disk usage of the application and typical disk capacity of servers available at the time of release. In nearly all cases, the default values are increased over time as disk drive sizes increase.

# 8 Performance Counters

## 8.1 Health Monitoring Counters

The following table lists the performance counters that should be watched on a regular basis to determine the health of the contact center application.

**Table 8-1: Performance Counters - Health Monitoring**

| Performance Object | Counter Name (Instance) | Type | Units (Range) | Threshold Green | Threshold Yellow | Threshold Red |
|---|---|---|---|---|---|---|
| Cisco ICM Router | Agents Logged On [1] | Int32 | # agents | ** | ** | ** |
| | The number of (contact center) agents currently logged on. (See note 1 below) | | | | | |
| Cisco ICM Router | Calls In Progress [1] | Int32 | # calls | ** | ** | ** |
| | The number of calls currently in progress (being controlled by the CCE application). (See note 1 below) | | | | | |
| Cisco ICM Router | Calls/sec [1] | Int32 | Calls per second | ** | ** | ** |
| | The (calculated) inbound call rate measured in the number of calls received per second. (See note 1 below) | | | | | |
| Processor | % Processor Time (_Total) | Int32 | Percentage (0 - 100%) | < 50% | 50% - 60% | > 60% (sustained) |
| | Primary indicator of processor activity; displays the average percentage of CPU busy time observed during the sample interval. | | | | | |
| System | Processor Queue Length | Int32 | # threads | < 2 * #CPUs | - | >= 2 * #CPUs (sustained) |
| | Number of threads in the processor queue waiting to be serviced. Note that Microsoft states that Processor Queue Length is OK up to 10 per CPU. This may be the case for non-realtime applications but Unified CC performance will be impacted if this queue length is excessive for a sustained period of time. Timeouts are likely if the server becomes CPU bound or a single application (or process) monopolizes the CPU. | | | | | |
| Memory | Available Bytes | Int32 | Percentage (0 - 100%) | > 30% | 20% - 30% | < 20% |
| | Amount of physical memory available to running processes; threshold values are a percentage of physical memory. This is a snap shot – not a running average. Sustained samples below 20% (available) may be indicative of a memory leak. | | | | | |
| Memory | Pages / sec | Int32 | # page faults | < 10 | >= 10 | > 10 (sustained) |
| | Pages/sec is the rate at which pages are read from or written to disk to resolve hard page faults. Excessive page faults adversely impacts performance – root cause must be investigated. | | | | | |
| Physical Disk | Avg. Disk Queue Length (_Total) | Float | Average # read/write requests | < 1.5 | - | >= 1.5 (sustained) |
| | Average number of both read and write requests that were queued for the selected disk during the sample interval. | | | | | |
| Physical Disk | % Disk Time (_Total) | Int32 | Percentage (0 - | < 60% | 60% - 80% | > 80% |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | 100%) | | | |
| | Percentage of elapsed time that the disk drive was busy servicing read or write requests. | | | | | |
| Network Interface | Bytes Total/sec | Int32 | Percentage (0 - 100%) | < 25% | 25% - 30% | > 30% |
| | Rate at which bytes are sent and received over each network adapter. Threshold values are a percentage of available bandwidth. | | | | | |
| Network Interface | Output Queue Length | Int32 | # packets in queue | 0 | 1 | > 1 (sustained) |
| | Length of the output packet queue (in packets). If too large, there are delays and the bottleneck should be found and eliminated. | | | | | |
| SQLServer:Buffer Manager | Buffer cache hit ratio | Int32 | Percentage (0 - 100%) | > 90% | - | < 90% |
| | This counter shows the percentage of pages in the buffer pool without needing to read from disk. Thresholds are expressed as a percentage of "hits": instances in which the requested page was found in the cache.<br>This counter is typically a good indicator of whether there is sufficient RAM installed in the server.<br>If you are using SQL Server Standard Edition in a large enterprise or hosted environment and this counter (as well as other performance counters) is not within the recommended range, upgrading SQL Server to Enterprise Edition may be the next step.  Note that upgrading SQL Server to Enterprise Edition requires and upgrade of the operating system to Windows Server 2003 Enterprise Edition as well. | | | | | |

*** No quantifiable Green/Yellow/Red threshold values for these counters – deployment specific*

[1] These counters are also quite useful for long-term trending to determine whether there are capacity issues now or whether there will be in the future.  The counters values can be compared to other PerfMon counters (CPU, Memory, Disk, NIC). Relationships and cause/effect analysis can greatly assist in confirming existing or predicting upcoming capacity/performance problems.

Threshold values are not monitored by the application itself – alarms are not generated if threshold are exceeded.  The responsibility for polling and threshold alarming is extended to the management station.

## 8.2 Diagnostic Counters – Automatic Collection

The following counters values are sampled and collected automatically (by the Node Manager Manager) – counter values are stored in a disk file on the server.  Counter values are sampled at a "one minute" interval.  Data files contain a rolling window of counter values – older data is discarded in lieu of new data.  Data is stored in multiple files (maximum size is 1 MB each) and a maximum of 45 days of data is saved.

| | |
|---|---|
| Data file location: | **`\icm\log`** |
| File naming convention: | **`Perf_MACHINENAME_YYYYMMDDHHMMSS.CSV`** |
| Where: | `MACHINENAME` is the assigned Windows computer name. |
| | `YYYYMMDD` is the year, month, day the file was created |
| | `HHMMSS` is the hour:minute:second the file was created |

Analysis of these counter values is beneficial when diagnosing a problem with a Unified CCE application component.

**Table 8-2: Performance Counters - Diagnostics**

| Component | Counter Name | Type | Units (Range) |
|---|---|---|---|
| Processor | % Processor Time (_Total) | Int32 | Percentage (0 – 100%) |
| % Processor Time is the percentage of elapsed time that the processor spends to execute a non-Idle thread. It is calculated by measuring the duration of the idle thread is active in the sample interval, and subtracting that time from interval duration. (Each processor has an idle thread that consumes cycles when no other threads are ready to run). This counter is the primary indicator of processor activity, and displays the average percentage of busy time observed during the sample interval. It is calculated by monitoring the time that the service is inactive, and subtracting that value from 100%. | | | |
| Process | Handle Count (_Total) | Int32 | # handles |
| The total count of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process. | | | |
| Memory | Page Faults / sec | Int32 | # faults |
| Page Faults/sec is the average number of pages faulted per second. It is measured in number of pages faulted per second because only one page is faulted in each fault operation, hence this is also equal to the number of page fault operations. This counter includes both hard faults (those that require disk access) and soft faults (where the faulted page is found elsewhere in physical memory.) Most processors can handle large numbers of soft faults without significant consequence. However, hard faults, which require disk access, can cause significant delays. | | | |
| Memory | Committed Bytes | Int32 | # bytes |
| Committed Bytes is the amount of committed virtual memory, in bytes. Committed memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average. | | | |
| Memory | Pages / sec | float | # pages per second |
| Pages/sec is the rate at which pages are read from or written to disk to resolve hard page faults. This counter is a primary indicator of the kinds of faults that cause system-wide delays. It is the sum of Memory\\Pages Input/sec and Memory\\Pages Output/sec. It is counted in numbers of pages, so it can be compared to other counts of pages, such as Memory\\Page Faults/sec, without conversion. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) non-cached mapped memory files. | | | |
| System | Threads | Int32 | # threads |
| Threads is the number of threads in the computer at the time of data collection. This is an instantaneous count, not an average over the time interval. A thread is the basic executable entity that can execute instructions in a processor. | | | |
| System | Processor Queue Length | Int32 | # threads |
| Processor Queue Length is the number of threads in the processor queue. Unlike the disk counters, this counter counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload. | | | |
| System | Processes | Int32 | # processes |
| Processes is the number of processes in the computer at the time of data collection. This is an instantaneous count, not an average over the time interval. Each process represents the running of a program. | | | |

## 8.3 Diagnostics Counters – All Components

If a problem occurs on a Unified CC/ICM component, to further diagnose the problem, these counters should be enabled using the Windows PerfMon tool. At first, set the interval to 15 seconds and collect a sample large enough before, during and after the problem.  Save the data in .CSV format for simple import into Microsoft Office Excel. Attach the file to the TAC case.

If the data does not provide enough resolution to diagnose root cause, increase the interval to 5 seconds. A sample interval more frequent than 3 seconds should not be attempted.

**Table 8-3: Diagnostic Counters - All Components**

| Performance Object | Instance | Counter Name |
|---|---|---|
| LogicalDisk | _Total | Avg. Disk Queue Length |
| LogicalDisk | C: | Avg. Disk Queue Length |
| LogicalDisk | <DB Drive> | Avg. Disk Queue Length |
| Network Interface | <NIC Name> | Packets Outbound Discarded |
| PhysicalDisk | _Total | Disk Transfers / sec |
| Process | _Total | Page Faults / sec |
| Process | _Total | Virtual Bytes |
| Process | _Total | Working Set |
| Processor | _Total | Interrupts / sec |
| Process | <virus scanner> | % Processor Time |
| Process | <virus scanner> | Page Faults / sec |
| Process | <virus scanner> | Virtual Bytes |
| Process | <virus scanner> | Working Set |

## 8.4 Diagnostic Counters – Logger / AW / HDS

These counters are intended for Unified CC/ICM components that have a SQL Server database installed.  Please note the SQL Server counters listed on the next slide.

Set the initial sample frequency to 15 seconds. If not sufficient resolution, go down to a 5 second interval.

**Table 8-4: Diagnostic Counters - Logger, AW, HDS**

| Performance Object | Instance | Counter Name |
|---|---|---|
| Physical Disk | <DB Drive> | % Disk Time |
| Physical Disk | <DB Drive> | Avg. Disk Queue Length |
| Physical Disk | <DB Drive> | Disk Transfers / sec |
| Process | ** See note | % Processor Time |
| Process | ** See note | Page Faults / sec |
| Process | ** See note | Virtual Bytes |
| Process | ** See note | Working Set |
| Process | sqlservr | % Processor Time |
| Process | sqlservr | Page Faults / sec |
| Process | sqlservr | Virtual Bytes |
| Process | sqlservr | Working Set |

| | | |
|---|---|---|
| *** Note**: | Logger Processes: | configlogger, histlogger, recovery, replication |
| | AW/HDS Processes: | configlogger, recovery, replication, rtclient, rtdist |

## 8.5 Diagnostic Counters – SQL Server

The listed counters are available on those servers on which a Unified CC/ICM database is installed.

Set the initial sample frequency to 15 seconds. If not sufficient resolution, go down to a 5 second interval.

**Table 8-5: Diagnostic Counters - SQL Server**

| Performance Object | Instance | Counter Name |
|---|---|---|
| SQLServer:Access Methods | | Full Scans / sec |
| SQLServer:Buffer Manager | | Buffer cache hit ratio |
| SQLServer:Buffer Manager | | Page reads / sec |
| SQLServer:Buffer Manager | | Page writes / sec |
| SQLServer:Buffer Manager | | Stolen pages |
| SQLServer:Databases | _Total | Transactions / sec |
| SQLServer:Databases | csco_awdb [1] | Transactions / sec |
| SQLServer:Databases | csco_hds [1] | Transactions / sec |
| SQLServer:General Statistics | | User Connections |
| SQLServer:Latches | | Average Latch Wait Time (ms) |
| SQLServer:Locks | _Total | Lock Timeouts / sec |
| SQLServer:Locks | _Total | Number of Deadlocks / sec |
| SQLServer:Memory Manager | | Memory Grants Pending |

[1] Where “csco” is the Unified ICM/CC instance name

## 8.6 Diagnostic Counters – WebView

Set the initial sample frequency to 15 seconds. If not sufficient resolution, go down to a 5 second interval.

**Table 8-6: Diagnostic Counters - WebView**

| Performance Object | Instance | Counter Name |
|---|---|---|
| Process | jagsrv | % Processor Time |
| Process | jagsrv | Page Faults / sec |
| Process | jagsrv | Virtual Bytes |
| Process | jagsrv | Working Set |
| Process | java | % Processor Time |
| Process | java | Page Faults / sec |
| Process | java | Virtual Bytes |
| Process | java | Working Set |
| Web Service | _Total | Current Connections |

# 9 Capacity Planning

The purpose of capacity planning is to:

- **Determine Current Solution Capacity -** "How close to the ceiling am I today?"
- **Estimate Growth Potential -** "With current growth plans, when will I need to upgrade hardware?"
- **Answer "What If" Scenarios -** "What if I added 200 agents?"

Capacity planning is not a one-time task—it should be part of routine contact center operations. A reliable capacity management plan will help prevent outages because the data will support proactive modifications to the deployment that will ultimately prevent a particular outage. How might this happen?

Example:

> When the system was initially designed and deployed, it had been sized for a specific number of agents with a certain number of skills groups configured per agent. At that time, there was sufficient room to accommodate modest growth. As time went on, small changes occurred with no hint of an issue in capacity – agents were added, skill groups were added. There was no capacity management plan in place and utilization increased with no one being aware. Eventually, utilization was near maximum thresholds where in the midst of a busy period, an unexpected outage occurs. If a capacity management plan was in place, the increase in utilization would have been seen with each change to the system. As utilization increased nearing maximum capacity, either additional changes would have been curtailed or an upgrade of hardware would have been done to accommodate the additional changes, thus preventing an outage.

Platform (server hardware) resource utilization data is at the foundation of capacity analysis. The health monitoring performance counters discussed in the prior section are used to determine the capacity utilization of the server. This section will describe the process recommended and the reasons for doing routine capacity analysis and planning.

Capacity Planning Requires the Following Action Steps:

1. Collect Data
   - Initiate data sampling
   - Collect samples after a defined monitoring period
2. Categorize Data
   The collected data is distributed into three buckets which equate to three different levels:
   1. Hardware Level: resources on a single server
   2. Component Level: resources associated with a single application or a single application component (for example, ICM/CC Router) on a multi-application or multi-component server
   3. Solution Level: collective utilization level across the entire solution
3. Analyze Data for Target Categories

Use the methods and calculations provided in section 9.4 - Calculating Capacity Utilization to determine utilization levels for each category.

Once the data is collected, categorized and analyzed, it can then be related to:

| | | |
|---|---|---|
| 1. | Today's utilization: | A baseline - where am I at today? |
| 2. | Recent changes: | What effect did the recent change have compared to the baseline? |
| 3. | Tomorrow's plans: | "What If?" Scenarios: If I add 200 agents, what will likely be the effect? |

## 9.1 Capacity Planning Process

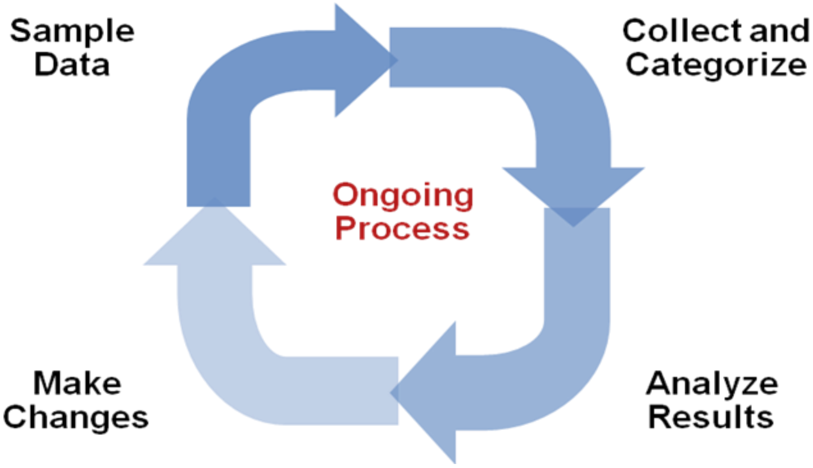


Figure 30: Capacity Planning Process

Changes to an existing Unified ICM/CC deployment should be made in small steps and then one should analyze the impact of each step with each iteration of a well-established, repeatable process. This process includes the following phases (steps):

1. **Sample Phase**
   - Initiate data sampling at the same time for the same interval for each change made
2. **Collect and Categorize Phase**
   - Collect the samples and distribute to appropriate buckets
3. **Analysis Phase**
   - Check application resource boundaries – has any component exceeded utilization limits?
   - Determine best fit for new deployment requirements
   - Estimate solution level capacity utilization for new requirements
4. **Change Phase**
   - Implement changes to solution based on analysis and estimate of impact
5. **Do it all over again**
   - Re-execute the process exactly the same it was done prior to ensure that an apples-to-apples comparison is made.

## 9.2 Capacity Planning – Getting Started

The first thing one must do to get started with a capacity management plan is to establish a baseline – answer the question: “what is my capacity utilization today?”. In order to answer this question, you must first determine the busiest, recurring period within a reasonable timeframe. For most business call centers, there is usually a 1-hour period of each day that is typically the busiest. Moreover, there will likely be busier days of the week (for example, Monday vs. Wednesday); busier days of the month (for example, the last business day of the month) or busier weeks of the year (for example, for insurance companies, the first week in January or for the IRS, the first two weeks of April). These traditionally busy hours, days or weeks represent the most taxing period on the deployment; these are the periods during which a capacity utilization calculation is best because you always want to ensure that your deployment is capable of handling the worst.

The steps to getting started are:

1. **Set up basic sampling – daily.**
    - Sample the performance counter values: CPU, Memory, Disk, Network, Call and Agent Traffic
2. **Determine the busy period**
    - Identify the RECURRING busy period – worst case scenario – by:
        - Per Component
        - Solution Wide
3. **Establish a baseline of utilization for the target period**
    - Determine hardware capacity utilization
    - Identify components with high capacity utilization
4. **Craft a recurring collection plan**
    - Devise a plan that is repeatable – preferably automated – that can be done on a weekly basis whereby samples are obtained during the busiest hour of the week.

Once a baseline is established and a busy hour identified, daily sampling is no longer necessary; sampling need only be done during the busy hour on a weekly basis. However, if regular reporting shows that the busy hour may have changed, then daily sampling must be done again so that the new busy hour can be identified. Once identified, weekly sampling during the busy hour can resume.

### Finding the “Busy” Hour

To find the busy hour, continuous data sampling must be initiated to cover a full week, 24x7. The data sampled are the performance counters for CPU, Memory, Disk and Network as listed in section 9.4 - Calculating Capacity Utilization. Performance counter values can be set up to be written to a disk file in comma separated values (.CSV) format which is easily imported into a Microsoft Excel workbook. Collect the data sample files, import them into Excel and graph them so as to see the busy hour. The data set can be graphed in a matter of minutes and the busy hour can be easily determined.

For example:

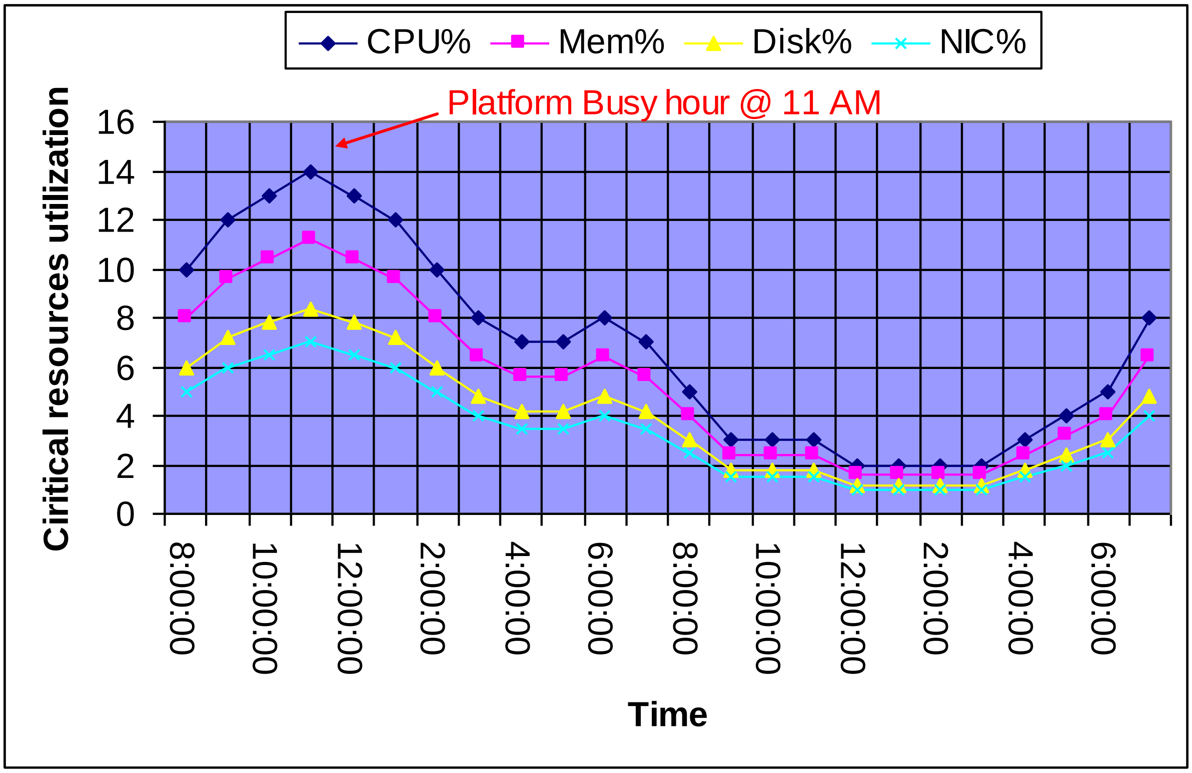


**Figure 31: Graph of Samples to Find Busy Hour**

## 9.3 Categorizing Collected Data

Collected data should be categorized by critical resource for each change event or need. Highlighted below are the instigators for sampling, collecting, categorizing, analyzing data to determine capacity utilization.

- **Current Deployment Design**
- **Configuration Info**
- **Traffic Load**
- **Migration Requirements**
- **Platform Performance**

### 9.3.1 Current Deployment Design

It is imperative that a deployment baseline be established and maintained; this baseline will be used to do before/after comparisons. At any time that a change in the deployment design is made, a new baseline must be established.

- Establish an initial baseline – today – with the current deployment design
- Re-establish a baseline after deployment changes occur, such as:
  - Add/delete a Peripheral Gateway
  - Add/delete an AW
  - Clustering over WAN – any change to WAN characteristics

It is also important to note that week-to-week comparisons can be used to identify changes that occurred that you were not aware of. For example: someone added additional skill groups without prior approval/notification – suddenly utilization jumped, inexplicably, by 5%. Such a change is noteworthy enough to ask questions: What changed? When? And why?

When analyzing the current solution, one must maintain deployment information and track changes

- Topology Diagrams (Network)
- Peripheral Counts
  - Cisco Unified Communications Manager Clusters
  - IP-IVR or CVP Peripherals (and port quantity)
- Network Devices
- Third-Party Add Ons

### 9.3.2 Configuration Information

Changes to Unified ICM/CC configuration can have an impact on computing resources and thus an impact on utilization for a hardware platform, an application component and in some cases, an impact on the entire solution.

- Configuration change examples:
  - Adding skill groups
  - Changing number of skill groups per agent
  - Adding ECC data
  - Increasing calls offered (per peripheral) per ½ hour

Using the baseline that you've established, you can now easily characterize the impact of the configuration change by comparing utilization before the change to utilization after change.

Secondarily, by making changes methodically in small steps, you can characterize each small change (for example, adding one skill group at a time) and note the impact. In the future, if a change request comes to add 10 skills group, you can make an educated guess at the overall utilization impact by extrapolating: adding one skill group caused a 0.5% increase in PG CPU utilization at the ½ hour, so adding 10 skill groups will probably result in a 5% increase in PG CPU utilization at the ½ hour. Thus begs the question: Can a 5% increase in PG CPU utilization be accommodated?

Configuration changes often have an impact on performance; track ongoing changes and analyze the impact**.** The following configuration changes are likely to impact utilization:

- Overall Database Size
- Number of Skill Groups per Agent
- Number of Skill Groups per Peripheral
- Number of Call Types
- Number of Dialed Numbers
- Number of Agents per Peripheral
- Total Agent Count
- Amount of Attached Call Data

Other configuration factors that can affect utilization:

- Agent level reporting
- Persistent ECC, per call type, per peripheral
- Percentage of call types per peripheral
- Average skill group per agents and total skills per system
- Number of AWs (real time feeds)
- Number of concurrent reporting users

### 9.3.3 Traffic Load

Examples of impacting traffic load changes:

- Inbound call rate

  For example, your marketing department is about to introduce a new discount program for an existing service: "Sign up before July 31 for the new discounted rate!" You've been religiously monitoring inbound call rate (ICM/CC Router: "Calls/sec" counter) and see a pretty consistent 4 calls/sec inbound rate during the Monday morning busy hour as compared to an average of 3 calls/sec during the rest of the day. You predict that the new marketing program will increase the inbound call rate to 6 calls/sec during the busy hour. You've calculated that utilization is at 50% during the busy hour while averaging at 40% during the rest of the day. You determine that the increase in call rate will push utilization as high as 75%, which the system can tolerate.
- Network utilization

  The Unified ICM/CC system is a collection of distributed, dependent software components that communicate by network messaging. Components communicate via a public network connection – some components also communicate via a private, dedicated network connection. On the public network, Unified ICM/CC may be competing for network bandwidth. Any increase in public network utilization may slow a Unified ICM/CC component's ability to transmit data on the network, causing output queues to grow more than normal. This may have an impact on memory utilization on the server not to mention the possible effects on timing of real-time operations.

Any change in traffic or load will have a corresponding impact on utilization and capacity**.** Additional examples of impacting traffic include:

- Overall Call Load—BHCA and Calls per Second
- Persistent ECC, per call type, per peripheral
- Percentage of call types per peripheral
- Number of concurrent agents logged in (including monitored IVR ports)
- Number of concurrent reporting users

### 9.3.4 Migration Requirements

When analyzing future growth, one must consider all possible migrations**:**

- Business Requirements for Migration
  - Adding a new line of business, additional skill groups
- Expected Growth
  - Recent history has shown a steady 10% increase in agent population
- Resource Consolidations of Separations

- Agents
- Call Types
- Reporting
- Queuing
- Merging two peripherals into one

- Other Requirements
  - Office moving to new location
  - Network infrastructure change: increased/decrease network latency.
  - Splitting PG sides over WAN
  - Changing data retention parameters on the HDS

### 9.3.5 Platform Performance

Any changes in the platform itself will likely have a corresponding impact on utilization. For example:

- Hardware upgrades
- Software upgrades

A “technology refresh” upgrade (upgrading both hardware and software) of Unified ICM/CC will have a significant effect on capacity utilization. Advances in hardware capabilities and a continued focus on streamlining bottlenecks in the software have yielded significant increases in server and component capacities.

In some cases, hardware upgrades (without a software upgrade) may be necessary to accommodate growth in the Unified ICM/CC deployment. If

A “common ground” upgrade (upgrading software while retaining existing hardware) of Unified ICM/CC may have a differing effect on capacity utilization depending on the changes made to the software from one release to the next. In some components, utilization may increase slightly because new functionality has been added to the component which has slightly decreased its execution performance. However, another component in which performance improvements have been introduced, utilization may decrease from one release to the next.

It is important to plan to re-establish a capacity utilization baseline after any upgrade.

## 9.4 Calculating Capacity Utilization

Platform resource utilization data is at the foundation of capacity analysis. This data is sampled values of performance counters such as: CPU, Memory, Disk and Network. The data set is from the busy hour as determined by the steps described above.

The recommended sample rate is one sample every 15 seconds of each of the listed counters. Of the sample set, we will base the calculation on the 95$^{th}$ percentile sample. The 95th percentile is the smallest number that is greater than 95% of the numbers in a given set. Using this value will eliminate short-duration spikes that are statistical outliers.

Counters are divided into two categories:

1. “Measurement” value

   A measurement value is only valid if the indicator value(s) is/are “good.” If the indicator value(s) is/are within acceptable levels, then the measurement value is used in the forthcoming calculation to determine utilization.

2. “Indicator” value

   An indicator value is a Boolean indication of “good” or “bad” – exceeding the maximum threshold is, of course, “bad”. If the indicator value is “bad”, assume that capacity utilization has been exceeded. If so, steps must be taken to return the system to < 100% utilization which may require hardware upgrade.

Capacity utilization is considered to be >= 100% if published sizing limits have been exceeded for any given component (as published in the *Hardware and System Software Specification* (AKA “Bill of Materials” or “BOM”) or the *Solution Reference Network Design* (SRND) document). For example: if the server on which a Unified CC PG is installed has a published capacity of 1,000 agents but there are 1,075 active agents at a particular time, the server is considered to be greater-than 100% utilization regardless of what might be calculated using the methods described herein. The reason for this is that although the server/application seems to be performing at acceptable levels, any legitimate change in usage patterns could drive utilization beyond 100% and cause a system outage because the published capacity has been exceeded. Published capacities seek to take into account differences between deployments and/or changes in usage patterns without driving the server into the red zones of performance thresholds. As such, all deployments must remain within these published capacities in order to enjoy continued Cisco support.

### 9.4.1 Calculating CPU Utilization

$$\overline{CPU_{\rho}}(t_n) = \frac{CPU_{95\%}(t_n)}{CPU_{Sat}} * 100$$

- $\text{CPU}_{95\%}$

  Measurement Counter: Processor – % Processor Time (_Total)

- $\text{CPU}_{\text{Sat}}$

  Maximum threshold: 60%

- Indicator

  Counter: System – Processor Queue Length
  Threshold: 2

### 9.4.2 Calculating Memory Utilization

$$Mem_{Sat} = Mem_{physical} * .8$$

$$\overline{Mem_{\rho}}(t_n) = \frac{Mem_{95\%}(t_n)}{Mem_{Sat}} * 100$$

- $\text{Mem}_{95\%}$

  Measurement Counter: Memory – Committed Bytes

- $\text{Mem}_{\text{Sat}}$

  Threshold: 80% (of physical memory)

- Indicator Counters

  Counter: Memory – Available Mbytes
  Threshold: < 20%

| | |
|---|---|
| Counter: | Memory – Pages / sec |
| Threshold: | 20% |
| Counter: | Paging File – % Usage |
| Threshold: | 80% |

### 9.4.3 Calculating Disk Utilization

$$\overline{Disk_{\rho}}(t_n) = \frac{DT_{95\%}(t_n)}{DT_{Sat}} * 100$$

- $DT_{95\%}$

  Measurement Counter: Physical Disk - % Disk Time (_Total)
- $DT_{Sat}$

  Maximum threshold: 50%
- Indicator

  | | |
  |---|---|
  | Counter: | Physical Disk – Avg. Disk Queue Length |
  | Threshold: | 1.5 |

### 9.4.4 Calculating NIC Utilization

$$NIC_{Sat} = NIC_{physical} * .03$$

$$\overline{NIC_{\rho}}(t_n) = \frac{NIC_{95\%}(t_n)}{NIC_{Sat}} * 100$$

- $NIC_{95\%}$

  Measurement Counter: Network Interface – Bytes Total / sec
- $NIC_{Sat}$

  Maximum threshold: 30%

  100 Mbps NIC: 3 MB / sec (approximately)
  1 Gbps NIC: 30 MB / sec (approximately)
- Indicator

  | | |
  |---|---|
  | Counter: | Network Interface – Output Queue Length |
  | Threshold: | 1 |

### 9.4.5 Calculating Maximum Utilization

The highest utilization can be determined with:

$$\overline{UTIL_{\rho}} = MAX\,(\overline{CPU_{\rho}}[t], \overline{Mem_{\rho}}[t], \overline{Disk_{\rho}}[t], \overline{NIC_{\rho}}[t])$$

### 9.4.6 Relating Traffic Load to Resources

Use Unified ICM/CC Router counters to relate traffic load to resource utilization. The Unified ICM/CC Router Performance Counters are:

- Calls/sec
- Calls In Progress

- Agents Logged On

Graphing these data sets relative to resource data sets may provide a compelling visual message.

# 10 ICM/CC Diagnostic Tools

## 10.1 DUMPLOG

### Using the DUMPLOG Utility Optional Cisco Log Message Format

The DUMPLOG utility converts binary log files written by Unified ICM/CC processes into readable text format. An enhancement has been added to DUMPLOG with release 7.2(1) of Unified ICM/CC to optionally display the binary log files in Cisco Log message format. See section 5.1 for details about the Cisco Log format. Refer to the *How to Use the DumpLog Utility* Tech Note located at:

**http://www.cisco.com/en/US/products/sw/custcosw/ps1001/prod_tech_notes_list.html**

for additional information about this utility.

### Header

Cisco Log formatted log entries include a more comprehensive header compared to DUMPLOG standard format.

### DumpLog Standard Format

Standard formatted DUMPLOG entries display the following fields:

```
<TIMESTAMP> <COMPONENT-PROCESS> <MESSAGE>
```

The timestamp is represented as a 24-hour value (hh:mm:ss). It does not include the date, which is displayed on a separate line at the beginning of the file and when a new day starts.  For example:

```
Events from February 8, 2007
00:37:44 ra-rtr MDS is in service.
```

### Cisco Log Format

Cisco Log formatted DUMPLOG entries display the following fields:

```
<SEQNUM>: <HOST>: <TIMESTAMP> <TIMEZONE>: %APPNAME: %<TAGS>:<MESSAGE>
```

Below is an example of a Cisco Log formatted DUMPLOG message. An actual log entry is displayed on a single line.

```
10: CICMRGRA: Feb 8 2007 05:37:44.658 +0000: %ICM_Router_ProcessSynchronization:
[comp=Router-A][pname=rtr][iid=ipcc][sev=info]: MDS is in service.
```

**Note**: The contents of the APPNAME and TAGS fields differ from those previously described in section 5.1.

**Table 10-1: APPNAME and TAGS Used in DUMPLOG Trace Output**

| Field | Description |
|---|---|
| APPNAME | PRODUCT_COMPONENT_MESSAGECATEGORY<br>PRODUCT - always ICM<br>COMPONENT – such as Router |

| | |
|---|---|
| | MESSAGECATEGORY – such as ProcessSynchronization |
| TAGS | Acceptable tags are:<br>[comp=%s] - component name including side, such as Router A<br>[pname=%s] - process name, such as rtr<br>[iid=%s] - instance name, such as ipcc<br>[sev=%s] – severity, such as info<br>and optionally [part=%1.%2/%3], which is used only for multi-line entries as described later in this section. |

## Timestamp

The timestamp displayed in DUMPLOG standard format is in local time relative to the server on which DUMPLOG is run. The timestamp displayed in Cisco Log format is in GMT time independent of the server on which DUMPLOG is run.

**Note:** Date/time options specified on the command line are entered in local time, regardless of whether the Cisco Log option is selected. Therefore, timestamps displayed as part of the Cisco Log formatted entry might appear to be outside of the date/time range selected.

## Multi-line Entries

The message portion of some DUMPLOG entries might contain one or more embedded new line characters ('\n'), which cause the messages to display on multiple lines and might also include blank lines. This is especially true for entries that contain statistics.

For a DUMPLOG standard formatted message, only the first line will contain the header field as shown in the following example:

```
00:36:09 ra-nm ICM\ipcc\RouterA node reporting process statistics for process ccag.
   Process name: ccag
   Process status: A
   Process ID: 6c0
   Number of times process started: 1
   Last start time: 00:35:31 2/8/2007
   Pings completed in zero time: 0
   Pings completed in first third: 0
   Total first third milliseconds: 0
   Pings completed in second third: 0
   Total second third milliseconds: 0
   Pings completed in third third: 0
   Total third third milliseconds: 0
   Longest Ping time: 0
```

For a Cisco Log formatted message, each line will contain a separate header as shown in the following example.

```
19: CICMRGRA: Feb 8 2007 05:36:09.890 +0000: %ICM_Router_unknown: [comp=Router-
A][pname=nm][iid=ipcc][sev=info][part=19.1/14]: ICM\ipcc\RouterA node reporting
process statistics for process ccag.

20: CICMRGRA: Feb 8 2007 05:36:09.890 +0000: %ICM_Router_unknown: [comp=Router-
A][pname=nm][iid=ipcc][sev=info][part=19.2/14]: Process name: ccag

21: CICMRGRA: Feb 8 2007 05:36:09.890 +0000: %ICM_Router_unknown: [comp=Router-
A][pname=nm][iid=ipcc][sev=info][part=19.3/14]: Process status ACTIVE
```

```
22: CICMRGRA: Feb 8 2007 05:36:09.890 +0000: %ICM_Router_unknown: [comp=Router-
A][pname=nm][iid=ipcc][sev=info][part=19.4/14]: Process ID 6c0

23: CICMRGRA: Feb 8 2007 05:36:09.890 +0000: %ICM_Router_unknown: [comp=Router-
A][pname=nm][iid=ipcc][sev=info][part=19.5/14]: Number of times process started 1

24: CICMRGRA: Feb 8 2007 05:36:09.890 +0000: %ICM_Router_unknown: [comp=Router-
A][pname=nm][iid=ipcc][sev=info][part=19.6/14]: Last start time: 00:35:31 2/8/2007

25: CICMRGRA: Feb 8 2007 05:36:09.890 +0000: %ICM_Router_unknown: [comp=Router-
A][pname=nm][iid=ipcc][sev=info][part=19.7/14]: Pings completed in zero time: 0

26: CICMRGRA: Feb 8 2007 05:36:09.890 +0000: %ICM_Router_unknown: [comp=Router-
A][pname=nm][iid=ipcc][sev=info][part=19.8/14]: Pings completed in first third: 0

27: CICMRGRA: Feb 8 2007 05:36:09.890 +0000: %ICM_Router_unknown: [comp=Router-
A][pname=nm][iid=ipcc][sev=info][part=19.9/14]: Total first third milliseconds: 0

28: CICMRGRA: Feb 8 2007 05:36:09.890 +0000: %ICM_Router_unknown: [comp=Router-
A][pname=nm][iid=ipcc][sev=info][part=19.10/14]: Pings completed in second third: 0

29: CICMRGRA: Feb 8 2007 05:36:09.890 +0000: %ICM_Router_unknown: [comp=Router-
A][pname=nm][iid=ipcc][sev=info][part=19.11/14]: Total second third milliseconds: 0

30: CICMRGRA: Feb 8 2007 05:36:09.890 +0000: %ICM_Router_unknown: [comp=Router-
A][pname=nm][iid=ipcc][sev=info][part=19.12/14]: Pings completed in third third: 0

31: CICMRGRA: Feb 8 2007 05:36:09.890 +0000: %ICM_Router_unknown: [comp=Router-
A][pname=nm][iid=ipcc][sev=info][part=19.13/14]: Total third third milliseconds: 0

32: CICMRGRA: Feb 8 2007 05:36:09.890 +0000: %ICM_Router_unknown: [comp=Router-
A][pname=nm][iid=ipcc][sev=info][part=19.14/14]: Longest Ping Time: 0
```

To differentiate each line in the entry, the part tag is added to each header where:

```
[part=#1.#2/#3]
```

`#1` = the sequence number of the first line (this is the same for all lines in the entry)
`#2` = the part number of the specific line
`#3` = the total number of parts in the entry

Note the line beginning with sequence number 32, [part=19.14/14]:

```
#1 = 19. #2 = 14 / #3 = 14
```

# 11 Appendix A

## Cisco Contact Center Applications MIB Results Example

The following example displays the data provided by the Cisco Contact Center Applications MIB SNMP agent on the target Unified ICM/CC installation icm70 in response to a series of SNMP GETNEXT requests beginning at node ciscoCcaMIB, OID 1.3.6.1.4.1.9.9.473.

For the purpose of example, assume that a single instance:

```
cccaInstanceName.2 = acme
```

has been installed with instance number 0 and the following components are installed:

Router:

```
cccaComponentName.instanceNumber(0).componentIndex(1) = RouterA
```

Logger:

```
cccaComponentName.instanceNumber(0).componentIndex(2) = LoggerA
```

Peripheral Gateway:

```
cccaComponentName.instanceNumber(0).componentIndex(3) = PG1A
```

Distributor Admin Workstation:

```
cccaComponentName.instanceNumber(0).componentIndex(4) = Distributor
```

A single CRSP NIC has been installed as part RouterA:

```
cccaNicType.instanceNumber(0).componentIndex(1).nicIndex(1) = crsp
```

A single Unified Contact Center Express PIM (acmiCRS) has been installed as part of PG1A:

```
cccaPimPeripheralName.instanceNumber(0).componentIndex(3).cccaPimNumber(1) = ACD 1
```

```
cccaName.0 = cc-rgr1a
cccaDescription.0 = Cisco Intelligent Contact Management / IP Contact Center
cccaVersion.0 = 7.1(1)
cccaTimeZoneName.0 = Eastern Standard Time
cccaTimeZoneOffsetHours.0 = 5
cccaTimeZoneOffsetMinutes.0 = 0
cccaSupportToolsURL.0 =
cccaInstanceName.0 = acme
cccaComponentType.0.1 = router(1)
cccaComponentType.0.2 = logger(2)
cccaComponentType.0.3 = pg(4)
cccaComponentType.0.4 = distAW(3)
cccaComponentName.0.1 = RouterA
cccaComponentName.0.2 = LoggerA
cccaComponentName.0.3 = PG1A
cccaComponentName.0.4 = Distributor
cccaComponentStatus.0.1 = started(4)
cccaComponentStatus.0.2 = started(4)
cccaComponentStatus.0.3 = started(4)
cccaComponentStatus.0.4 = started(4)
cccaComponentElmtName.0.1.1 = ccagent
cccaComponentElmtName.0.1.2 = crspnic
cccaComponentElmtName.0.1.3 = dbagent
cccaComponentElmtName.0.1.4 = mdsproc
cccaComponentElmtName.0.1.5 = router
cccaComponentElmtName.0.1.6 = rtsvr
cccaComponentElmtName.0.1.7 = testsync
cccaComponentElmtName.0.2.8 = configlogger
cccaComponentElmtName.0.2.9 = csfs
cccaComponentElmtName.0.2.10 = histlogger
cccaComponentElmtName.0.2.11 = recovery
cccaComponentElmtName.0.3.12 = mdsproc
cccaComponentElmtName.0.3.13 = opc
cccaComponentElmtName.0.3.14 = pgagent
cccaComponentElmtName.0.3.15 = acmipim
cccaComponentElmtName.0.3.16 = testsync
cccaComponentElmtName.0.4.17 = configlogger
cccaComponentElmtName.0.4.18 = rtclient
cccaComponentElmtName.0.4.19 = rtdist
cccaComponentElmtName.0.4.20 = updateaw
cccaComponentElmtRunID.0.1.1 = 3336
cccaComponentElmtRunID.0.1.2 = 2992
cccaComponentElmtRunID.0.1.3 = 3600
cccaComponentElmtRunID.0.1.4 = 3920
cccaComponentElmtRunID.0.1.5 = 4040
cccaComponentElmtRunID.0.1.6 = 3532
cccaComponentElmtRunID.0.1.7 = 4100
cccaComponentElmtRunID.0.2.8 = 948
cccaComponentElmtRunID.0.2.9 = 3248
cccaComponentElmtRunID.0.2.10 = 1248
cccaComponentElmtRunID.0.2.11 = 3272
cccaComponentElmtRunID.0.3.12 = 4724
cccaComponentElmtRunID.0.3.13 = 4864
cccaComponentElmtRunID.0.3.14 = 4964
cccaComponentElmtRunID.0.3.15 = 5236
cccaComponentElmtRunID.0.3.16 = 5228
cccaComponentElmtRunID.0.4.17 = 5460
cccaComponentElmtRunID.0.4.18 = 5488
cccaComponentElmtRunID.0.4.19 = 5504
cccaComponentElmtRunID.0.4.20 = 5536
cccaComponentElmtStatus.0.1.1 = active(5)
cccaComponentElmtStatus.0.1.2 = started(4)
cccaComponentElmtStatus.0.1.3 = active(5)
cccaComponentElmtStatus.0.1.4 = active(5)
cccaComponentElmtStatus.0.1.5 = active(5)
cccaComponentElmtStatus.0.1.6 = active(5)
cccaComponentElmtStatus.0.1.7 = active(5)
cccaComponentElmtStatus.0.2.8 = active(5)
cccaComponentElmtStatus.0.2.9 = active(5)
cccaComponentElmtStatus.0.2.10 = active(5)
cccaComponentElmtStatus.0.2.11 = active(5)
cccaComponentElmtStatus.0.3.12 = active(5)
cccaComponentElmtStatus.0.3.13 = active(5)
cccaComponentElmtStatus.0.3.14 = active(5)
cccaComponentElmtStatus.0.3.15 = standby(6)
cccaComponentElmtStatus.0.3.16 = active(5)
cccaComponentElmtStatus.0.4.17 = active(5)
cccaComponentElmtStatus.0.4.18 = active(5)
cccaComponentElmtStatus.0.4.19 = active(5)
cccaComponentElmtStatus.0.4.20 = active(5)
cccaRouterSide.0.1 = sideA(1)
cccaRouterCallsPerSec.0.1 = 0
cccaRouterAgentsLoggedOn.0.1 = 0
cccaRouterCallsInProgress.0.1 = 0
cccaRouterDuplexPairName.0.1 = cc-rgr1a
cccaRouterNicCount.0.1 = 1
cccaNicType.0.1.1 = crsp(5)
cccaNicStatus.0.1.1 = started(4)
cccaLoggerSide.0.2 = sideA(1)
```

```
cccaLoggerType.0.2 = standard(1)
cccaLoggerRouterSideAName.0.2 = cc-rgr1a
cccaLoggerRouterSideBName.0.2 = cc-rgr1a
cccaLoggerDuplexPairName.0.2 = cc-rgr1a
cccaLoggerHDSReplication.0.2 = 0
cccaDistAwSide.0.4 = sideA(1)
cccaDistAwType.0.4 = standard(0)
cccaDistAwAdminSiteName.0.4 = cc-rgr1a
cccaDistAwRouterSideAName.0.4 = cc-rgr1a
cccaDistAwRouterSideBName.0.4 = cc-rgr1a
cccaDistAwLoggerSideAName.0.4 = cc-rgr1a
cccaDistAwLoggerSideBName.0.4 = cc-rgr1a
cccaDistAwDuplexPairName.0.4 = cc-rgr1a
cccaDistAwHDSEnabled.0.4 = 0
cccaDistAwWebViewEnabled.0.4 = false(2)
cccaDistAwWebViewServerName.0.4 =
cccaPgNumber.0.3 = 1
cccaPgSide.0.3 = sideA(1)
cccaPgRouterSideAName.0.3 = cc-rgr1a
cccaPgRouterSideBName.0.3 = cc-rgr1a
cccaPgDuplexPairName.0.3 = cc-rgr1a
cccaPgPimCount.0.3 = 1
cccaPimPeripheralName.0.3.1 = ACD 1
cccaPimPeripheralType.0.3.1 = acmiCRS(19)
cccaPimStatus.0.3.1 = started(4)
cccaPimPeripheralHostName.0.3.1 = LabHost
```

# 12 Appendix B – Unified CCE SNMP Notifications

Notes:

1. The message ID also contains the severity in the two most significant bits of the integer value. The message ID value shown is with these two bits masked to zero.
2. Alarms with an asterisk next to the Message ID are deemed to be "***critical***" alarms.
3. The "%n" label (where 'n' is a numeric value) indicates a substitution field whereby node-specific or process-specific information is inserted.

**Table 12-1: SNMP Notifications**

| MsgID (hex) | Type | Severity | Message Class | MessageText |
|---|---|---|---|---|
| **Description** | | | | **Action** |
| 1028001 | Clear | Warning | NM INITIALIZING | Node Manager initializing. |
| The node management library, common to nearly all ICM processes, is initializing itself. This is standard practice when a process (re)starts. | | | | No action is required. |
| 1028003 | Clear | Informational | NM INITIALIZING | Node Manager started.  Last shutdown was by operator request. |
| The Node Manager successfully started. The last reason the Node Manager stopped was because a clean shutdown of the ICM code was requested by the operator. | | | | No action is required. |
| 1028004 | Clear | Informational | NM INITIALIZING | Node Manager started.  Last shutdown was due to system shutdown. |
| The Node Manager successfully started. The last reason the Node Manager stopped was because a clean shutdown of the node was requested by the operator. | | | | No action is required. |
| 1028005 | Raise | Warning | NM INITIALIZING | Operator initiated node shutdown. |
| The operator/administrator has requested that the ICM software be shutdown. | | | | No action is required. |
| 1028101 | Clear | Warning | NM INITIALIZING | %1 Node Manager initializing. |
| The node management library, common to nearly all ICM processes, is initializing itself. This is standard practice when a process (re)starts. | | | | No action is required. |
| 1028103 | Clear | Informational | NM INITIALIZING | %1 Node Manager started.  Last shutdown was by operator request. |
| The Node Manager successfully started. The last reason the Node Manager stopped was because a clean shutdown of the ICM code was requested by the operator. | | | | No action is required. |
| 1028104 | Clear | Informational | NM INITIALIZING | %1 Node Manager started.  Last shutdown was due to system shutdown. |
| The Node Manager successfully started. The last reason the Node Manager stopped was because a clean shutdown of the node was requested by the operator. | | | | No action is required. |
| 1028105 | Raise | Warning | NM INITIALIZING | The operator/administrator has shutdown the ICM software on %1. |

| | | | | |
|---|---|---|---|---|
| | Node Manager on the ICM node has been given the command to stop ICM services. This occurs when an operator/administrator stops ICM services using ICM Service Control, 'nmstop', 'net stop', Control Panel Services, or shuts down the node. | | | Contact the operator/administrator to determine the reason for the shutdown. |
| 1029001 | Clear | Informational | NM INITIALIZING | Node Manager Manager started. |
| | The Node Manager Manager process (which oversees the Node Manager process) has started. | | | No action is required. |
| 1029101 | Clear | Informational | NM INITIALIZING | %1 Node Manager Manager started. |
| | The Node Manager Manager process (which oversees the Node Manager process) has started. | | | No action is required. |
| 102C001* | Raise | Error | NM REBOOT ON FAIL | Critical process %1 died. Rebooting node. |
| | A critical process needed to run the ICM software on this node has died. The Node Manager is forcing a reboot of the node. | | | Contact the Support Center. |
| 102C003* | Clear | Warning | NM REBOOT ON FAIL | Restarting process %1. |
| | The Node Manager is restarting process %1 after the process died or was terminated. | | | No action is required. |
| 102C007 | Clear | Informational | NM INITIALIZING | Node Manager started. Last shutdown was for reboot after failure of critical process. |
| | The Node Manager has started. The last shutdown was requested by the Node Manager since it recognized that a critical process for the node failed. | | | No action is required. |
| 102C008 | Clear | Error | NM INITIALIZING | Node Manager started. Last shutdown was for unknown reasons. Possible causes include a power failure, a system crash or a Node Manager crash. |
| | The Node Manager has started. The Node Manager cannot determine why the system is restarting. Possible causes are power failure, a system crash (Windows NT blue screen), a system hang (in which an operator forced a reboot), or the Node Manager itself crashed. | | | Contact the Support Center. |
| 102C009* | Raise | Warning | NM REBOOT ON FAIL | Process %4 exited after %1 seconds. Minimum required uptime for %4 process is %2 seconds. Delaying process restart for %3 seconds. |
| | Process %4 exited after running for %1 seconds. Such processes must run for at least %2 seconds before the Node Manager will automatically restart them after they terminate. The Node Manager will restart the process after delaying %3 seconds for other environmental changes to complete. | | | No action is required. |
| 102C00A* | Clear | Warning | NM REBOOT ON FAIL | Restarting process %2 after having delayed restart for %1 seconds. |
| | The Node Manager is restarting process %2 after the requisite delay of %1 seconds. | | | No action is required. |
| 102C00B* | Raise | Error | NM REBOOT ON FAIL | Terminating process %1. |
| | The Node Manager is terminating process %1. | | | No action is required. |
| 102C00C* | Raise | Error | NM REBOOT ON FAIL | Process %1 exited after having detected a software failure. |
| | Process %1 exited (terminated itself) after it detected an | | | If the process continues to terminate itself, call the |

| | | | | |
|---|---|---|---|---|
| | internal software error. | | | Support Center. |
| 102C00D* | Raise | Warning | NM REBOOT ON FAIL | Process %1 detected failure and requested that it be restarted by the Node Manager. |
| | Process %1 has detected a situation that requires it to request that the Node Manager restart it. This often indicates a problem external to the process itself (for example, some other process may have failed). | | | If the process continues to terminate itself, call the Support Center. |
| 102C00E* | Raise | Error | NM REBOOT ON FAIL | Process %1 exited with unexpected exit code %2. |
| | Process %1 exited (terminated) with exit code %2. This termination is unexpected and the process died for an unknown reason. | | | Contact the Support Center. |
| 102C00F* | Raise | Warning | NM REBOOT ON FAIL | Process %3 exited after %1 seconds. Process restart will be delayed for a minimum of %2 seconds. |
| | Process %3 exited after running for %1 seconds. The Node Manager will restart the process after delaying %2 seconds for other environmental changes to complete. | | | If the process continues to terminate itself, call the Support Center. |
| 102C010* | Clear | Warning | NM REBOOT ON FAIL | Process %1 successfully reinitialized after restart. |
| | Process %1 was successfully restarted. | | | No action is required. |
| 102C011* | Clear | Informational | NM REBOOT ON FAIL | Process %1 successfully started. |
| | Process %1 was successfully started. | | | No action is required. |
| 102C012* | Raise | Warning | NM REBOOT ON FAIL | Process %1 exited cleanly and requested that it be restarted by the Node Manager. |
| | Process %1 terminated itself successfully and has requested that the Node Manager restart it. | | | No action is required. |
| 102C013 | Raise | Warning | NM REBOOT ON FAIL | Process %1 exited from Control-C or window close. |
| | Process %1 exited as a result of a CTRL-C request or a request to close the process's active window. | | | No action is required. |
| 102C014* | Raise | Error | NM INITIALIZING | Process %1 exited and requested that the Node Manager reboot the system. |
| | Process %1 terminated itself successfully but, due to other conditions, has requested that the Node Manager reboot the machine. | | | No action is required. |
| 102C101* | Raise | Error | NM REBOOT ON FAIL | %1 node critical process %2 died. Rebooting node. |
| | A critical process needed to run the ICM software on this node has died. The Node Manager is forcing a reboot of the node. | | | Contact the Support Center. |
| 102C103* | Clear | Warning | NM REBOOT ON FAIL | %1 node restarting process %2. |
| | The Node Manager is restarting process %2 after the process died or was terminated. | | | No action is required. |
| 102C107* | Clear | Informational | NM INITIALIZING | %1 Node Manager started. Last shutdown was for reboot after failure of critical process. |
| | The Node Manager has started. The last shutdown was requested by the Node Manager since it recognized that a critical process for the node failed. | | | No action is required. |

| | | | | |
|---|---|---|---|---|
| 102C108* | Clear | Error | NM INITIALIZING | %1 Node Manager started. Last shutdown was for unknown reasons. Possible causes include a power failure, a system crash or a Node Manager crash. |
| | | | | |
| 102C109* | Raise | Warning | NM REBOOT ON FAIL | %4 node process %5 exited after %1 seconds. Minimum required uptime for %5 process is %2 seconds. Delaying process restart for %3 seconds. |
| | Process %5 exited after running for %1 seconds. Such processes must run for at least %2 seconds before the Node Manager will automatically restart them after they terminate. The Node Manager will restart the process after delaying %3 seconds for other environmental changes to complete. | | | No action is required. |
| 102C10A* | Clear | Warning | NM REBOOT ON FAIL | %2 node restarting process %3 after having delayed restart for %1 seconds. |
| | The Node Manager is restarting process %3 after the requisite delay of %1 seconds. | | | No action is required. |
| 102C10B* | Raise | Error | NM REBOOT ON FAIL | Terminating process %2. |
| | The %1 Node Manager is terminating process %2. | | | No action is required. |
| 102C10C* | Raise | Error | NM REBOOT ON FAIL | %1 node process %2 exited after having detected a software failure. |
| | Process %2 exited (terminated itself) after it detected an internal software error. | | | If the process continues to terminate itself, call the Support Center. |
| 102C10D* | Raise | Warning | NM REBOOT ON FAIL | Process %2 on %1 has detected a failure. Node Manager is restarting the process. |
| | The specified Process has detected a situation that requires it to request that the Node Manager restart it. This often indicates a problem external to the process itself (for example, some other process may have failed). | | | Node Manager on the ICM node will restart the process. The node should be checked to assure it is online using rttest. If the condition is common, the process logs must be examined for cause. |
| 102C10E* | Raise | Error | NM REBOOT ON FAIL | Process %2 on %1 went down for unknown reason. Exit code %3. It will be automatically restarted. |
| | The specified Process exited (terminated) with the indicated exit code. This termination is unexpected and the process died for an unknown reason. It will be automatically restarted. | | | Contact the Support Center. |
| 102C10F* | Raise | Warning | NM REBOOT ON FAIL | Process %4 on %3 is down after running for %1 seconds. It will restart after delaying %2 seconds for related operations to complete. |
| | Specified process is down after running for the indicated number of seconds. It will restart after delaying for the specified number of seconds for related operations to complete. | | | Determine if process has returned to service or has stayed offline. If process is offline or bouncing determine the cause from logs. |
| 102C110* | Clear | Warning | NM REBOOT ON FAIL | %1 node process %2 successfully reinitialized after restart. |
| | Process %2 was successfully restarted. | | | No action is required. |
| 102C111* | Clear | Informational | NM REBOOT ON FAIL | %1 node process %2 successfully started. |
| | Process %2 was successfully started. | | | No action is required. |
| 102C112* | Raise | Warning | NM REBOOT ON FAIL | %1 node process %2 exited cleanly and requested that it be restarted by the Node Manager. |

| | | | | |
|---|---|---|---|---|
| | Process %2 terminated itself successfully and has requested that the Node Manager restart it. | | | No action is required. |
| 102C113 | Raise | Warning | NM REBOOT ON FAIL | %1 node process %2 exited from Control-C or window close. |
| | Process %2 exited as a result of a CTRL-C request or a request to close the process's active window. | | | No action is required. |
| 102C114* | Raise | Error | NM INITIALIZING | %1 node process %2 exited and requested that the Node Manager reboot the system. |
| | Process %2 terminated itself successfully but, due to other conditions, has requested that the Node Manager reboot the machine. | | | No action is required. |
| 102D001* | Raise | Error | NM INITIALIZING | Node Manager crashed after having been up for %1 seconds. Scheduling system reboot in %2 seconds. |
| | The Node Manager has itself crashed after having run for %1 seconds. The machine will be rebooted after waiting %2 seconds. | | | Contact the Support Center. |
| 102D002* | Raise | Error | NM INITIALIZING | Node Manager crashed after having been up for %1 seconds. Auto-reboot is disabled. Will attempt service restart. |
| | The Node Manager has itself crashed after having run for %1 seconds. The machine cannot be rebooted since auto-reboot is disabled. The Node Manager Manager will attempt to restart the service. | | | Contact the Support Center. |
| 102D003* | Raise | Error | NM INITIALIZING | Node Manager requested reboot after having been up for %1 seconds. Scheduling system reboot in %2 seconds. |
| | The Node Manager has requested the machine be rebooted after having run for %1 seconds. The machine will be rebooted after waiting %2 seconds. | | | Contact the Support Center. |
| 102D004* | Raise | Error | NM INITIALIZING | Node Manager requested reboot after having been up for %1 seconds. Auto-reboot is disabled. Will attempt service restart. |
| | The Node Manager has requested the machine be rebooted after having run for %1 seconds. The machine cannot be rebooted since auto-reboot is disabled. The Node Manager Manager will attempt to restart the service. | | | Contact the Support Center. |
| 102D101* | Raise | Error | NM INITIALIZING | %3 Node Manager crashed after having been up for %1 seconds. Scheduling system reboot in %2 seconds. |
| | The Node Manager has itself crashed after having run for %1 seconds. The machine will be rebooted after waiting %2 seconds. | | | Contact the Support Center. |
| 102D102* | Raise | Error | NM INITIALIZING | %2 Node Manager crashed after having been up for %1 seconds. Auto-reboot is disabled. Will attempt service restart. |
| | The Node Manager has itself crashed after having run for %1 seconds. The machine cannot be rebooted since auto-reboot is disabled. The Node Manager Manager will attempt to restart the service. | | | Contact the Support Center. |
| 102D103* | Raise | Error | NM INITIALIZING | %3 Node Manager requested reboot after having been up for %1 seconds. Scheduling system reboot in %2 seconds. |

| | | | | |
|---|---|---|---|---|
| | The Node Manager has requested the machine be rebooted after having run for %1 seconds. The machine will be rebooted after waiting %2 seconds. | | | Contact the Support Center. |
| 102D104* | Raise | Error | NM INITIALIZING | %2 Node Manager requested reboot after having been up for %1 seconds. Auto-reboot is disabled. Will attempt service restart. |
| | The Node Manager has requested the machine be rebooted after having run for %1 seconds. The machine cannot be rebooted since auto-reboot is disabled. The Node Manager Manager will attempt to restart the service. | | | Contact the Support Center. |
| 102D105* | Raise | Error | NM INITIALIZING | %2 A Critical Process has requested a reboot after the service has been up for %1 seconds. Auto-reboot on Process Request is disabled. Will attempt service restart. |
| | A Critical Process has requested a reboot after the service has been up for %1 seconds. The machine cannot be rebooted since Auto-reboot on Process Request is disabled. The Node Manager Manager will attempt to restart the service. | | | Contact the Support Center. |
| 102D106* | Raise | Error | NM INITIALIZING | %3 A Critical Process has requested a reboot after having been up for %1 seconds. Scheduling system reboot in %2 seconds. |
| | A Critical Process has requested the machine be rebooted after having run for %1 seconds. The machine will be rebooted after waiting %2 seconds. | | | Contact the Support Center. |
| 1040010* | Raise | Warning | MDS SYNCH CONNECT TIMEOUT | Synchronizer timed out trying to establish connection to peer. |
| | The MDS message synchronizer was unable to connect to its duplexed partner within the timeout period. Either the duplexed partner is down, or there is no connectivity to the duplexed partner on the private network. | | | Verify reliable network connectivity on the private network. Call the Cisco Systems, Inc. Customer Support Center in the event of a software failure on the duplexed partner. |
| 1040022* | Raise | Error | MDS SYNCH CONNECT TIMEOUT | Connectivity with duplexed partner has been lost due a failure of the private network, or duplexed partner is out of service. |
| | The MDS message synchronizer has lost connectivity to its duplexed partner. This indicates either a failure of the private network, or a failure of the duplexed partner. | | | Confirm services are running on peer machine. Check MDS process to determine if it is paired or isolated. Ping test between peers over the private network. Check PGAG and MDS for TOS (Test Other Side) messages indicating the private network has failed and MDS is testing the health of the peer over the public network. |
| 1040023* | Clear | Informational | MDS SYNCH CONNECT TIMEOUT | Communication with peer Synchronizer established. |
| | The MDS message synchronizer has established communication with its duplexed partner. | | | No action is required. |
| 105007D* | Clear | Informational | RTR PERIPHERAL | Peripheral %2 (ID %1) is on-line. |
| | The specified peripheral is on-line to the ICM. Call and agent state information is being received by the Router for this site. | | | No action is required. |
| 105007E* | Raise | Error | RTR PERIPHERAL | ACD/IVR %2 (ID %1) is off-line and not visible to the Peripheral Gateway. Routing to this site is impacted. |
| | The specified ACD/IVR is not visible to the Peripheral Gateway. No call or agent state information is being received | | | ACD/IVR Vendor should be contacted for resolution. If Peripheral Gateway is also offline per messaging |

| | | | | |
|---|---|---|---|---|
| | by the Router from this site. Routing to this site is impacted. | | | (message ID 10500D1) or rttest then proceed with troubleshooting for Peripheral Gateway off-line alarm first. |
| 10500D0* | Clear | Informational | RTR PHYSICAL CONTROLLER | Physical controller %2 (ID %1) is on-line. |
| | The Router is reporting that physical controller %2 is on-line. | | | No action is required. |
| 10500D1* | Raise | Error | RTR PHYSICAL CONTROLLER | Peripheral Gateway %2 (ID %1) is not connected to the Central Controller or is out of service. Routing to this site is impacted. |
| | The specified Peripheral Gateway is not connected to the Central Controller. It could be down. Possibly it has been taken out of service. Routing to this site is impacted. | | | Communication (network) between the Central Controller (Router) and the PG should be checked using 'ping' and 'tracert'. Must have visible and visible high priority connection from PG to Route. CCAG process on Router and PGAG process on PG should be checked. PG may have been taken out of service for maintenance. |
| 10500D2* | Clear | Informational | RTR PERIPHERAL | PG has reported that peripheral %2 (ID %1) is operational. |
| | PG has reported that peripheral %2 (ID %1) is operational. | | | No action is required. |
| 10500D3* | Raise | Error | RTR PERIPHERAL | PG has reported that peripheral %2 (ID %1) is not operational. |
| | This may indicate that the peripheral is off-line for maintenance or that the physical interface between the peripheral and the PG is not functioning. | | | Check that the peripheral is not itself off-line and that the connection from the peripheral to the PG is intact. |
| 10500F6 | Raise | Informational | RTR SCRIPT TABLE | ScriptTable %2 (ID %1) is available only on side A. |
| | ScriptTable %2 is only available on the side A Router. If the side A Router goes down, no DB Lookup requests can be processed as side B cannot access the ScriptTable. | | | You probably want to configure a ScriptTable on side B that is identical to that on side A. |
| 10500F7 | Raise | Informational | RTR SCRIPT TABLE | ScriptTable %2 (ID %1) is available only on side B. |
| | ScriptTable %2 is only available on the side B Router. If the side B Router goes down, no DB Lookup requests can be processed as side A cannot access the ScriptTable. | | | You probably want to configure a ScriptTable on side A that is identical to that on side B. |
| 10500F8 | Raise | Error | RTR SCRIPT TABLE | ScriptTable %2 (ID %1) is not available on either side. |
| | No DB Lookup requests can be processed as ScriptTable %2 is unavailable on either side of the central controller. | | | Configure a ScriptTable on either side A or side B, preferably both. |
| 10500F9 | Clear | Informational | RTR SCRIPT TABLE | ScriptTable %2 (ID %1) is available on both sides A & B. |
| | ScriptTable %2 is configured on both sides of the central controller. | | | No action is required. |
| 10500FF* | Clear | Informational | RTR PTOCESS OK | Side %1 %2 process is OK. |
| | The Router is reporting that side %1 process %2 is OK. | | | No action is required. |
| 1050100* | Raise | Error | RTR PROCESS OK | Process %2 at the Central Site side %1 is down. |
| | The specified process at the central controller site is down. The central controller side is indicated. Attempts will be made to automatically restart the process. | | | This alarm only occurs for Central Controller (Router and Logger) processes. If the process for BOTH sides is down there is a total failure for that process. Critical processes include: - 'mds' - Router - Message Delivery Service coordinates messaging between duplexed Routers AND Loggers. When this process is down the Central Controller is down and no routing logic is occurring via ICM. - 'rtr' - Router - call routing intelligence. - 'clgr / hlgr' - Logger - configuration / historical data processing to configuration database. - |

| | | | | |
|---|---|---|---|---|
| | | | | 'rts' - Router - Real Time Server data feed from the router to the Admin Workstations of reporting. - 'rcv' - Logger Recovery - the process that keeps the redundant historical databases synchronized between duplexed loggers. |
| 10501F1* | Clear | Informational | RTR NODE | ICM Node %2 (ID %1) is on-line. |
| | The specified node is on-line to the ICM. | | | No action is required. |
| 10501F2* | Raise | Error | RTR NODE | ICM Node %2 (ID %1) is off-line. |
| | The specified node is not visible to the ICM. Distribution of real time data may be impacted. | | | No action is required. |
| 10501F6 | Clear | Informational | RTR STATE SIZE OK | The router's state size of %1 mb is now below the alarm limit of %2 mb. |
| | The router's state size of %1 mb is now below the alarm limit of %2 mb. | | | No action is required. |
| 10501F7 | Raise | Error | RTR STATE SIZE OK | The router's state size of %1 mb has grown beyond the alarm limit of %2 mb. |
| | The router's state size of %1 mb has grown beyond the alarm limit of %2 mb. This may indicate a memory leak, or it may be indicate that the customers configuration size has grown larger. The alarm limit can be raised with the rtsetting tool. Large state sizes may cause problems when synchronizing routers, so the bandwidth of the private link may also need to be investigated. | | | Contact the Support Center. |
| 10501F8* | Clear | Informational | RTR NODE | ICM Node %2 (ID %1) on system %3 is on-line. |
| | The specified node is on-line to the ICM. | | | No action is required. |
| 10501F9* | Raise | Error | RTR NODE | ICM Node %2 (ID %1) on system %3 is off-line. |
| | The specified node is not visible to the ICM. Distribution of real time data may be impacted. | | | No action is required. |
| 10501FD | Clear | Informational | RTR ROUTER CONFIGURED | The router has completed loading the initial configuration from the logger. |
| | The specified node is on-line to the ICM. | | | No action is required. |
| 10501FE | Raise | Error | RTR ROUTER CONFIGURED | The router has not loaded a configuration from the logger. |
| | This condition indicates that the router has not yet completed the initialization step of loading a configuration from the logger. It is normal for this condition to exist briefly while the system is loading. If it does not clear, it may indicate a problem with the logger machine, or the communications paths that connect the rotuer and logger. | | | No action is required. |
| 105023C* | Single-State Raise | Error | RTR SYNC CHECK | The router has detected that it is no longer synchronized with its partner. |
| | The router has detected that it is no longer synchronized with its partner. One result of this is that the router might be routing some calls incorrectly. | | | Recommended action: Stop the router on both sides. After both sides are completely stopped, restart both routers. Alternate Action: Restart the router on one side. After doing this, the routers might still route some calls incorrectly, but they will be in sync. Other actions: Collect all rtr, mds, ccag process logs from both routers from the entire day. Collect all sync*.sod files (where * is some number) that exist in the icr\<instance>\ra directory of |

| | | | | |
|---|---|---|---|---|
| | | | | router A and in the icr\<instance>\rb directory of router B. Contact the Support Center. |
| 106003A | Raise | Error | AW W3SVC | World Wide Web Publishing Service may be down. ICM cannot communicate with web server. |
| | World Wide Web Publishing Service may be down. ICM cannot communicate with web server. | | | Start World Wide Web Publishing Service if it is not running. Otherwise, look for messages in the IIS error log. |
| 106003B | Clear | Informational | AW W3SVC | World Wide Web Publishing Service is up. |
| | World Wide Web Publishing Service is up. | | | No action is required. |
| 108C020* | Clear | Informational | OPC CTI SERVER | The Enterprise CTI Server associated with this Peripheral Gateway is on-line on %1. |
| | The Enterprise CTI server associated with this Peripheral Gateway is on-line. Enterprise CTI Client applications are able to connect to the server and exchange call and agent data. | | | No action is required. |
| 108C021* | Raise | Error | OPC CTI SERVER | The Enterprise CTI server associated with this Peripheral Gateway is down. |
| | The Enterprise CTI server associated with this Peripheral Gateway is off-line. Enterprise CTI Client applications are not able to connect to the server and exchange call and agent data. | | | No action is required. |
| 10F8004 | Clear | Informational | DMP DEVICE PATH IDLE | Device %1 path changing to idle state. |
| | The indicated device is using this side of the Central Controller for its idle communication path (and is therefore using the other side of the Central Controller for its active communication path). | | | No action is required. |
| 10F8005 | Clear | Informational | DMP DEVICE PATH IDLE | Device %1 path changing to active state. |
| | The indicated device is using this side of the Central Controller for its active communication path. | | | No action is required. |
| 10F8007 | Raise | Error | DMP DEVICE PATH IDLE | Device %1 path realignment failed. |
| | The indicated device failed to realign its message stream to this side of the Central Controller. | | | No action is required. |
| 10F8008 | Raise | Error | DMP DEVICE PATH IDLE | Device %1 disconnected. |
| | The indicated device has been disconnected from this side of the Central Controller. This may be caused by a network problem or device failure. | | | Remedy network problems, if any. Call the Cisco Systems, Inc. Customer Support Center in the event of a software failure on the device. |
| 10F800E | Raise | Warning | DMP DEVICE PATH IDLE | Device %1 path reset. |
| | The communication path between this side of the Central Controller and the indicated device has been reset to an initial state. | | | No action is required. |
| 10F800F | Clear | Informational | DMP DEVICE PATH IDLE | Device %1 initializing message stream. |
| | The indicated device is initializing its message stream with this side of the Central Controller. | | | No action is required. |
| 10F8018 | Raise | Error | DMP DEVICE PATH IDLE | Device %1 is not acknowledging data. Breaking device connection. |

| | | | | |
|---|---|---|---|---|
| | The indicated device has failed to acknowledge messages from this side of the Central Controller. The connection to the device will be forcibly reset. This usually indicates severe performance problems and/or hardware problems on the indicated device. | | | Run diagnostics on the hardware. |
| 10F801A | Raise | Error | DMP DEVICE PATH IDLE | Device %1 failed to acknowledge multiple roll-forward requests. Breaking device connection. |
| | The indicated device has failed to acknowledge multiple DMP protocol messages from this side of the Central Controller. The connection to the device will be forcibly reset. | | | No additional corrective action is necessary. Frequent or continuous occurrences suggest severe performance problems and/or hardware problems on the indicated device, in which case diagnostic tools should be used to find the cause. |
| 10F801D | Raise | Warning | DMP DEVICE PATH IDLE | The Network communications between the Peripheral Gateway or NIC %2 has been down for %1 minutes. |
| | No communication path from the indicated device to this side of the Central Controller has existed for the indicated time period. This indicates either an extended network outage or an extended outage at the device. | | | One or more network links between the named device and the named side of the ICM Router has failed. If alarms exist for BOTH Routers the site is offline. If alarms exist for one side of the Router then the site should be up but network redundancy is degraded. Communication (network) between the Central Controller (Router) and the PG should be checked using 'ping' and 'tracert'. Must have visible and visible high priority connection from PG to Route. CCAG process on Router and PGAG process on PG should be checked. |
| 118C002 | Single-State Raise | Informational | LGR FREE SPACE | %1%% of the available free space is used in %2 database. |
| | %1%% of the available free space is used in %2 database. This is an indication of how full the database is. When this value gets too high, the Logger will begin deleting the oldest historical records from the database. | | | No action is required. |
| 118C00C | Single-State Raise | Informational | LGR LOG SPACE | %1%% of the available log space is used in %2 database. |
| | %1%% of the available log space is used in %2 database. | | | No action is required. |
| 118C00F | Raise | Warning | LGR BEGIN AUTOPURGE | Begin Automatic Purge: %1%% of the available data space is used in the %2 database. |
| | Automatic Purge is being run in order to keep the database from running out of space. The parameters for the daily purge need to be adjusted to match the database storage capacity. | | | Contact the Support Center. |
| 118C010 | Clear | Warning | LGR BEGIN AUTOPURGE | Automatic Purge Complete: %1%% of the available data space is used in the %2 database. |
| | Automatic Purge has been run in order to keep the database from running out of space. The parameters for the daily purge need to be adjusted to match the database storage capacity. | | | No action is required. |
| 118C015 | Clear | Informational | LGR CONNECTED CLIENT | Connected To Client on %1 using port %2. |
| | The Logger has successfully connected to a client for the SQL Server. | | | No action is required. |
| 118C017 | Raise | Informational | LGR CONNECTED CLIENT | Logger or HDS on connection %1 using TCP/IP port %2 is either out of service or communication has broken. |

| | | | | |
|---|---|---|---|---|
| | Logger or HDS on the specified TCP/IP connection and port number is either out of service or communication has broken. | | | The Historical Data Server (HDS) or the peer Logger ( on the other side of the duplexed central controller ) is no longer getting its historical feed from this Logger. This can occur due to networking outages, SQL issues on the Logger or HDS, or the Logger or HDS may have been shut down or otherwise disabled. |
| 118C040 | Single-State Raise | Warning | LGR MISSING NETWORK ROUTING CLIENT | Found %1 records with DateTime greater than current Central Controller Time %2 in %3 table. Check and correct the errors. |
| | Found historical records with DateTime greater than current Central Controller Time. Delete the records which have date time greater than the current central controller time. | | | No action is required. |
| 118C04F | Raise | Warning | LGR HDS RUN BEHIND | HDS Running Behind: %1 is running behind its logger %2 by %3 minutes. |
| | Historical Database Server replicates behind its Logger by the time period specified in the registry. The HDS running status needs to be checked and/or the performance of both HDS and Logger needs to be monitored. The alarm controlling parameters may need to be adjusted to satisfy the specific requirement. | | | Verify HDS is running correctly. Check the performance of both Logger and HDS. If the HDS has been shut down purposely, the alarm controlling parameters need to be adjusted on the Logger in order to avoid additional alarms. |
| 12B001F | Clear | Error | APPGW APPGW | Application Gateway has connected with the host. Application Gateway ID = %1 |
| | The application gateway is now connected to the host process. | | | No action is required. |
| 12B0020 | Raise | Error | APPGW APPGW | The external database has disconnected from the Application Gateway (ID = %1). Routing may be impacted. |
| | An external database used in some Scripts has disconnected from the specified Application Gateway. Error recovery mechanisms will attempt to reconnect. Routing may be impacted. | | | Support group for external database should be contacted. If host database has been off line for extended period, re-starting Application Gateway process may be necessary to re-connect. |
| 12E8006 | Clear | Informational | CTI SESSION | CONNECTION MONITOR SERVICE: Enterprise CTI session established by Client %1 (%2) at %3. |
| | An Enterprise CTI session has been opened by ClientID %1 (Signature %2) from IP address %3. | | | No action is required. |
| 12E8007 | Raise | Warning | CTI SESSION | CONNECTION MONITOR SERVICE: Enterprise CTI session closed by Client %1 (%2) at %3. |
| | The Enterprise CTI session with ClientID %1 (Signature %2) at IP address %3 has been closed by the client. | | | This indicates that an Enterprise CTI Client application that is normally always connected to the Enterprise CTI Server has closed its connection. The CTI Client application software may need to be checked for proper operation. |
| 12E8008 | Raise | Error | CTI SESSION | CONNECTION MONITOR SERVICE: Enterprise CTI session terminated with Client %1 (%2) at %3. |
| | The Enterprise CTI session with ClientID %1 (Signature %2) at IP address %3 has been terminated by the Enterprise CTI Server. | | | This indicates that an Enterprise CTI Client application that is normally always connected to the Enterprise CTI Server has been disconnected due to errors. If the problem persists, the CTI Client application software may need to be checked for proper operation. |
| 12E800C | Clear | Informational | CTI NORMAL OBJECT EVENT | Client:%1 Object:%2 Normal Event Report: %3 |

| | | | | |
|---|---|---|---|---|
| | The Enterprise CTI client %1 application software has reported the following normal event for object %2: %3. | | | No action is required. |
| 12E800D | Raise | Warning | CTI NORMAL OBJECT EVENT | Client:%1 Object:%2 Warning Event Report: %3 |
| | The Enterprise CTI client %1 application software has reported the following warning for object %2: %3. | | | This indicates that the CTI Client application software has detected a possible error or other abnormal condition and may need to be checked for proper operation. |
| 12E800E | Raise | Error | CTI NORMAL OBJECT EVENT | Client:%1 Object:%2 Error Event Report: %3 |
| | The Enterprise CTI client %1 application software has reported the following error for object %2: %3. | | | This indicates that the CTI Client application software has detected an error condition and may need to be checked for proper operation. |
| 13E0002 | Raise | Error | MEISVR CONNECT | Message Integration Service (MIS) was unable to connect to %1%2 on %3 TCP/IP Port %4. |
| | Message Integration Service was unable to connect to the indicated component and address. | | | Confirm Component is available, Configuration of IP address(es) and Port(s) are correct, and Network connectivity would allow for connection |
| 13E0003 | Clear | Informational | MEISVR CONNECT | Connection to %1%2 on Address[%3:%4] Succeeded. |
| | Message Integration Service was able to connect to the indicated component and address. | | | No action is required. |
| 13E0004 | Raise | Error | MEISVR SESSION | Message Integration Service (MIS) was unable to open a session to %1%2. |
| | Message Integration Service was unable to open a session to the indicated component | | | No action is required. |
| 13E0005 | Clear | Informational | MEISVR SESSION | Session to %1%2 Opened. |
| | Message Integration Service was able to open a session to the indicated component and address. | | | No action is required. |
| 13E0006 | Single-State Raise | Error | MSGIS NON CONFIGURED TRUNKGROUP | TrunkGroup:%1 Trunk:%2 Received in Msg from Vru-%3 Not Configured |
| | A message pertaining to the indicated trunk group and trunk has not been configured with MIS | | | Configure Extension, Trunk Group, and Trunk in MIS |
| 13E0007 | Single-State Raise | Error | MSGIS CALL TRACKING ERROR | Call Tracking Error: %1 |
| | A call within MIS could not be tracked successfully. | | | Determine where tracking problem occurred and correct (For MIS problem could be MIS, VRU, or PG) |
| 1438000 | Raise | Error | CAMPAIGN MANAGER | Blended Agent Campaign Manager on [%1] is down. |
| | The Blended Agent Campaign Manager is not running. Dialer(s) will only run for a short period of time without a Campaign Manager. In addition, configuration messages will not be forwarded to Dialer(s) or the Import process. | | | Make sure the Campaign Manager process is enabled in the registry. Also, check that the Blended Agent database server is running. The Blended Agent private database should have been created with the ICMDBA tool. |
| 1438001 | Clear | Error | CAMPAIGN MANAGER | Blended Agent Campaign Manager on [%1] is up. |
| | Blended Agent Campaign Manager is ready to distribute customer records and configuration data. | | | No action is required. |
| 1438002 | Raise | Error | BA IMPORT | Failed to execute import into table [%1] due to a change |

| | | | | |
|---|---|---|---|---|
| | | | | in the tables' schema. |
| The schema for a specified table has been changed but the overwrite option has not been enable. This means that an existing database table does not match the configured import. | | | | Change the import to an overwrite import. This will drop the existing customer table and create a new table that will match the import. Please note that all existing customer data for that import will be lost. |
| 1438003 | Raise | Error | BA IMPORT | Import failed due to an invalid table [%1] definition. |
| Could not create the specified table due to invalid import schema definition. | | | | Check that all table columns for the failed import are correct. Matching a character column too long could cause this failure. |
| 1438004 | Clear | Error | BA IMPORT | The import for table [%1] has been successful. |
| An import has completed successfully. | | | | No action is required. |
| 1438005 | Raise | Error | BA IMPORT | Failed to import data into table [%1]. |
| This error could occur if the import file did not match the table definition. | | | | Check that the import table definition matches the import file. |
| 1438006 | Raise | Error | BA IMPORT | Failed to build dialing list from table [%1]. |
| A Dialing list could not be populated from the specified table. | | | | Check if another process has the dialing list table locked. For example, if a report was running on the table while the dialing list was being generated. |
| 1438008 | Raise | Error | BA DIALER NETWORK | Could not connect to Campaign Manager. |
| Either the Campaign Manager is not running or a network connection cannot be established due to connectivity issues. | | | | Check that the Campaign Manager is up. Make sure ICM instance numbers match between Campaign Manager and Dialer. Check the network cabling on the Dialer. Ping the Campaign Manager computer from the Dialer computer. |
| 1438009 | Raise | Error | BA IMPORT | Could not open [%1] database. |
| The Blended Agent private database has not been initialized or SQL Server is not running. | | | | Make sure SQL Server is running. Check the ODBC configuration settings for the Blended Agent private database. Was the ICMDBA tool run to create the Blended Agent private database? |
| 1438010 | Raise | Error | BA IMPORT | An import was started but its configuration was deleted while it was running. |
| An import started running but part of its configuration was deleted before it was able to do anything. | | | | Reschedule the import. |
| 1438011 | Raise | Error | BA CTI | Blended Agent CTI Server connection on computer [%1] is down. |
| The Blended Agent CTI Server connection has been terminated. | | | | Make sure CTI Server is active. Also make sure the PIM has connectivity to the switch. |
| 1438012 | Clear | Error | BA CTI | Blended Agent CTI Server connection on computer [%1] is active. |
| Blended Agent CTI Server connection is active. | | | | No action is required. |
| 1438017 | Raise | Error | CAMPAIGN MANAGER IMPORT | Process is down on computer [%1]. |
| Process is not running on the specified computer. | | | | Please check that the Import process has not been shutdown. Check that the BA Private database has been created. Also, check that SQL Server is running. |
| 1438018 | Clear | Error | CAMPAIGN MANAGER IMPORT | Process is up on computer [%1]. |
| Process is running on the specified computer. | | | | No action is required. |

| | | | | |
|---|---|---|---|---|
| 1438019 | Raise | Error | CAMPAIGN MANAGER DIALER | Process is down on computer [%1]. |
| | Process is down on the specified computer. | | | Verify that the Dialer has been started by Node Manager. |
| 1438020 | Clear | Error | CAMPAIGN MANAGER DIALER | Process is up on computer [%1]. |
| | Process is up on the specified computer. | | | No action is required. |
| 1438030 | Raise | Error | BA DIALER MR | MR PIM disconnected from Dialer [%1]. |
| | MR PIM disconnected from Dialer <dialer name>. | | | No action is required. |
| 1438031 | Clear | Error | BA DIALER MR | MR PIM connected to Dialer [%1]. |
| | MR PIM connected to Dialer <dialer name>. | | | No action is required. |
| 1438032 | Raise | Error | BA DIALER MR | MR Routing disabled on Dialer [%1]. |
| | MR Routing disabled on Dialer <dialer name>. | | | No action is required. |
| 1438033 | Clear | Error | BA DIALER MR | MR Routing enabled on Dialer [%1]. |
| | MR Routing enabled on Dialer <dialer name>. | | | No action is required. |
| 1438034 | Raise | Error | BA DIALER CALLMGR | Dialer [%1], Port [%2], extension [%3] disconnected from Callmgr [%4]. |
| | Dialer <dialer name>, Port <port number>, extension <extension> disconnected from Callmgr <call manager name>. | | | No action is required. |
| 1438035 | Clear | Error | BA DIALER CALLMGR | Dialer [%1], Port [%2], extension [%3] connected to Callmgr [%4]. |
| | Dialer <dialer name>, Port <port number>, extension <extension> connected to Callmgr <call manager name>. | | | No action is required. |
| 1438036 | Raise | Error | BA DIALER CALLMGR | Dialer [%1], Port [%2], extension [%3] failed to registered with Callmgr [%4]. |
| | Dialer <dialer name>, Port <port number>, extension <extension> failed to registered with Callmgr <call manager name>. | | | No action is required. |
| 1438037 | Clear | Error | BA DIALER CALLMGR | Dialer [%1], Port [%2], extension [%3] registered with Callmgr [%4]. |
| | Dialer <dialer name>, Port <port number>, extension <extension> registered with Callmgr <call manager name>. | | | No action is required. |
| 1438038 | Single-State Raise | Error | BA IMPORT | Failed to rename or delete the import file for Import Rule Id: %1. This Import Rule has been temporarily disabled. To correct this condition: manually remove the import file and disable and re-enable the import rule using Import Configuration Component. |
| | Failed to rename or delete the import file for Import Rule Id: <id; filename>. This Import Rule has been temporarily disabled. To correct this condition: manually remove the import file and disable and re-enable the import rule using Import Configuration Component. | | | File polling is enabled for this import rule. After the import, the BAImport process was unable to rename or delete the file. This import rule is temporarily disabled. Rename or delete the import file, disable and re-enable this import rule from the BAImport Configuration Component. |
| 12A0003 | Heart beat | 0x00 | - | HeartBeat Event for %1 |
| | Periodic message to indicate MDS is in service and that the event stream is active. | | | No action is required. |