



Serviceability Best Practices Guide for Cisco Unified ICM/Contact Center Enterprise & Hosted Release 8.0(1)

Last Updated: February 2010

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 SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS 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.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

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.

CCDE, CCENT, CCSI, Cisco Eos, Cisco HealthPresence, Cisco IronPort, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco Nurse Connect, Cisco StackPower, Cisco StadiumVision, Cisco TelePresence, Cisco Unified Computing System, Cisco WebEx, DCE, Flip Channels, Flip for Good, Flip Mino, Flip Video, Flip Video (Design), Flipshare (Design), Flip Ultra, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn, Cisco Store, and Flip Gift Card are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company(1002R).

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.

Serviceability Best Practices Guide for Cisco Unified ICM/Contact Center Enterprise & Hosted for Release 8.0(1)

Copyright © 2005-2010, Cisco Systems, Inc.

All rights reserved

Table of Contents

1	INTRODUCTION	9
1.1	TARGET AUDIENCE	9
1.2	DOCUMENT INTENT AND FOCUS	9
1.3	PRODUCT NAMES	9
2	PRODUCT ARCHITECTURE	11
2.1	OVERVIEW – CISCO UNIFIED CONTACT CENTER	11
2.2	ROUTER	12
2.2.1	Network Interface Controller	13
2.3	LOGGER	13
2.4	PERIPHERAL GATEWAY	14
2.4.1	Open Peripheral Controller	16
2.4.2	Peripheral Interface Manager	16
2.4.3	JTAPI Gateway	17
2.4.4	CTI Gateway (CTI Server)	17
2.4.5	CTI Object Server	17
2.4.6	Cisco Agent Desktop	18
2.5	CONFIGURATION SYSTEM	19
2.5.1	Administration & Data Server	19
2.5.2	Configuration Updates	20
2.6	REPORTING SYSTEM	21
2.6.1	Historical Data Server	22
2.6.2	WebView (Enterprise Reporting)	22
2.6.3	Unified Intelligence Suite	23
2.6.4	Unified Contact Center Management Portal	27
2.7	OUTBOUND OPTION	30
3	MONITORING SNMP HEALTH	32
3.1	SNMP OVERVIEW	32
3.1.1	Faults	32
3.1.2	Instrumentation	34
3.2	BASE-LEVEL SNMP MIB SUPPORT	34
3.2.1	SNMP Master Agent	34
3.2.2	Base Level SNMP Subagents	34
3.3	CISCO-CONTACT-CENTER-APPS-MIB	37
3.3.1	CISCO-CONTACT-CENTER-APPS-MIB Overview	37
3.3.2	CISCO-CONTACT-CENTER-APPS-MIB Structure	38
3.3.3	Mapping CCCA-MIB to Standard Host MIBs	41
3.3.4	CISCO-CONTACT-CENTER-APPS-MIB Object Descriptions	43
3.4	CONFIGURING THE SNMP AGENTS	60
3.4.1	Installation Prerequisites for SNMP Support	60
3.4.2	Installing the Windows SNMP Component on Windows 2000 Server	61
3.4.3	Installing the Windows SNMP Components on Windows Server 2003	61
3.4.4	SNMP Agent Configuration	61
4	UNDERSTANDING UNIFIED ICM/CC SNMP NOTIFICATIONS	70
4.1	UNIFIED ICM/CC NOTIFICATION TYPE	70
4.2	DUAL STATE OBJECTS	72
4.3	CORRELATING DUAL STATE NOTIFICATIONS	74
4.4	SINGLE STATE OBJECTS	75
4.5	ORGANIZING SNMP NOTIFICATIONS	76
4.6	CSFS HEARTBEAT NOTIFICATION	76

5	THE SYSLOG MESSAGING INTERFACE	78
5.1	THE CISCO LOG MESSAGE FORMAT	78
5.2	CONFIGURING SYSLOG DESTINATIONS	79
6	UNIFIED ICM/CC SERVICES AND PROCESSES	81
6.1.1	<i>Services</i>	81
6.2	USING THE LOCAL DESKTOP	86
6.3	ICM SERVICE CONTROL AND WINDOWS TASK MANAGER	86
6.4	USING THE LOCAL REGISTRY	89
6.5	USING THE REMOTE SNMP MANAGEMENT STATION	90
7	UNIFIED ICM/ UNIFIED CC TRACE LEVELS	92
7.1	TRACE LEVELS CONFIGURATIONS.....	93
7.1.1	<i>Trace: All Nodes</i>	93
7.1.2	<i>Trace: Administration & Data Server (AKA Distributor AW)</i>	93
7.1.3	<i>Trace: Router</i>	94
7.1.4	<i>Trace: Logger</i>	94
7.1.5	<i>Trace: Peripheral Gateway</i>	95
7.1.6	<i>Diagnostic Framework</i>	97
7.2	SETTING ROUTER TRACING.....	98
7.3	SETTING OPC TRACING	99
7.3.1	<i>General Diagnostics</i>	99
7.3.2	<i>Diagnosing Network Transfer Issues</i>	99
7.3.3	<i>Diagnosing Multi Media Issues</i>	99
7.3.4	<i>Diagnosing VRU PG Issues</i>	99
7.4	SETTING UNIFIED CCM PIM TRACING	100
7.4.1	<i>ARS Gateway Registry Trace Settings</i>	101
7.4.1	<i>ARS PIM Trace Settings</i>	101
7.5	SETTING JTAPI GATEWAY TRACING	101
7.6	SETTING CTI SERVER TRACING	102
7.7	SETTING CTI OS TRACING.....	102
7.8	SETTING VRU PIM TRACING.....	103
7.9	SETTING OUTBOUND OPTION TRACING	103
7.9.1	<i>Setting CampaignManager Tracing</i>	103
7.9.2	<i>Setting balImport Tracing</i>	104
7.9.3	<i>Setting Dialer Tracing</i>	104
7.10	SETTING TRACE FILE RETENTION PARAMETERS	105
8	PERFORMANCE COUNTERS	107
8.1	PLATFORM HEALTH MONITORING COUNTERS	107
8.2	PLATFORM DIAGNOSTIC COUNTERS – AUTOMATIC COLLECTION	108
8.3	PLATFORM DIAGNOSTIC COUNTERS	109
8.3.1	<i>All Components</i>	109
8.3.2	<i>Logger / Administration & Data Server / HDS</i>	110
8.3.3	<i>SQL Server</i>	111
8.3.4	<i>WebView</i>	111
8.4	COMPONENT-SPECIFIC COUNTERS.....	112
8.4.1	<i>Router</i>	112
8.4.2	<i>Logger</i>	113
8.4.3	<i>Administration & Data Server</i>	113
8.4.4	<i>PG – OPC</i>	115
8.4.5	<i>PG – Communications Manager (EA) PIM</i>	117
8.4.6	<i>PG – VRU PIM</i>	117
8.4.7	<i>CTI Server</i>	118
8.4.8	<i>CTI OS Server</i>	119
8.4.9	<i>Outbound Option Campaign Manager</i>	123

8.4.10	<i>Outbound Option Import</i>	123
8.4.11	<i>Outbound Option Dialer</i>	124
8.4.12	<i>Message Delivery Service</i>	125
8.4.13	<i>QoS</i>	135
9	CAPACITY PLANNING	138
9.1	CAPACITY PLANNING PROCESS.....	139
9.2	CAPACITY PLANNING – GETTING STARTED	140
9.3	CATEGORIZING COLLECTED DATA	141
9.3.1	<i>Current Deployment Design</i>	141
9.3.2	<i>Configuration Information</i>	142
9.3.3	<i>Traffic Load</i>	143
9.3.4	<i>Migration Requirements</i>	143
9.3.5	<i>Platform Performance</i>	144
9.4	CALCULATING CAPACITY UTILIZATION.....	144
9.4.1	<i>Calculating CPU Utilization</i>	145
9.4.2	<i>Calculating Memory Utilization</i>	146
9.4.3	<i>Calculating Disk Utilization</i>	146
9.4.4	<i>Calculating NIC Utilization</i>	147
9.4.5	<i>Calculating Maximum Utilization</i>	147
9.4.6	<i>Relating Traffic Load to Resources</i>	147
10	UNIFIED ICM/CC DIAGNOSTIC TOOLS	148
10.1	DIAGNOSTIC FRAMEWORK	148
10.1.1	<i>Overview</i>	148
10.1.2	<i>Installation and Configuration</i>	148
10.1.3	<i>Security</i>	152
10.1.4	<i>Usage</i>	157
10.1.5	<i>Diagnostic Framework API</i>	171
10.1.6	<i>Diagnostic Framework Troubleshooting</i>	186
10.2	DUMPLOG	186
11	APPENDIX A - CISCO CONTACT CENTER APPLICATIONS MIB RESULTS EXAMPLE	190
12	APPENDIX B – UNIFIED CCE SNMP NOTIFICATIONS	193

List of Tables

Table 1-1: Product Names	9
Table 2-1: CAD Services and Executables	19
Table 3-1: CCCA MIB Base Objects	43
Table 3-2: CCCA MIB Instance Table Objects	44
Table 3-3: CCCA MIB Component Table Objects	44
Table 3-4: CCCA MIB Component Element Table Objects	45
Table 3-5: CCCA MIB Router Table Objects	46
Table 3-6: CCCA MIB NIC Table Objects	48
Table 3-7: CCCA MIB Logger Table Objects	49
Table 3-8: CCCA MIB Administration Server and Real-time Data Server Table Objects	50
Table 3-9: CCCA MIB Peripheral Gateway Table Objects	52
Table 3-10: CCCA MIB Peripheral Interface Manager Table Objects	53
Table 3-11: CCCA MIB CTI Gateway Table Objects	54
Table 3-12: CCCA MIB CTI OS Table Objects	55
Table 3-13: CCCA MIB Outbound Option Campaign Manager Table Objects	57
Table 3-14: CCCA MIB Outbound Option Dialer Table Objects	58
Table 3-15: SNMP General Information Properties	66
Table 4-1: ICM/CC Notification Type Objects	70
Table 4-2: Example "Raise" Notification	73
Table 4-3: Example "Clear" Notification	74
Table 4-4: Example "Single-State Raise" Notification	75
Table 4-5: CSFS Heartbeat Notification	77
Table 5-1: Cisco Log Message Fields	78
Table 6-1: Unified ICM/CC Processes	81
Table 7-1: Trace: All Nodes	93
Table 7-2: Trace: Administration & Data Server (AKA Distributor AW)	93
Table 7-3: Trace: Router	94
Table 7-4: Trace: Logger	94
Table 7-5: Trace: Peripheral Gateway	95
Table 7-6: Diagnostic Framework	97
Table 7-1: Setting Router Tracing	98
Table 7-2: Setting Unified CCM PIM Tracing	100
Table 7-3: Setting ARS Gateway Registry Tracing	101
Table 7-4: Setting ARS PIM Tracing	101
Table 7-5: Setting Unified CM PIM Tracing	101
Table 7-6: Setting CTI Server Tracing	102
Table 7-7: Setting CTI Server Tracing	102
Table 7-8: Setting VRU PIM Tracing	103
Table 7-7: Registry Items	106
Table 8-1: Performance Counters - Health Monitoring	107
Table 8-2: Platform Diagnostic Counters Values	108
Table 8-3: Performance Counters - Diagnostics	108

Table 8-4: Diagnostic Counters - All Components	110
Table 8-5: Diagnostic Counters - Logger, Administration & Data Server, HDS	110
Table 8-6: Diagnostic Counters - SQL Server	111
Table 8-7: Diagnostic Counters - WebView	111
Table 8-8: Router Performance Counters	112
Table 8-9: Logger Performance Counters.....	113
Table 8-10: Administration & Data Server Real-Time Counter	113
Table 8-11: Administration & Data Server Replication Counters	115
Table 8-12: PG - OPC Counters	115
Table 8-13: PG - CM PIM Counters	117
Table 8-14: PG - VRU PIM Counters	117
Table 8-15: CTI Server Counters.....	118
Table 8-16: CTI OS Server Counters	119
Table 8-17: Outbound Option Campaign Manager Counters.....	123
Table 8-18: Outbound Option Import Counters	123
Table 8-19: Outbound Option Dialer Counters	124
Table 8-20: MDS Client Counters	125
Table 8-21: MDS Process Client Counters	126
Table 8-22: MDS Process Counters	127
Table 8-23: Cisco ICM Qos	135
Table 9-1: Calculating CPU Utilization	145
Table 9-2: Calculating Memory Utilization	146
Table 9-3: Calculating Disk Utilization.....	146
Table 9-4: Calculating NIC Utilization	147
Table 10-1: CPU Threshold	151
Table 10-2: Domain Authorization Combination	154
Table 10-3: Diagnostic Framework Certificate Manager Utility Tasks.....	156
Table 10-3: CLI Commands	162
Table 10-5: System Mode Syntax.....	163
Table 10-6: System Commands.....	163
Table 10-7: Device, Protocol and Command Mapping	168
Table 10-8: Mapping of System CLI commands to IOS CLI commands	169
Table 10-9: Trace Levels	172
Table 10-10: Diagnostic Framework Troubleshooting	186
Table 10-11: APPNAME and TAGS Used in DUMPLOG Trace Output	187
Table 12-1: SNMP Notifications	193

List of Figures

Figure 1: Unified CCE Architecture	12
Figure 2: Central Controller Architecture.....	14
Figure 3: Peripheral Gateway Architecture.....	15
Figure 4: DMP Flows.....	16
Figure 5: Configuration System Message Flow.....	20
Figure 6: Reporting Architecture.....	21
Figure 7: WebView Architecture.....	23
Figure 8: CUIS Simple Deployment	25
Figure 9: CUIS Standard Deployment	26
Figure 10: CUIS Scaled Deployment	27
Figure 11: Unified CCMP Architecture	28
Figure 12: Unified CCMP Services	30
Figure 13: Outbound Option Component Relationships.....	31
Figure 14: ICM/CC Event Message Flow.....	33
Figure 15: CISCO-CONTACT-CENTER-APPS-MIB Structure	39
Figure 16: CCCA MIB – Component Inventory Example.....	40
Figure 17: Mapping CCCA MIB Objects to Host MIB Objects.....	41
Figure 18: Mapping CCCA MIB to SYSAPPL MIB	42
Figure 19: SNMP Community Name Configuration Dialog.....	64
Figure 20: SNMP User Name Configuration Dialog	65
Figure 21: SNMP General Information Configuration Dialog	67
Figure 22: SNMP Trap Destination Configuration Dialog	69
Figure 23: syslog Feed Configuration Dialog	80
Figure 24: ICM Service Control.....	87
Figure 25: Windows Task Manager – Applications List	88
Figure 26: Windows Task Manager - Process List	89
Figure 27: Registry Editor	90
Figure 28: Router Trace Utility	98
Figure 29: Capacity Planning Process.....	139
Figure 30: Graph of Samples to Find Busy Hour	141
Figure 31: Real Time Monitoring Tool.....	158
Figure 32: Using Unified System CLI from Command Prompt	159
Figure 33: Unified CLI Architecture.....	160
Figure 34: Unified ICM-CCE-CCH Diagnostic Framework.....	171

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 monitor and manage 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 CCE) 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 CCE 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 are 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	Intelligent Contact Management	Intelligent Call Router	

Current Name	Previous Name	AKA(s)	Notes
Enterprise (Unified ICME)	Enterprise Edition (ICM/E)	(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 CC 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 CC 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 CC is a distributed solution; there is no single-server implementation but rather Unified CCE employs multiple servers each with multiple software components. Deployment options are extremely flexible with performance, capacity and network topology driving the deployment design.

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

The Unified ICM platform was originally designed to route calls between various nodes in the 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's 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 CC 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:

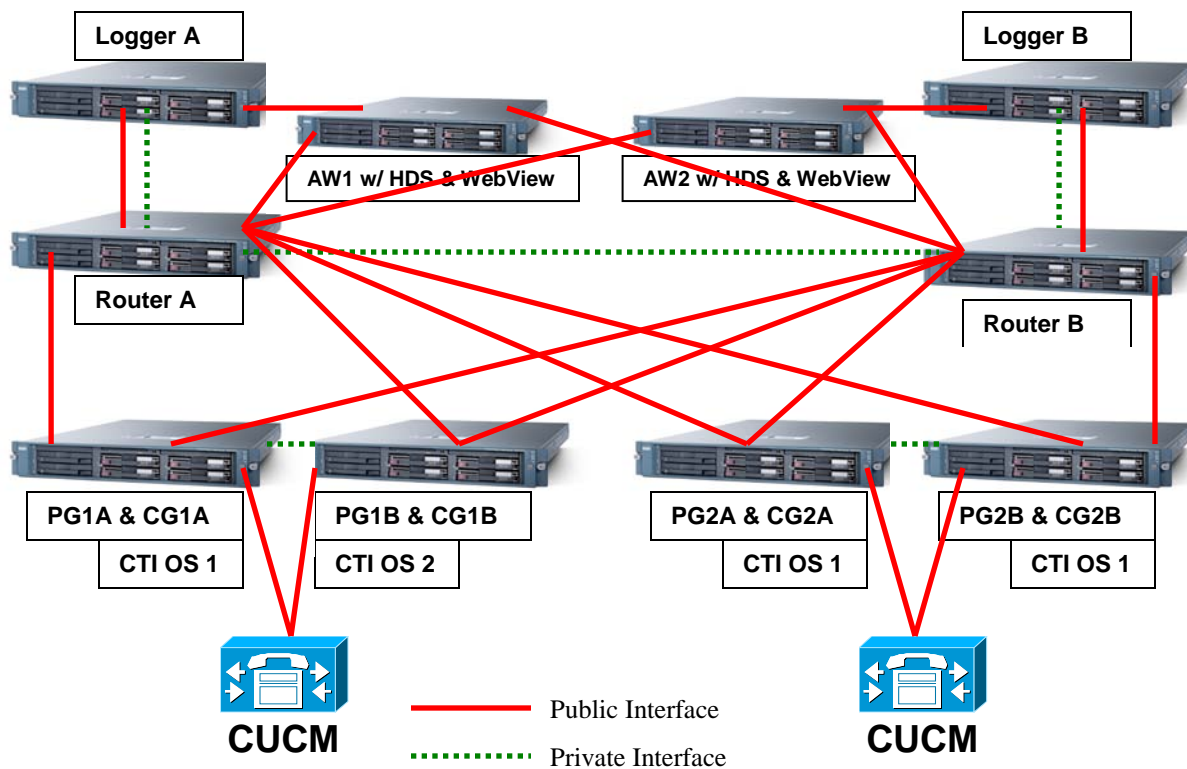


Figure 1: Unified CCE Architecture

There are four major components of a Unified CCE deployment: The Router, the Logger, the Peripheral Gateway (PG) and the Administration & Data Server. 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, chat and so on).
2. Logger: Store (and replicate) all configuration, real-time and historical data.
3. Peripheral Gateway: 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: 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 Administration & Data Server 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 PGs, and Administration and Data Servers (ADS).

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 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 PG, a Network Interface Controller (NIC) is a type of routing client. A NIC is more limited than a PG, however. A NIC’s purpose 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 Router to Logger, using bulk database copy algorithms for efficiency.

The Loggers also distribute historical data to HDS and configuration and real time data to the Administration & Data Servers 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”.

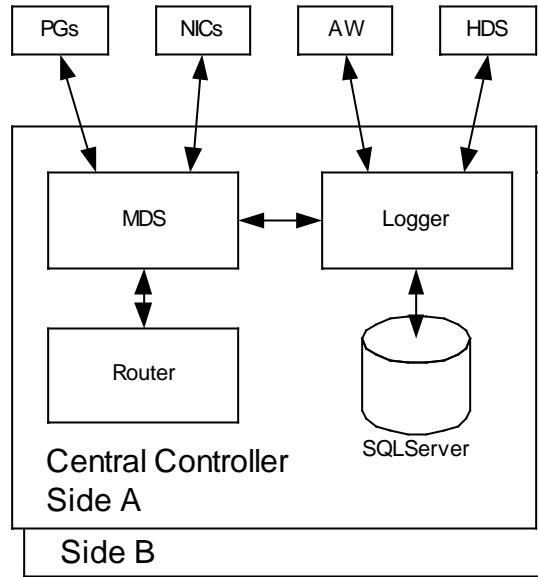


Figure 2: Central Controller Architecture

2.4 Peripheral Gateway

The 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 ACDs, 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, etc), 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 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 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 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.

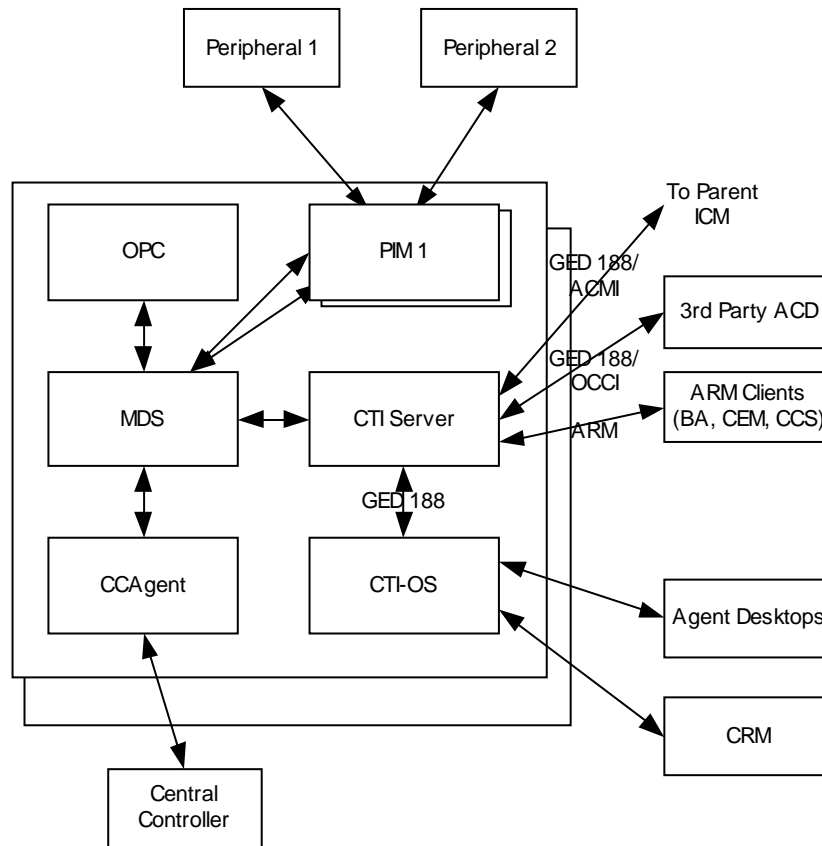


Figure 3: Peripheral Gateway Architecture

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:

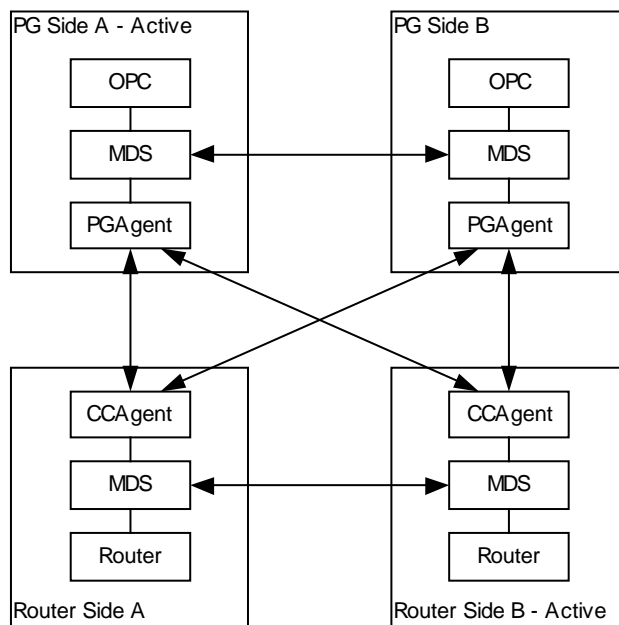


Figure 4: DMP Flows

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 PIM 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 CCE 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 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 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 is the connection from the PG to desktop clients and also 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 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 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, see *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's configuration database is authoritative.

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

2.5.1 Administration & Data Server

The Administration & Data Server is the main interface to the Unified ICM/CC configuration. On the Administration & Data Server 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's DBAgent process.

The two main clients in the Administration & Data Server 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 CCMP.

The Administration & Data Server 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 Administration & Data Servers. Administration & Data Servers 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.

Configuration data is supported on multiple Administration & Data Server types:

1. Administration Server and Real-time Data Server (AW Distributor) (with no HDS; configuration and real-time data but no historical or call detail data)
2. Administration Server, Real-time and Historical Data Server, and Detail Data Server (AW-HDS-DDS), (configuration, real-time, historical and call detail data)
3. Administration Server and Real-time and Historical Data Server (AW-HDS) (configuration, real-time and historical data but no call detail data)
4. Administration & Data Server Configuration (AW-CONFIG, configuration data only)

Configuration changes are NOT supported on the HDS-DDS type (which includes historical and call detail data but excludes real-time data); this type includes configuration data needed only for historical reporting purposes.

2.5.2 Configuration Updates

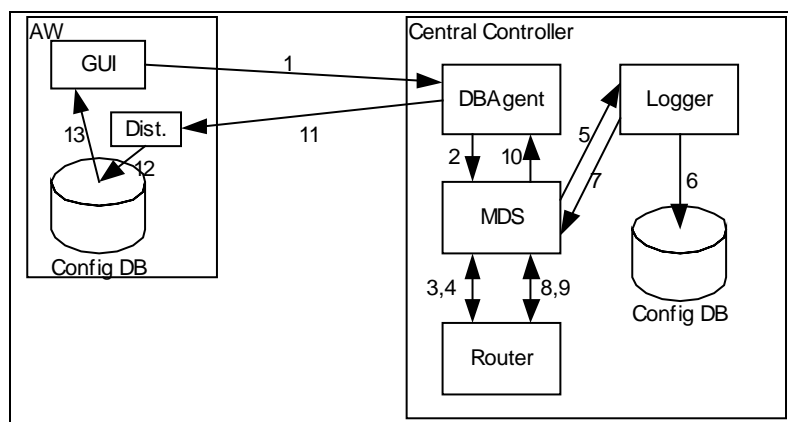


Figure 5: Configuration System Message Flow

The message flow for a configuration update is shown in Figure 5: Configuration System Message Flow. Figure 5 illustrates how a configuration update may happen in Unified CCE:

- The first step (not shown) is that an Administration Client reads configuration from the 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)
- 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, etc) (Step 9, 10).
- DBAgent sends this message to each of its Administration Server and Real-time Data Servers (Step 11). The Administration Server and Real-time Data Servers update their database (Step 12)
- The Configuration GUI sees the change happen (Step 13).

2.6 Reporting System

The reporting system for Unified ICM/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 Router.
- The Router sends these to the Logger, and then on to DBAgent processes which distribute them to Administration Server and Real-time Data Servers.
- 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.
- From here, a reporting component is used to pull the data from the databases. WebView and Cisco Unified Intelligence Suite (Unified IS) are web applications that uses Java Servlets to build reports to be viewed from thin (web browser) clients.

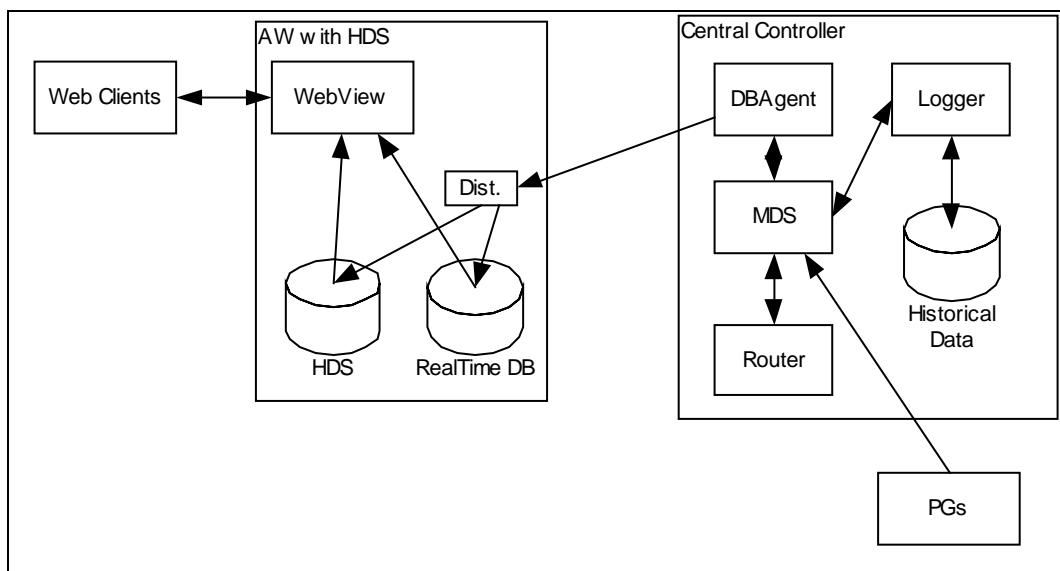


Figure 6: Reporting Architecture

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 HDS is an option to be installed with an Administration Server and Real-time Data Server . It uses the same distributor technology used to keep the 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.

There are three types of HDSs:

1. **Administration Server, Real-time and Historical Data Server, and Detail Data Server (AW-HDS-DDS):** HDS with call detail data store. This type includes both real-time and configuration data and may be used to source historical data for the Analysis Call Path tool and the CUIS Archiver. This type is intended for small- to medium-sized deployments. There may be a maximum of two AW-HDS-DDS servers per Logger side in small/medium deployments but only one per Logger side in a large deployment (presumably with multiple AW-HDS servers).
2. **Administration Server and Real-time and Historical Data Server (AW-HDS):** HDS without a call detail data store (no call detail, call variable, agent state data). This type also includes both real-time and configuration data but may not be used to source data for the Analysis Call Path tool or for the CUIS Archiver. This type is intended for large deployments. There may be a maximum of three AW-HDS per Logger side.
3. **HDS-DDS:** HDS with call detail data store but no real-time data or configuration data. This type may be used to source historical data for the Analysis Call Path tool and the CUIS Archiver. This type is intended for large deployments and to be used in conjunction with multiple AW-HDS servers. There may be a maximum of one HDS-DDS per Logger side (presumably with multiple AW-HDS servers).

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; New Atlanta’s ServletExec is what is currently used. The servlet engine connects to Sybase’s 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 Administration & Data Server. 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.

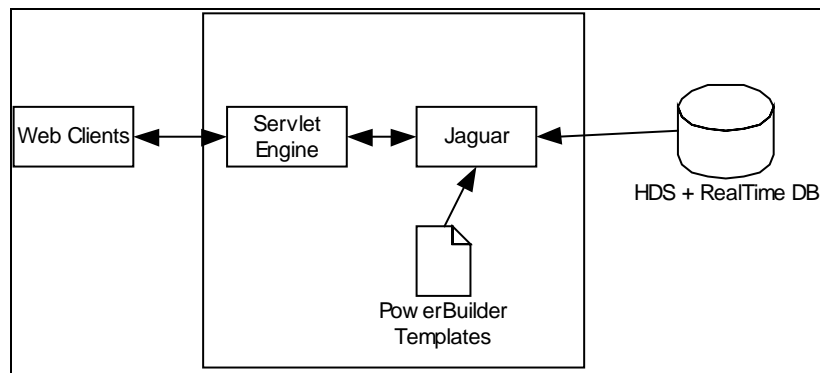


Figure 7: WebView Architecture

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 is supported by Unified ICME, Unified ICMH, 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 AW that houses the Administration & Data Server database (_awdb) and the Historical Data Server (_hds). The Administration & Data Server is the data source for real-time reports. The Historical Data Server is the data source for historical reports (AW-HDS-DDS and HDS-DDS types – see section 2.6.1 for more details on HDS types).

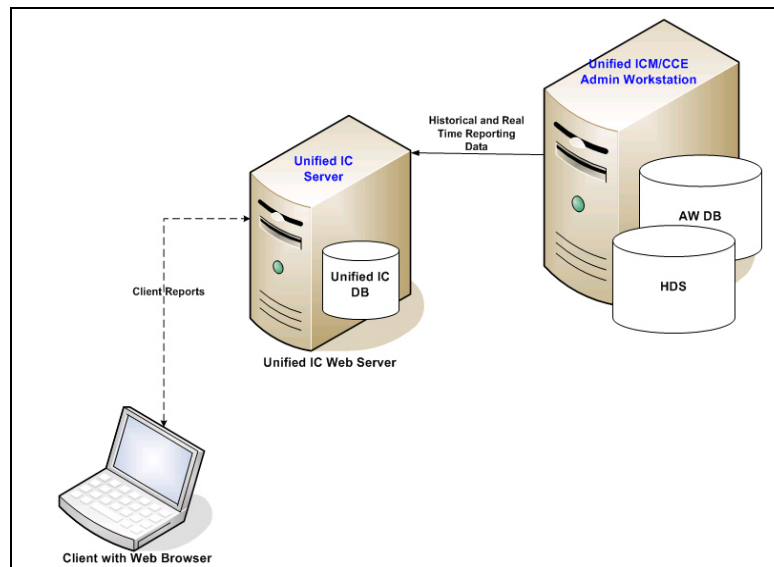


Figure 8: CUIS Simple Deployment

2.6.3.2 Unified IS “Standard” Deployment Model

In the standard deployment model, the Unified IC connects to the Unified ICM/CC Administration & Data Server 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 database on the Administration & Data Server. 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 Administration & Data Server 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.

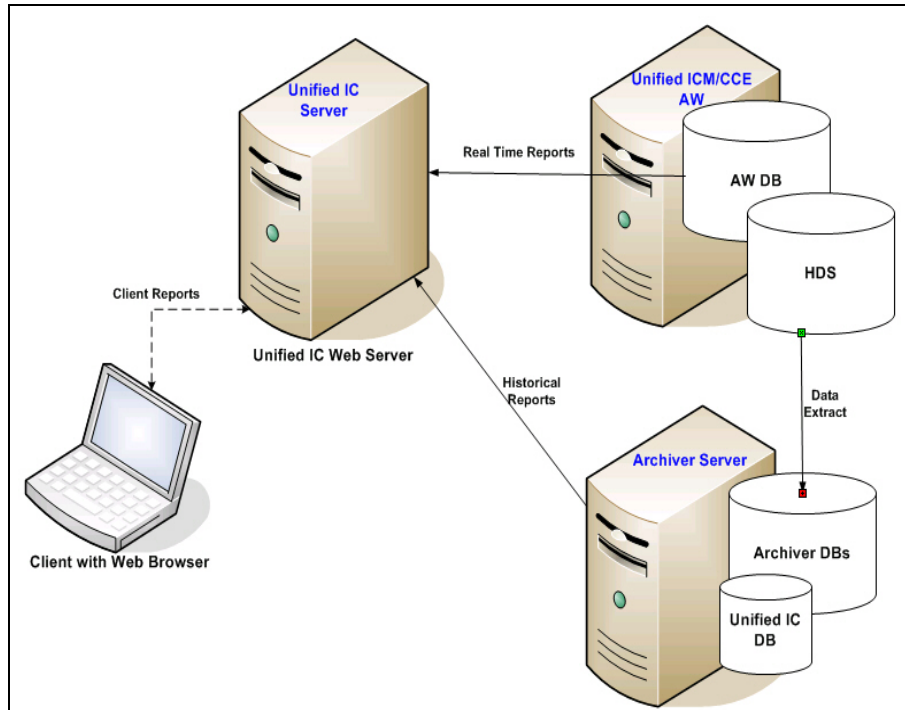


Figure 9: CUIS Standard Deployment

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.

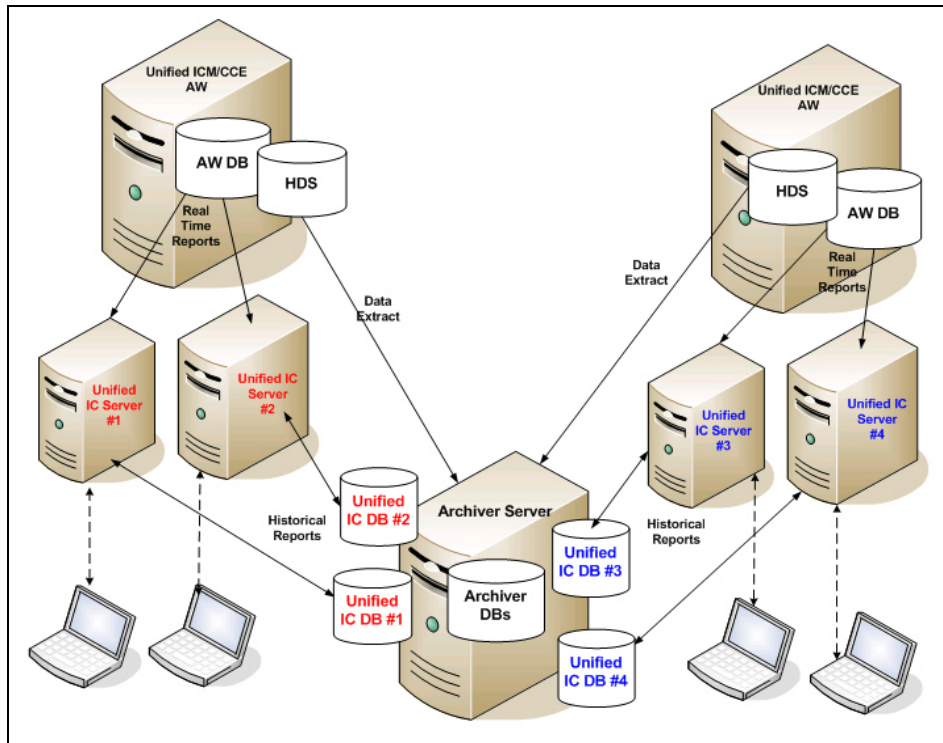


Figure 10: CUIS Scaled Deployment

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 *Datamart*) 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's 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.

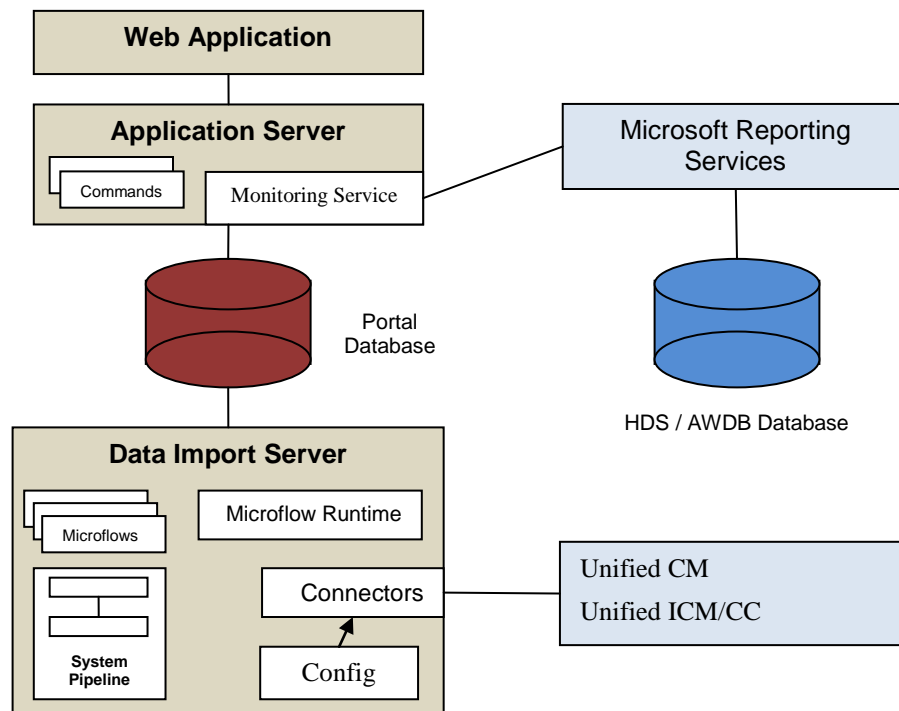


Figure 11: Unified CCMP Architecture

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, which defines 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.

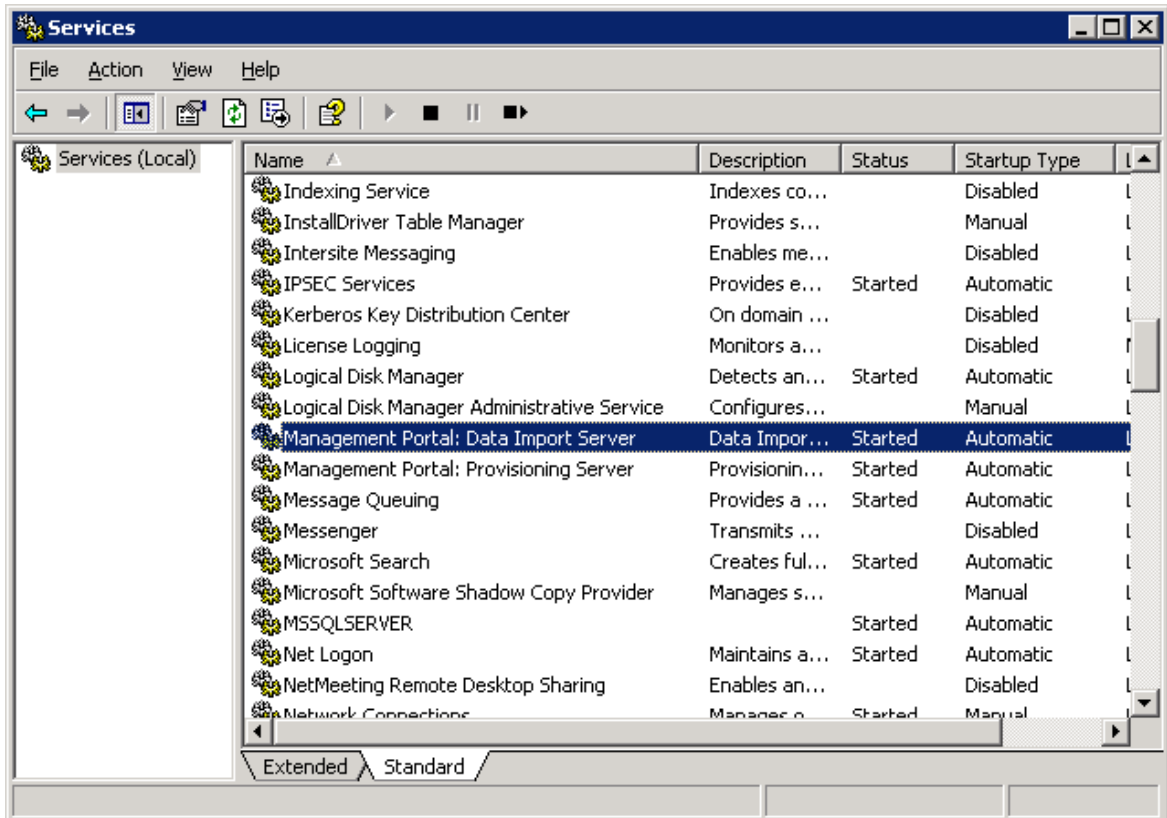


Figure 12: Unified CCMP Services

Unified CCMP exposes a rich set of performance (AKA “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’s 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. In a deploy

Unified CCE Release 8.0(1) offers the Session Initiation Protocol (SIP) Dialer alongside the Skinny Call Control Protocol (SCCP) Dialer that has been the sole Dialer offered in previous releases of Outbound Option. In an Outbound Option deployment that uses the SIP Dialer, functions such as dialing, call control, and Call Progress Analysis for Outbound campaigns are handled by the Voice Gateway, and not by Unified CM. This increases the number of Outbound agents that a deployment can service on a PG, and reduces the number of PGs and Dialers customers need to deploy for larger enterprise systems.

The following diagram provides a high level view of the Outbound Option components and their relationship with other Unified ICM components.

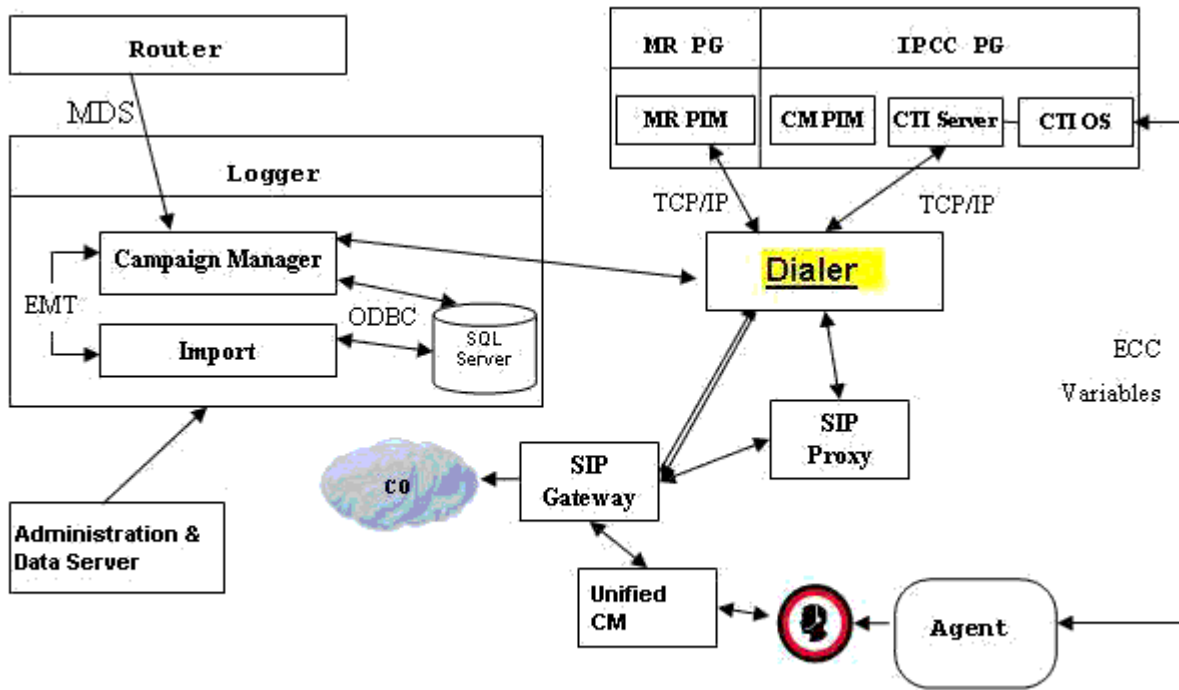


Figure 13: Outbound Option Component Relationships

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, e.g. 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.

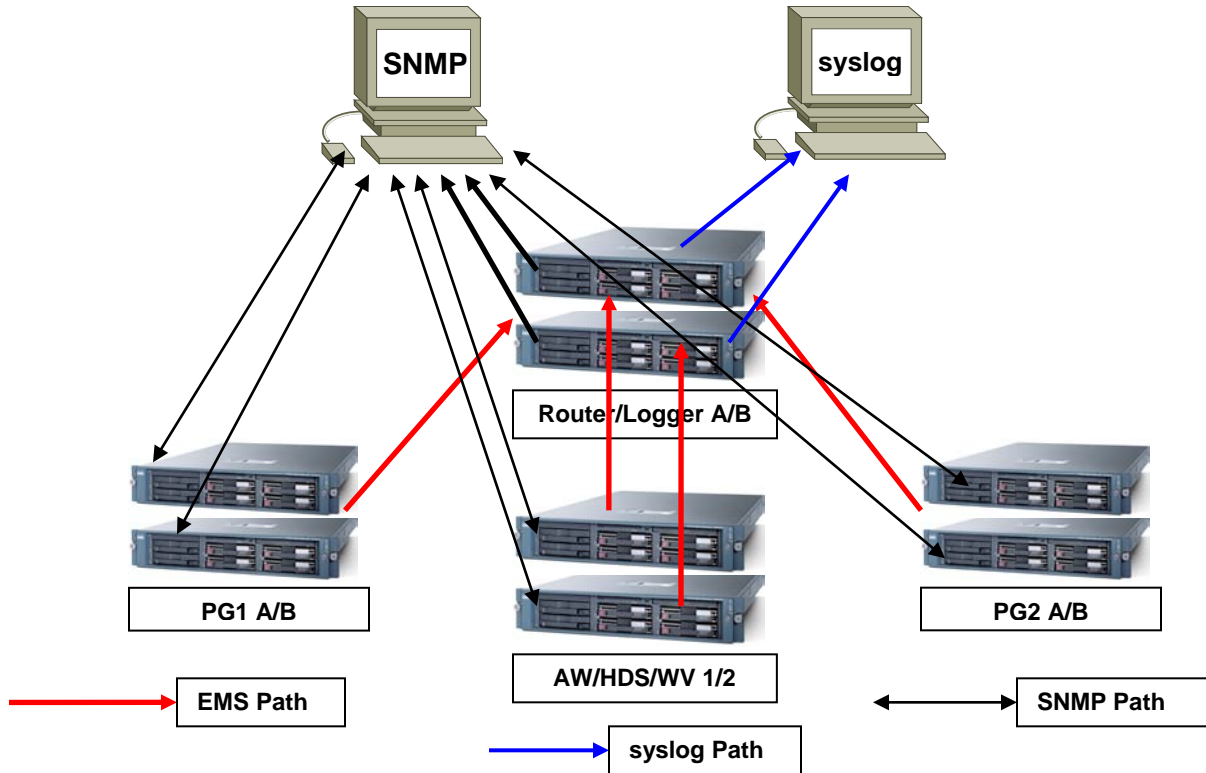


Figure 14: ICM/CC Event Message Flow

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 over 400 configured SNMP notifications for Unified ICM/CC.

3.1.2 Instrumentation

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

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

The servers may (optionally) expose platform MIBs appropriate for the vendor-originated server model; these MIBs and subagents are provided by the server vendor. If the provided subagent is a Microsoft Windows extension agent (designed to integrate with the Windows SNMP service), it 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 HP/IBM 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, IBM. This subagent provides instrumentation for low-level attributes of the specific hardware.

- 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

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 a Unified ICM/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 for cccaComponentElmtRunID.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 hrSWRunStatus object 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 in 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 ICM/CC because installation on Cisco MCS servers is not guaranteed (i.e. Unified ICM is supported on non-MCS hardware).

Note: The CDP driver may cause low-level system halts (e.g. “blue screens”) if installed on servers with an unsupported NIC chipset. This is the reason that the CDP driver and subagent is optionally installed for Unified ICM/CC.

3.2.2.4 MIB2

The MIB2 is defined in RFC 1213. It contains objects such as interfaces, ip, icmp, etc.

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

The Cisco Contact Center Applications MIB SNMP subagent provides access to component inventory, component status, performance metrics, and links to IETF standard host-based MIBs. Appendix A - section 11 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 AW/HDS.

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

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

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

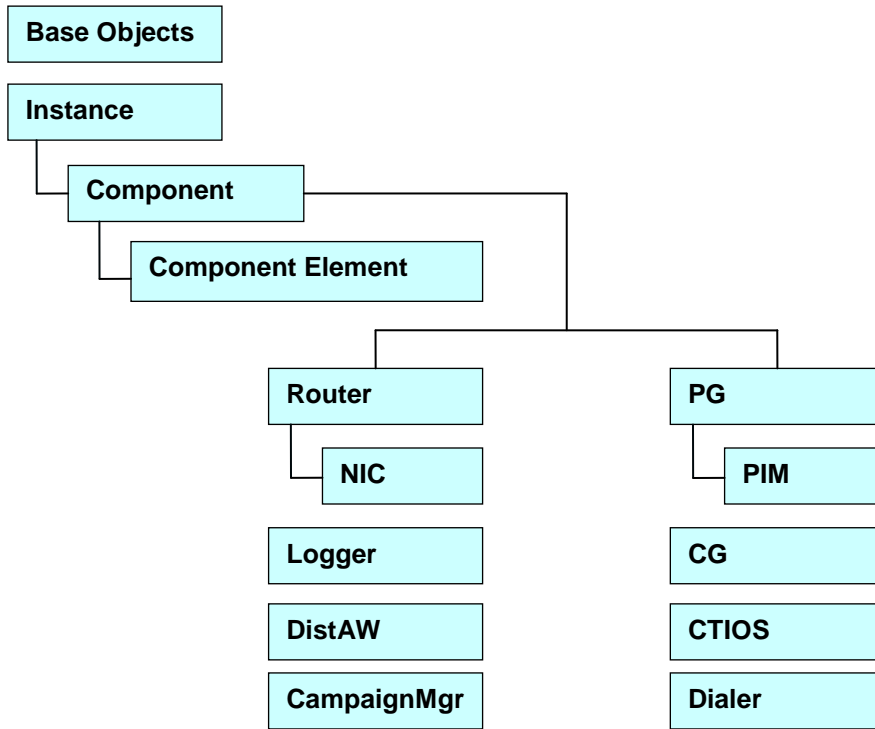


Figure 15: CISCO-CONTACT-CENTER-APPS-MIB Structure

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 which is arbitrarily assigned by the agent; the value of the Component number could change from one run period to another.

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

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

So, from an inventory standpoint (a network management station taking inventory of the server itself), the Network Management Station (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's been installed on this server and can see what is actually running. Let's say this is a Unified CCE central controller and the NMS wants to know what the inbound call rate is. With the Component entry for the Router, using the Component index of that entry, the NMS 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:

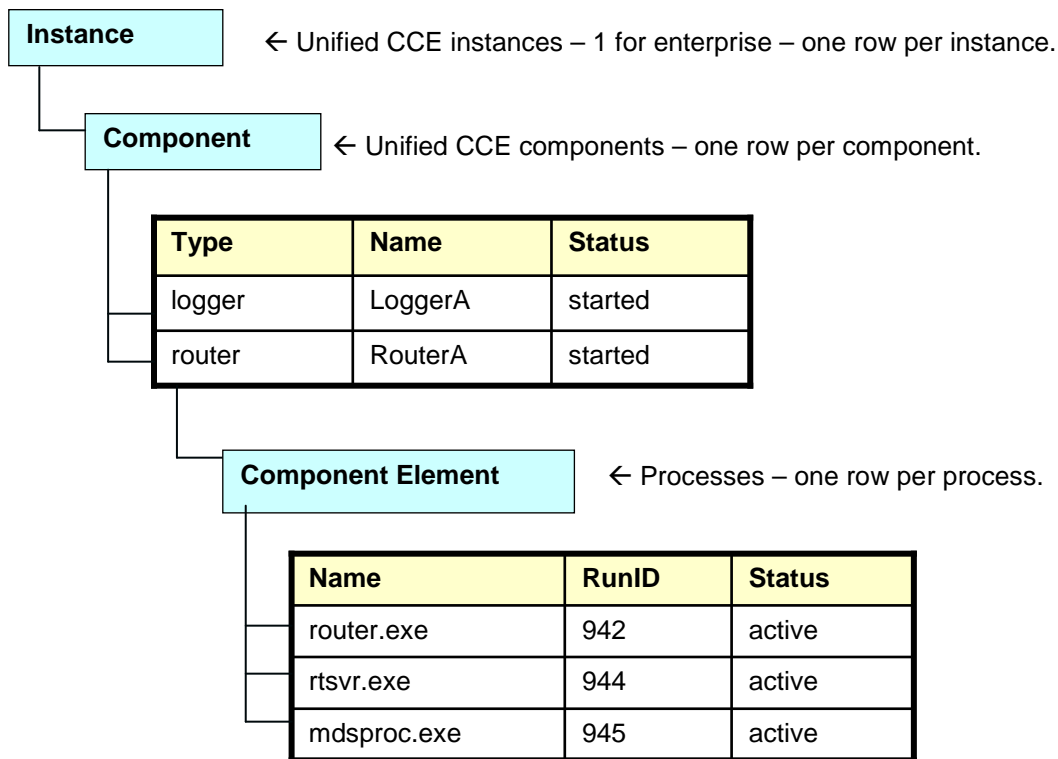


Figure 16: CCCA MIB – Component Inventory Example

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 components (e.g. Router, Logger).

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

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

The Component Element table lists all Unified CCE processes that should be executing, and exposes 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 CCA-MIB to Standard Host MIBs

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

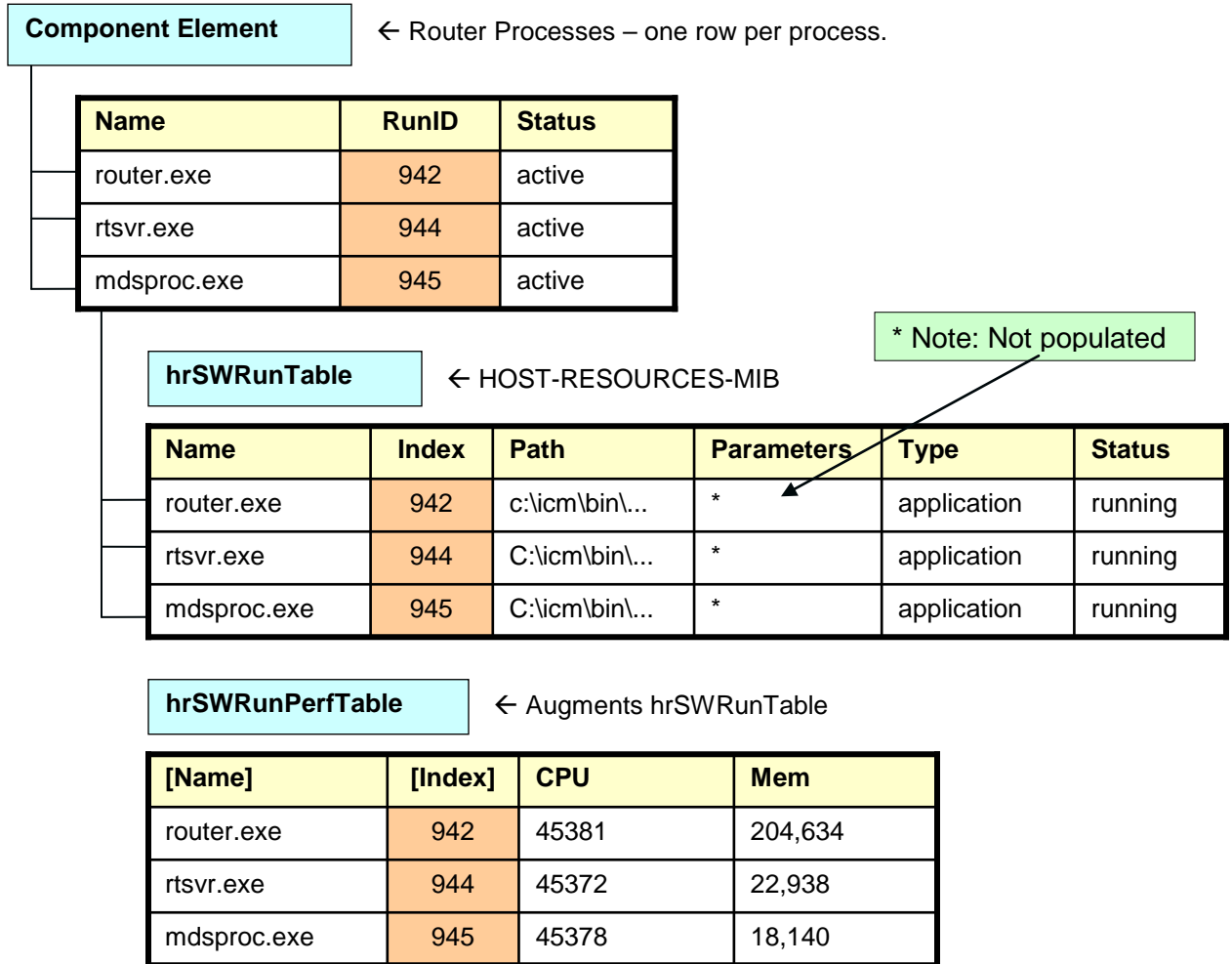


Figure 17: Mapping CCA MIB Objects to Host MIB Objects

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 cannot be implemented or simply are not necessary. There are no cut-and-dried rules. That some objects within these tables do not have values is not necessarily indicative of a failed implementation.

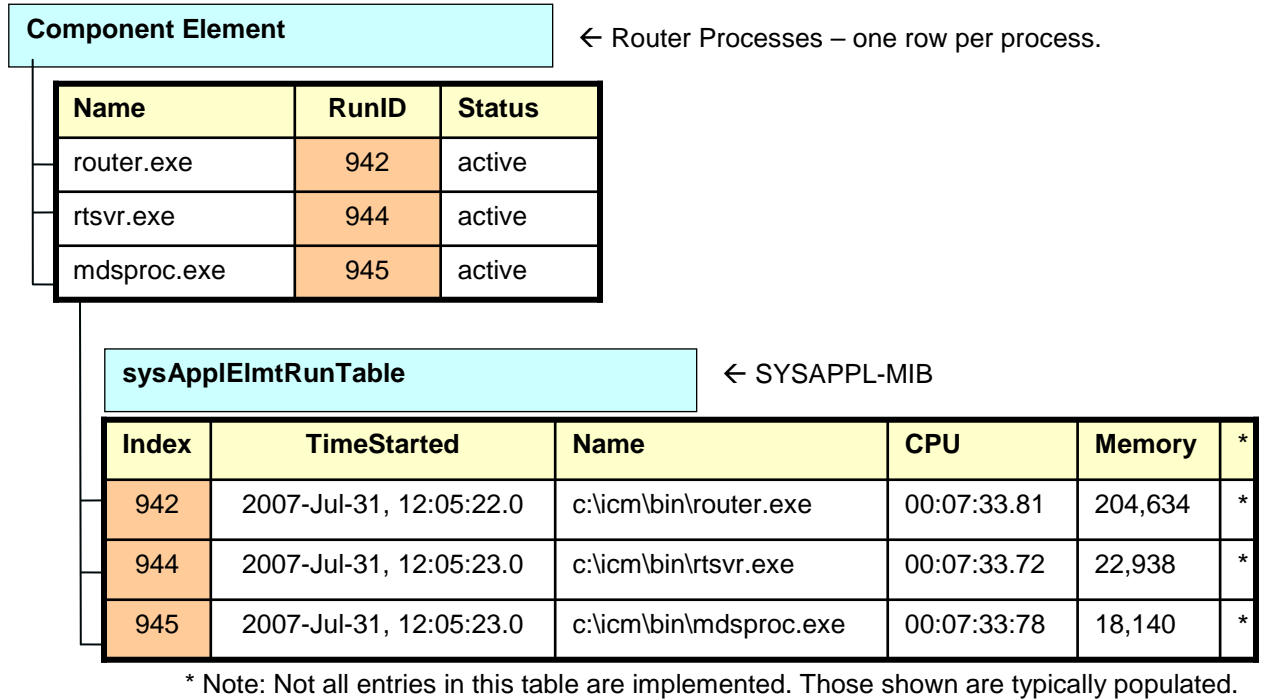


Figure 18: Mapping CCA MIB to SYSAPPL MIB

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 sysAppElmtRunTable.

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 (ccaRouterTable)
 - NIC Table (ccaNicTable) – since nearly always installed on the Router, this is considered a subcomponent of the Router
- Logger Table (ccaLoggerTable)
- Distributor Admin Workstation Table (ccaDistAwTable)
- Peripheral Gateway Table (ccaPgTable)
 - Peripheral Interface Manager Table (ccaPimTable) – since always installed on the PG, this is a subcomponent of the PG
- CTI Gateway Table (ccaCgTable)
- CTI Object Server Table (ccaCtiOsTable)
- Outbound Option Campaign Manager (ccaCampaignMgrTable)

- Outbound Option Dialer (cccaDialerTable)

A single notification object is defined in the MIB which is used to describe the format and content of all notifications generated by Unified ICM and Unified 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 Administration Server and Real-time Data Server host. This object offers a navigation point from the management station (assuming a web interface) can quickly access the Support Tools application server.
cccaWebSetupURL	The Web setup URL object holds the URL for the enterprise contact center application setup web service. The setup web service is a component of every Unified ICM and Unified CCE/CCH server and allows for an administrator to configure parameters of the contact center application as it relates to the installation of the product itself (not to be confused with provisioning).
cccaNotificationsEnabled	The notifications enabled object allows a management station

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

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 (e.g. 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. 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.

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

The component table is a list of enterprise contact center application functional components. A Unified CCE solution includes a collection of interconnected functional components (e.g. 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 'sysAppElmtRunIndex' value of 'sysAppElmtRunTable' of the SYSAPPL-MIB. This object value provides the

Object Name	Description
	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	<p>The last known status of a system process or service that is a necessary element of an enterprise contact center application functional component.</p> <p>unknown(1): The status of the component element cannot be determined.</p> <p>disabled(2): The component element has been explicitly disabled by an administrator.</p> <p>stopped(3): The component element is stopped; it may be dysfunctional or impaired.</p> <p>started(4): The component element has been started.</p> <p>active(5): The component element is currently running.</p>

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 'sysAppElmtRunTable' 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: CCA 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.

Object Name	Description
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 (i.e. the other side) server of an enterprise contact center application fault tolerant router component. If this component is not part of a duplex pair (i.e. 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.
cccaRouterCallsInQueue	The router calls in queue object indicates the total number of calls queued in all network Voice Response Units (VRUs), from the router's perspective, including those calls that are in the process of transferring to the VRU for queuing.
cccaRouterAppGwEnabled	The router application gateway enabled object indicates whether an application gateway is configured and a part of this contact center application deployment. An application gateway provides an external interface to business back-end systems that may be used as external input to call scripting logic, or, that logic which controls how a customer call is handled (routed).
cccaRouterDBWorkerEnabled	The router database worker enabled object indicates whether a database worker process has been configured and is a part of this contact center application deployment. A database worker provides an interface to an external database from which data may be retrieved and used as input to call scripting logic, or, that logic which controls how a customer call is handled (routed).
cccaRouterPGsEnabledCount	The router PGs enabled count object holds the number of PGs that have been enabled for this router; during normal operation, this is the number of PGs that will connect to this router functional component. There is an imposed architectural limit of 150 peripheral gateways per deployment.
cccaRouterPublicHighAddr	The router public high address object holds the address of the local high-priority interface of this router functional component to the public network. The 'public' network interface is exposed outside the realm of the Unified ICM or Unified CC application and is used for the transfer of data between this router and other functional components of the contact center deployment. This interface is reserved for high-priority messages; network prioritization is typically configured for this interface to ensure a level of quality of service.
cccaRouterPublicNonHighAddr	The router public non-high address object holds the address

Object Name	Description
	of the local interface of this router functional component to the public network that is used for best effort priority messages. The 'public' network interface is exposed outside the realm of the Unified ICM or Unified CC application and is used for the transfer of data between this router and other functional components of the deployment. This interface is used for normal-priority messages.
cccaRouterPrivateHighAddr	The router private high address object holds the address of the local high-priority interface of this router functional component to the private network. The 'private' network interface is used exclusively by the Unified ICM or Unified CC application for the transfer of synchronization data between duplexed pairs and for the transfer of application data from the router to the logger. This interface is reserved for high-priority messages and as much as 90% of the available network bandwidth is allocated to this interface.
cccaRouterPrivateNonHighAddr	The router private non-high address object holds the address of the local interface of this router functional component to the private network that is used for best effort priority messages. The 'private' network is used exclusively by the Unified ICM or Unified CC application for the transfer of synchronization data between duplexed pairs and for the transfer of application data from the router to the logger. This interface is used for normal-priority messages."

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

The Router table has a sparse dependent relationship with the component table. The instance number acts as the primary index for the Router table 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: CCA 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 (i.e. the other side) server of an enterprise contact center application fault tolerant logger component. If this component is not part of a duplex pair (i.e. simplex), the object value will be the null string. The logger connects to its duplex pair via a 'private' interface -- a closed subnet that guarantees a quality of service level that 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 Administration Server, Real-time and Historical Data Server, and Detail Data Server. If 'true', the logger feeds historical data at regular intervals to the HDS for long-term storage. In this configuration, administrator reports are generated by accessing data from the HDS rather than the logger in order to remove the performance impact of reporting on the logger.
cccaLoggerAvgDBWriteTime	The logger average database write time expresses the average amount of time, in 100 nanosecond units, required to write data to a table in the central controller database. This value represents the average time per write of the write operations that occurred in the past second. This object is a good indicator of contention for database access.

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 Administration Server and Real-time Data Server 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 Administration Server and Real-time Data Server side value is either 'A' or 'B'. For simplex configurations, the Administration Server and Real-time Data Server side value defaults to 'A'.
cccaDistAwType	Which type of enterprise contact center application distributor administrator workstation, is installed on this server. The Administration Server and Real-time Data Server type varies based on the configuration of the contact center solution.
cccaDistAwAdminSiteName	A user-defined textual name that uniquely identifies the location or the configuration of the Administration Server and Real-time Data Server component.
cccaDistAwRouterSideAName	The host name of the side 'A' router that this enterprise contact center application Administration Server and Real-time Data Server functional component is associated. The the Administration Server and Real-time Data Server component must be connected to a router that is part of the same instance. If the side B router is the active router and a failure occurs, the side A router then immediately assumes the role. In this case, the Administration Server and Real-time Data Server 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 Administration Server and Real-time Data Server functional component is associated. The Administration Server and Real-time Data Server component must be connected to a router that is part of the same instance. If the side A router is the active router and a failure occurs, the side B router then immediately assumes the role. In this case, the Administration Server and Real-time Data Server will lose its connection to the side A router and thus use this object value to connect to the side B router.

Object Name	Description
cccaDistAwLoggerSideAName	The host name of the side 'A' logger that this enterprise contact center application Administration Server and Real-time Data Server functional component is associated. The Administration Server and Real-time Data Server component must be connected to a logger that is part of the same instance. If the side B logger is the active logger and a failure occurs, the side A logger then immediately assumes the role. In this case, the Administration Server and Real-time Data Server 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 Administration Server and Real-time Data Server functional component is associated. The Administration Server and Real-time Data Server component must be connected to a logger that is part of the same instance. If the side A logger is the active logger and a failure occurs, the side B logger then immediately assumes the role. In this case, the distributor AW 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 (i.e. the other side) server of an enterprise contact center application fault tolerant Administration Server and Real-time Data Server component. If this component is not part of a duplex pair (i.e. 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 Administration Server and Real-time Data Server 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 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.

Object Name	Description
cccaDistAwWebReskillingURL	The administration and data server web re-skilling URL object holds the URL for the contact center application web re-skilling tool. The web re-skilling tool allows contact center administrators and supervisors to re-skill agents (reassign contact center agents to different skill groups allowing them to take calls of a different topic).

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

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

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.

Object Name	Description
cccaPgDuplexPairName	The host name of the duplex pair (i.e. 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 (i.e. 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.
cccaPgCallsInProgress	The call in progress object shows the number of calls that are currently active and being managed/monitored by this peripheral gateway.
cccaPgAgentsLoggedIn	The agents logged on object shows the number of agents associated with this peripheral gateway that are currently logged on and are being managed/monitored by this peripheral gateway.
cccaPgAgentsReady	The agents ready object shows the number of agents associated with this peripheral gateway that are currently logged on and in a 'Ready' state, i.e., ready to receive calls.
cccaPgAgentsTalking	The agents talking object shows the number of agents associated with this peripheral gateway that are currently logged on and taking a call (in a 'Talking' state).
cccaPgID	The PG identifier is a unique numeric identifier for this enterprise contact center application peripheral gateway. The identifier is assigned by the contact center application.

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

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

Table 3-10: CCA 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, e.g. 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 PG and to the appropriate instance.

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

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' 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 Name	Description
	object value to connect to the side 'B' PG.
cccaCgDuplexPairName	The host name of the duplex pair (i.e. 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 (i.e. simplex), the object value will be the null string.
cccaCgOpenSessions	The CG open sessions object indicates the number of sessions (connections) that have been established between the CTI Gateway and CTI clients. These are active sessions that are functioning normally.
cccaCgOtherSessions	The CG other sessions objects indicates the total number of sessions (connections) between the CTI Gateway and CTI clients that are not normal, open/active sessions. This includes sessions that are 'opening' (not yet established and initialized), session that are 'closing' (connections being torn down) as well as sessions that are in an 'unknown' state and sessions that have failed. While this object value will fluctuate from time to time, during normal operation, it will stabilize. A steadily increasing value indicates a problem that should be investigated.
cccaCgID	The CG number is a unique numeric identifier for this enterprise contact center application CTI gateway. The identifier is assigned by the contact center application.

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

Object Name	Description
	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.
cccaCtiOsActiveClients	The active clients object holds the number of CTI OS active client mode desktop connections. This value indicates the total number of desktops connected to the CTIOS server. The number of desktops connected to the A and B side of CTIOS determine the total desktops connected through this instance of CTI OS server.
cccaCtiOsActiveMonitors	The active monitors object holds the number of CTI OS active monitor mode desktop connections. CTIOS only supports two monitor mode connections per each CTI OS server. This value indicates how many monitor mode connections are in use. Once there are two in use further monitor mode connection attempts are rejected.
cccaCtiOsCallsInProgress	The calls in progress object indicate the total number of active calls being tracked by CTI OS. This value shows how many calls are currently being handled by CTI OS. This value should go up and down based on the call arrival rate and the agent call completion rate.
cccaCtiOsCallsFailed	The calls failed object holds the total number of calls that failed via a failure event being reported to CTI OS. If this count begins to rise, the log file should be captured to gather more specific information on the failure events.

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

The CTI OS table has a sparse dependent relationship with the component table. The instance number acts as the primary index for the CTI OS table in order to properly relate a CTI OS 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 3-13: CCCA MIB Outbound Option Campaign Manager Table Objects

Object Name	Description
cccaCampaignMgrDbUtilization	The campaign manager and Import processes share a private database on the Side A Logger. The campaign manager database utilization object shows what percentage of allocated space in the database is currently utilized. An administrator should start paying attention when this value exceeds 80 percent.
cccaCampaignMgrQueueDepth	The campaign manager is a multithreaded process. There is one main dispatch thread that is involved in most processing. The queue depth object indicates how many messages are queued to this internal dispatch thread. By default, the campaign manager will deliberately restart when this value exceeds 10,000 messages in queue as a self-defense mechanism; the administrator must then investigate the reason for this performance bottleneck.
cccaCampaignMgrAvgQueueTime	The campaign manager is a multithreaded process; however, there is one main dispatch thread that is involved in most message processing. The average queue time object shows the average amount of time a message spends in the main dispatch thread queue awaiting processing (in milliseconds).
cccaCampaignMgrActiveDialers	The campaign manager process feeds several dialer components which manage the dialing of customers for outbound campaigns. The active dialers counter indicates how many dialers are currently registered to this campaign manager.

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

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

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

The SNMP agent constructs the Campaign Manager table at startup. Since Campaign Manager components can only be configured while the enterprise contact center application is stopped, Campaign Manager table entries cannot be added to or deleted from the table either by the agent or the management station when the application is running. The agent will update the values of

Campaign Manager entry objects as their values change when the application is running. All objects in this table are read-only to the management station.

Each Campaign Manager entry represents an enterprise contact center application Campaign Manager server functional component configured on the server. The Campaign Manager component, which resides on the ICM/CC Logger (side A), is responsible for:

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

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

Table 3-14: CCA MIB Outbound Option Dialer Table Objects

Object Name	Description
cccaDialerCampaignMgrName	The dialer campaign manager name object holds the host name or IP address of the Outbound Option Campaign Manager to which this dialer is associated. The dialer connects to the campaign manager to exchange data related to an outbound dialing campaign.
cccaDialerCampaignMgrStatus	The dialer campaign manager status indicates the current connection status between this dialer and the Outbound Option Campaign Manager component which is co-resident with the logger (side A).
cccaDialerCtiServerAName	The dialer CTI server A name object holds the host name or IP address of the contact center application CTI Server side A functional component which this dialer is dependent upon. The dialer connects to the CTI Server to monitor skill group statistics (to choose an agent) and execute call control once an available agent is selected.
cccaDialerCtiServerBName	The dialer CTI server B name object holds the host name or IP address of the contact center application CTI Server side B functional component which this dialer is dependent upon. The dialer connects to the CTI Server to monitor skill group statistics (to choose an agent) and execute call control once an available agent is selected.
cccaDialerCtiServerStatus	The dialer CTI server status indicates the current connection status between this dialer and the active CTI server component.
cccaDialerMediaRouterStatus	The dialer media router status indicates the current connection status between this dialer and the Media Routing (MR) Peripheral Interface Manager (PIM) component. The dialer uses the MR PIM interface to

Object Name	Description
	reserve an available agent as a recipient for a dialed customer call.
cccaDialerQueueDepth	The dialer is a multithreaded process which communicates between threads using inter-thread messaging. The queue depth object indicates how many messages are currently queued for the main dispatch thread. When this object is used in combination with the average queue time object, message processing performance can be gauged. By default, the Dialer process will deliberately restart when this value exceeds 10,000 messages.
cccaDialerAvgQueueTime	The dialer is a multithreaded process that communicates between threads using messaging. There is one main dispatch thread that is involved in most message processing. The average queue time shows the average amount of time (in milliseconds) that a message spent in the queue before being de-queued processing. When this object used in combination with the queue depth object, message processing performance can be gauged.
cccaDialerTalkingAgents	For an agent campaign, the dialer places calls to customers and transfers those customer calls to agents. The talking agents object indicates how many agents are currently talking in the monitored campaign skill group.
cccaDialerCallAttemptsPerSec	The call attempts per second object tracks how many calls the dialer is placing per second, rounded to the nearest integer. If the dialing rate is too high, it can result in network congestion on the voice network which can result in inefficient dialing.
cccaDialerConfiguredPorts	The dialer configured ports object is a count of the total number of ports that have been configured for placing calls to customers and for transferring calls to agents during outbound calling campaigns. During normal operation, the dialer configured ports object value is equal to a sum of busy and idle ports.
cccaDialerBusyCustomerPorts	The dialer busy customer ports object is a count of the number of ports currently in use for customer calls. The port is the unit on the Dialer that places calls to reserve agents and to contact customers.
cccaDialerBusyReservationPorts	The dialer busy reservation ports object tracks how many ports are currently busy reserving agents. The port is the unit on the Dialer that places calls to reserve agents and to contact customers.
cccaDialerIdlePorts	The dialer idle ports object is a count of the number of ports that are currently idle, i.e. there are no calls to customers or to agents using these ports and they are available to the dialer for placing new calls.
cccaDialerBlockedPorts	The dialer blocked ports object is a count of the number of

Object Name	Description
	ports that are currently unusable for placing calls. A blocked port may be an impaired or inoperable port or one that has a 'stuck' call that has not been dropped. A 'stuck' call is a call that has been identified by the application as exceeding a duration threshold.

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

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

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

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

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

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

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 (e.g. 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**.

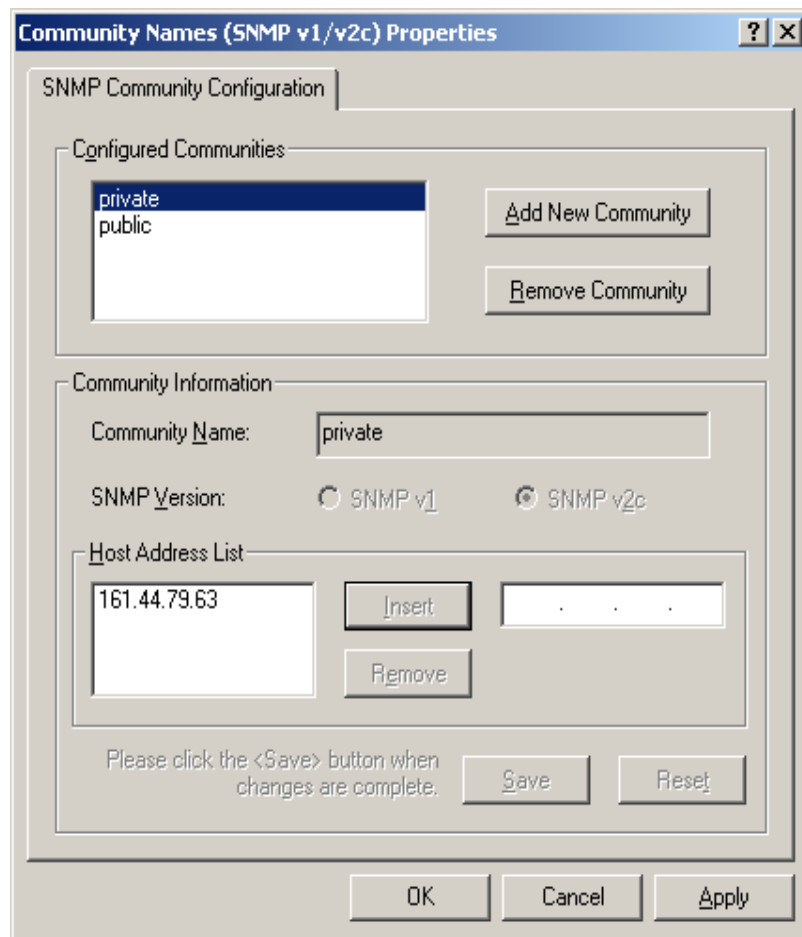


Figure 19: SNMP Community Name Configuration Dialog

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**.

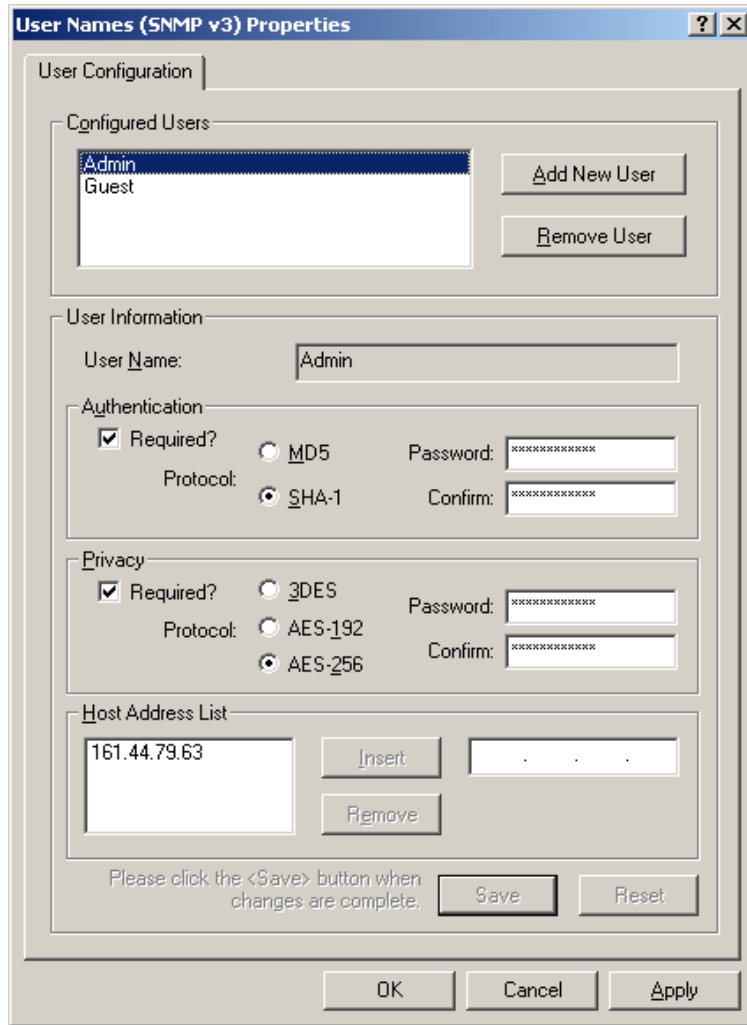


Figure 20: SNMP User Name Configuration Dialog

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 3-15: 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 (e.g. wrong community name), the device will generate a failed authentication trap.

The notifications are explained in <INSTALL_DRIVE>/icm/snmp/CCA-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.

Definition of Agent Performance Settings:

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 or the Analysis Manager.

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

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**.

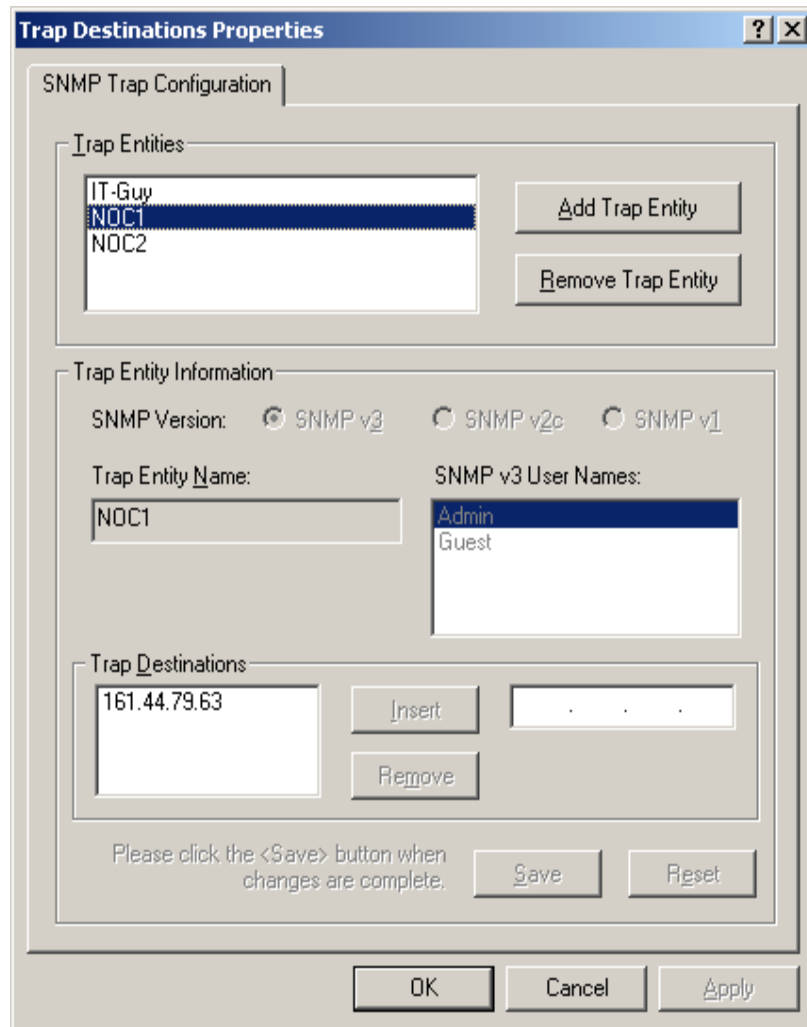


Figure 22: SNMP Trap Destination Configuration Dialog

4 Understanding Unified 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 Unified ICM/CC Notification Type

cccalcmEvent

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: <i>'clear'</i> (0): The clear state indicates that the condition which generated a previous raise notification has been resolved. <i>'applicationError'</i> (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. <i>'raise'</i> (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. <i>'singleStateRaise'</i> (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

Object Name	Description
	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	<p>The type of enterprise contact center application functional component or subcomponent that generated this notification. The node types are:</p> <p>'unknown' (0): The notification originates from an unknown source.</p> <p>'router' (1): The notification was generated by the router functional component.</p> <p>'pg' (2): The notification was generated by the peripheral gateway functional component.</p> <p>'nic' (3): The notification was generated by the network interface controller functional component.</p> <p>'aw' (4): The notification was generated by the administrator workstation functional component.</p> <p>'logger' (5): The notification was generated by the logger functional component.</p> <p>'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).</p> <p>'cg' (7): The notification was generated by the CTI gateway functional component.</p> <p>'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.</p>
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.

Object Name	Description
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 this notification. The severity levels are: <i>'informational'</i> (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. <i>'warning'</i> (2): The notification contains serious health or operational state information that could be a precursor to system impairment or eventual failure. <i>'error'</i> (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's `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: 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

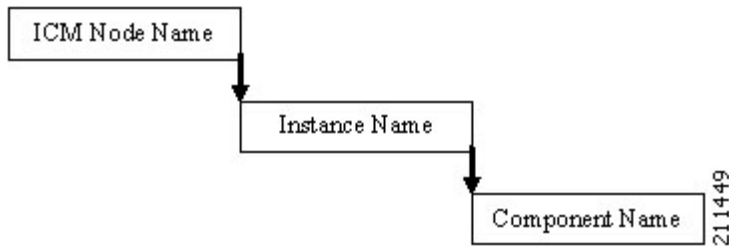
Field	Value / Description
NOTIFICATION	105023C
cccaEventMessageId	3775201852 (0xE105023C)
DESCRIPTION	The router has detected that it is no longer synchronized with its partner. One result of this is that the router might be routing some calls incorrectly.
cccaEventState	Single-state Raise
SUBSTITUTION 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. 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

Field	Value / Description
	the icm\ <instance>\rb b.="" center.<="" contact="" directory="" of="" router="" support="" td="" the=""> </instance>\rb>
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)
```



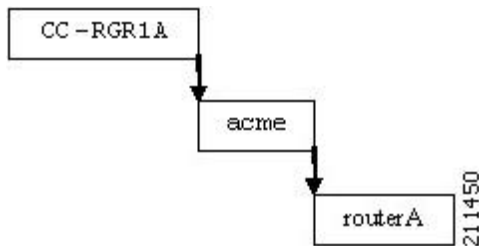
where:

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's 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.

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.

Field	Description
MONTH	Current month represented in MMM format (e.g. “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: PRODUCT_COMPONENT_SUBCOMPONENT PRODUCT – such as ICM COMPONENT – such as Router SUBCOMPONENT – such as CallRouter
SEVERITY	Supported severity values are: 3 (Error) 4 (Warning) 6 (Informational) 7 (Debug)
MSGID	Hexadecimal message id that uniquely identifies the message, such as 10500FF.
TAGS	(Optional) Supported tags are: [comp=%s] - component name including side, such as Router-A [pname=%s] - process name, such as rtr [iid=%s] - instance name, such as ipcc1 [mid=%d] - message id, such as 10500FF [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**.

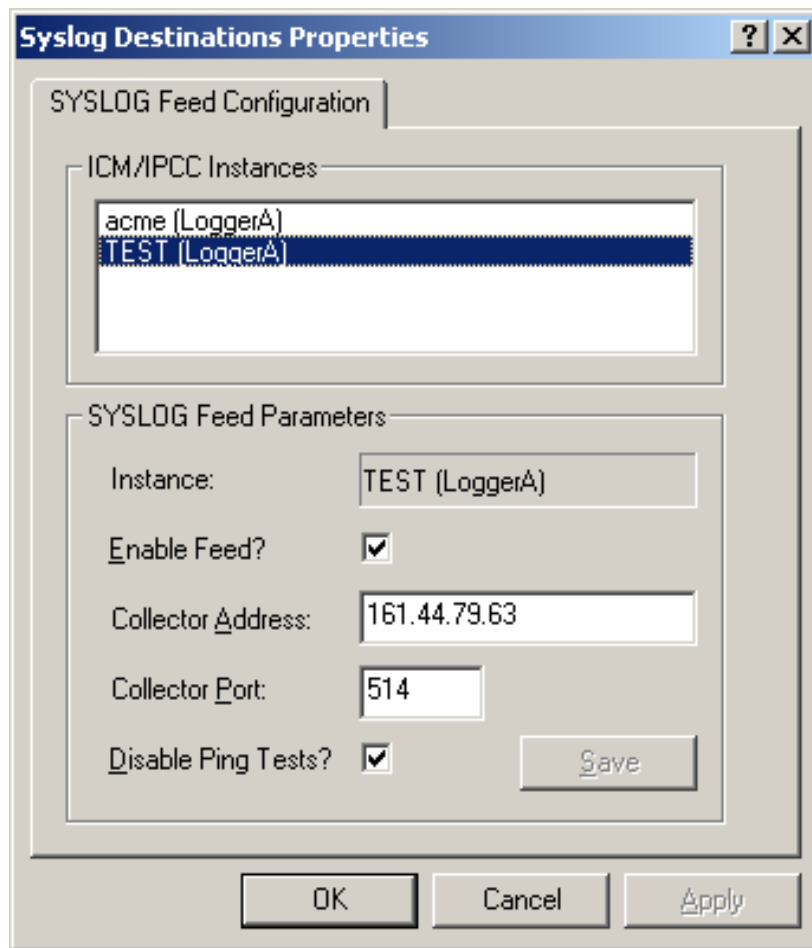


Figure 23: syslog Feed Configuration Dialog

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*.

6.1.1 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 Administration & Data Server
	* 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

Component	Process	Description
		for the Router to query external databases.
	* [NIC].exe	<p>[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.</p> <p>The presence of a NIC process in a CCE deployment is <u>very rare</u>.</p> <p>NIC process names: atnic.exe, cainnic.exe, netwrkcic.exe, crspnic.exe, cwcnic.exe, gktmpnic.exe, incrpnice.exe, mcinic.exe, gennic.exe, ntnic.exe, ntlmic.exe, sprnic.exe, ss7innic.exe, stentornic.exe, timnic.exe, unisourcenic.exe</p>
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 Administration & Data Server.
	* csfs.exe	[Critical]: The alarm/event processor. CSFS distributes alarms/events send via EMS to supported alarm/event feeds, e.g. SNMP, syslog. CSFS stands for Customer Support Forwarding Service, which in Unified ICM's infancy, forwarded events to a central monitoring location.
	* cw2kfeed.exe	<p>[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.</p> <p>If a syslog collector is not configured, this process will not be executing.</p>
	* campaignmanager.exe	<p>[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.</p> <p>This process is installed and executes on the "A" side Logger only.</p>
	* baimport.exe	[Optional/Critical]: Outbound Option Import process. This process imports contact lists into the Outbound Option

Component	Process	Description
		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. 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
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. This is <u>very</u> common in CCE deployments but may not be present on all PGs. There may be multiple instances of this process running.
	* acmpim.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). 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). 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

Component	Process	Description
		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.
Administration & Data Server (AW/HDS)	* configlogger.exe	[Critical]: Processes inbound configuration data.
	* updateaw.exe	[Critical]: Updates the local configuration database with configuration data from the central controller.

Component	Process	Description
	* rtclient.exe	[Critical]: Receives a real-time data feed (from a real-time distributor) and updates the local 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. 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. 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

Component	Process	Description
	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

6.2 Using the Local Desktop

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

6.3 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's Node Manager service is identified as **Cisco ICM acme RouterA**.

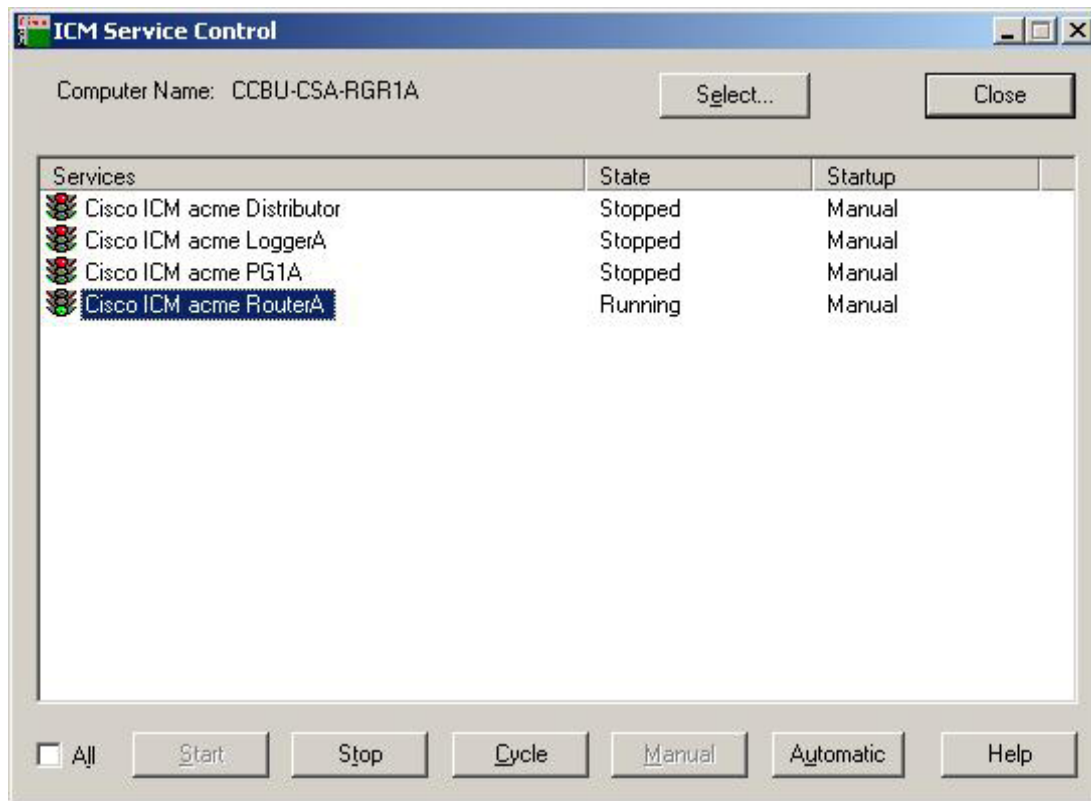


Figure 24: ICM Service Control

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.

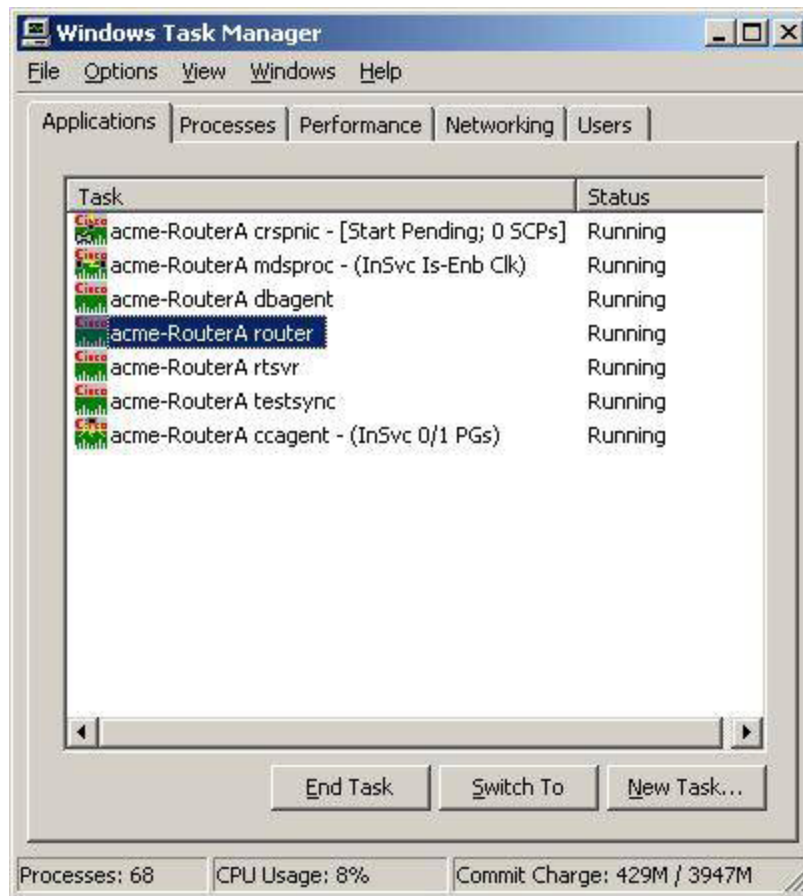


Figure 25: Windows Task Manager – Applications List

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.

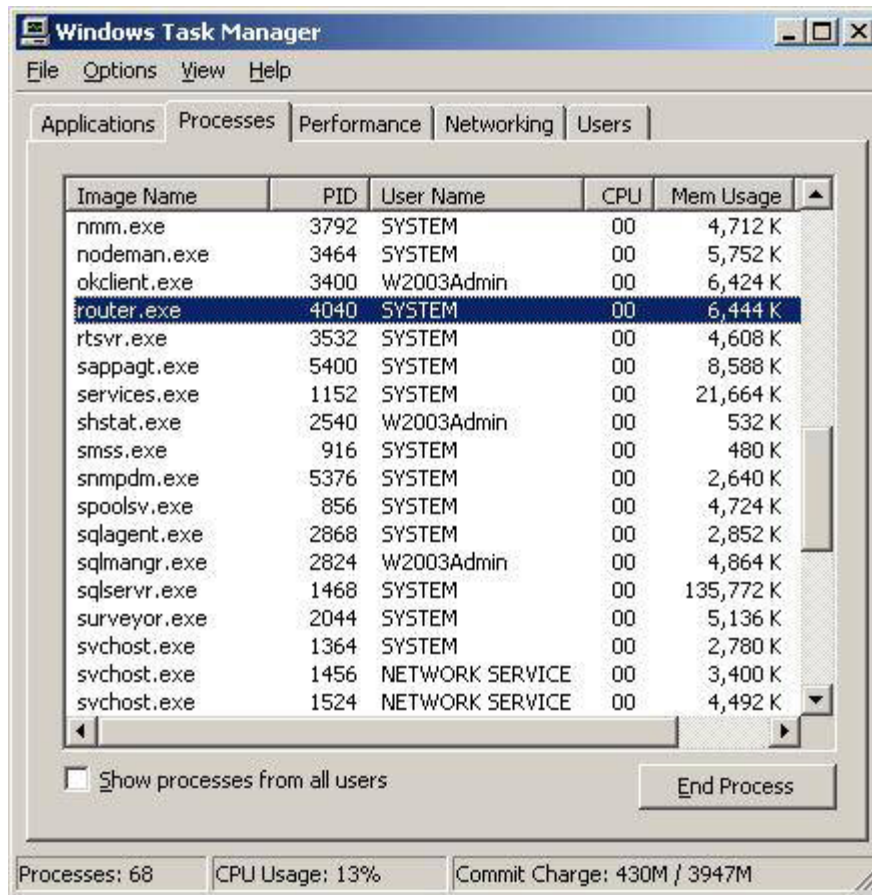


Figure 26: Windows Task Manager - Process List

6.4 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. 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.

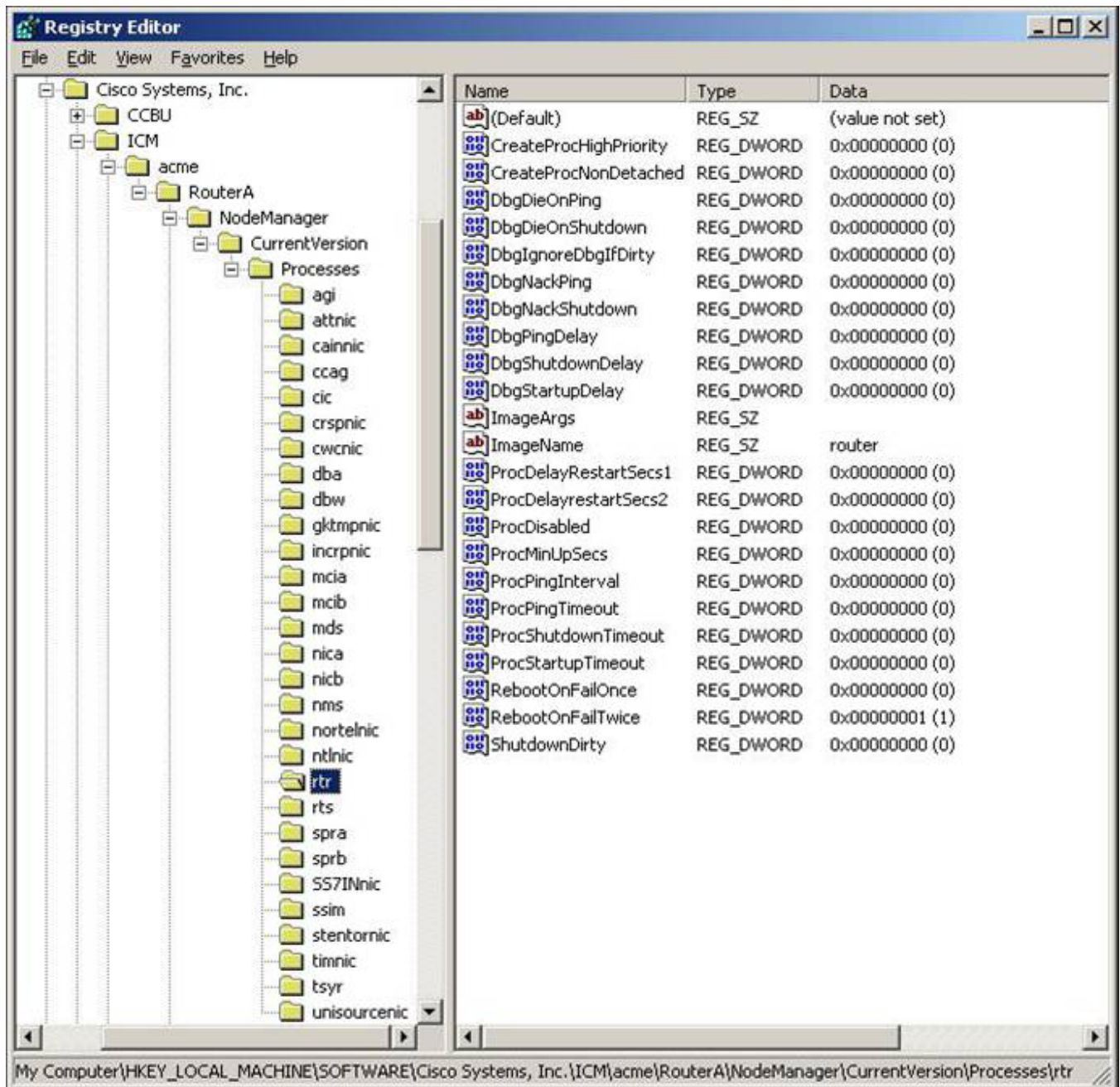


Figure 27: Registry Editor

6.5 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 *ccaInstanceTable*, *ccaComponentTable*, and *ccaComponentElmtTable*. 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

With serviceability enhancement, 8.0 utility tools provide centralized control for setting up each component trace level. You can manually modify it from the registry key settings, too.

Users can either use the tool or manually modify the registry key value.

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 levels 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 (e.g. agent capacity, IVR port capacity, 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 (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. 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 Trace Levels Configurations

Following are the four trace levels that are identified for the Unified Communication solution:

- Default (0): Product install default, should have no/minimal performance impact
- Warning (1): Log detailed (plus default level) trace messages, small performance impact
- Error (2): Log detailed (plus warning/default level) trace messages, medium performance impact.
- Debug (3): Log most detailed (plus error/warning/default level) trace messages, high performance impact.

If the trace level does not match any of the above defined levels, it will show custom (99).

Note: Get/set trace level and collect trace files are supported on the following processes. For any process that is not in this list, the user must manually do it on the machine. If the levels have the same trace settings, then GetTraceLevel will always return the highest value of the same levels. An example, for Logger/baimport, always returns Level 3.

7.1.1 Trace: All Nodes

Table 7-1: Trace: All Nodes

Process	Level 0 (Default – Error)	Level 1 (Warning)	Level 2 (Informa tional)	Level 3 (Debug)	Notes
NM	0x00	0x0F	0x0F	0x0F	Node Manager: Will always return Level 2 for Level 1 and 2.
NMM	0x00	0x0F	0x0F	0x0F	Node Manager: Will always return Level 2 for Level 1 and 2.

Note: Admin Client is not supported.

7.1.2 Trace: Administration & Data Server (AKA Distributor AW)

Table 7-2: Trace: Administration & Data Server (AKA Distributor AW)

Process	Level 0 (Default - Error)	Level 1 (Warning)	Level 2 (Informational)	Level 3 (Debug)	Notes
CONFIGLOGG ER	0x00	0x0F	0xFF	0xFFF	
CMSNODE	0x00	0x00	0x00	0xFFFFF FFF	Will always return Level 2 for Level 0 through 2.
CMS_JSERVER	0x00	0x00	0x00	0xFFFFF FFF	Will always return Level 2 for Level 0 through 2.

Process	Level 0 (Default - Error)	Level 1 (Warning)	Level 2 (Informational)	Level 3 (Debug)	Notes
REPLICATION	0x00	0x0F	0xFF	0xFFF	
RTCLIENT	0x00	0x0F	0xFF	0xFFF	
RTDIST	0x00	0x0F	0xFF	0xFFF	
UPDATEAW	0x00	0x0F	0xFF	0xFFF	
ISEMAN	0x00	0x00	0x00	0x01	Will always return Level 2 for Level 0 through 2.

7.1.3 Trace: Router

Table 7-3: Trace: Router

Process	Level 0 (Default - Error)	Level 1 (Warning)	Level 2 (Informational)	Level 3 (Debug)	Notes
APPGW	0x00	0x01	0x07	0x3F	
CCAGENT	0x00	0x03	0x0F	0xFF	
DBAGENT	0x00	0x01	0xFF	0xFF	Will always return Level 3 for Level 2 and 3.
DBWORKER	0x00	0x01	0xFF	0xFF	Will always return Level 3 for Level 2 and 3.
MDSPROC	0x00	0x07	0xFF	0xFF	Will always return Level 3 for Level 2 and 3.
ROUTER *	Turn off everything Hex=0x0,0x0	Route Requests Hex=0x10000,0x0	Network VRU Trans Route VRU Bank CIC Request Script Select Hex=0x91100000,0x1100000	Call Queuing Agent changes Call Type Real Time Hex=0x91140000,0x1B0000	Use RTRTRACE or RTRTEST Note: Router Restart will clear these settings.
RTSVR	0x00	0x0F	0xFF	0xFFF	

7.1.4 Trace: Logger

Table 7-4: Trace: Logger

Process	Level 0 (Default - Error)	Level 1 (Warning)	Level 2 (Informational)	Level 3 (Debug)	Notes
BAIMPORT	0xFF EMSUserData = 0xFFFF	0xFF EMSUserData = 0xFFFF	0xFF EMSUserData = 0xFFFF	0xFF EMSUserData = 0xFFFF	Will always return Level 3.
CAMPAIGN MANAGER	0xFF EMSUserData = 0xFFFF	0xFF EMSUserData = 0xFFFF	0xFF EMSUserData = 0xFFFF	0xFF EMSUserData =	Will always return Level 3

Process	Level 0 (Default - Error)	Level 1 (Warning)	Level 2 (Informational)	Level 3 (Debug)	Notes
				0xFFFF	
CONFIGLOGGER	0x00	0x0F	0xFF	0xFFF	
CSFS	0x00	0x00	0x00	0xFF	Will always return Level 2 for Level 0 through 2.
CW2KFEE D	0x00	0x00	0x00	0x07	Will always return Level 2 for Level 0 through 2.
DTP	0x00	0x04	0x06	0x0F	
HISTLOGGER	0x00	0x0F	0xFFF	0xFFF	
RECOVERY	0x00	0x0F	0xFFF	0xFFF	
REPLICATION	0x00	0x0F	0xFFF	0xFFF	

7.1.5 Trace: Peripheral Gateway

Table 7-5: Trace: Peripheral Gateway

Process	Level 0 (Default - Error)	Level 1 (Warning)	Level 2 (Informational)	Level 3 (Debug)	Notes
JTAPIGW *	JT_JTAPI_EVENT_USED JT_TPREQUESTS JT_PIM_EVENT JT_ROUTE_MESSAGE hex=0x4860	JT_CONNECTION *CONF* hex=0x4962	JT_JTAPI* JT_HEX JT_ROUTE* JT_TERM* JT_LOW* hex=0x7f7f	JT* hex=0xffff	Use PROCMON. When setting the new trace settings, the previous settings must be cleared, then apply the new settings.
MDSPROC	0x00	0x07	0x0F	0xFF	
MSGIS	0x00	0x00	0x00	0x3F	EMSUserData=0x00 Will always return Level 2 for Level 0 thru 2.
OPC **	default, cstacer DebugControlFlags1 =0x01000000 DebugControlFlags2 =0x10000000 EMSTraceMask = 0x40	agent, incmsg, closedcalls, tpmsg, routing DebugControlFlags1=0x01940000 DebugControlFlags2=0x	calls, NCT, simplified DebugControlFlags1=0x01b40000 DebugControlFlags2=0x05630000 EMSTraceMask = 0x40 Note:- Need to	Missingdata, halfhour DebugControlFlags1=0x01b60000 DebugControlFlags2=0x05630000 EMSTraceMask = 0x40	Use OPCTest. When setting the new trace settings, the default setting must be removed first.

Process	Level 0 (Default - Error)	Level 1 (Warning)	Level 2 (Informational)	Level 3 (Debug)	Notes
		00220000 EMSTrace Mask = 0x40 Note: Need to remove "default" tracing set in Default(0) level, but to include cstacer.	remove "default" tracing set in Default(0) level, but to include Level 1.	Note: Need to remove "default" tracing set in Default(0) level, but to include Level 2.	
PGAGENT	0x00	0x03	0x0F	0xFF	
EAGTPIM *	tp* precall *event csta* call_object teld_agent_state opcrequest hex=0x000000000000 0000088 0000000000000200 000002 0000000000000000 00000 07FFFFFFF060	periph* jtapi_dialed * hex=0x0000 0000000000 00984 0000000000 0002000000 0200 0000000000 0000000000 0 7FFFFFFF 060	autoconfig* - teld* call_match_timing timer* hex=0x0000000000 00000D878 0000000000002780 00002 0000000000000000 00000 007FFFFFFF07A	lock* universal* service* threadid jtapi* hex=0x00000000 00000000 0DFF80000000 000003F80000 0FFE00000000 000000000000 007FFFFFFF FE	Use PROCMON. When setting the new trace settings, the previous settings must be cleared, then apply the new settings.
VRUPIM	EMSTraceMask= 0x0 EMSUserData= 0x0	EMSTrace Mask= 0x0 EMSUserDa ta= 0x0	EMSTraceMask= 0x0 EMSUserData= 0x0	EMSTraceMask = 0x0 EMSUserData= 0x7FF7E0	Will always return Level 2 for Level 0 thru 2.
ACMIPIM	EMSUserData = (hex) 01, 7f, 46, 00, 00, 00, 00, 00, 00, 00, 00, 01, 00, 00, 00, 00, 00, 3f, ff, ff, ff, 67, cf, d7, fd, ef, ff, ff, ff, ff, ff, ff, ff, ff, ff, ff, f0, fa For reference, this is the default + all_peripherals	EMSUserDa ta = (hex) f5, 7f, 46, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 01, 00, 00, 01, 00, 00, 00, 00, 00, 3f, ff, ff, ff, 67, cf, d7, fd, ef, ff, ff, ff, ff, ff, ff, ff, ff, ff, ff, f0, fa For reference this is level 0 + timer	EMSUserData = (hex) f5, 7f, c6, 00, 00, 00, 00, 00, 00, 00, 00, 01, 00, 00, 00, 00, 00, 3f, ff, ff, ff, 67, cf, d7, fd, ef, ff, ff, ff, ff, ff, ff, ff, ff, ff, ff, f0, fa For reference this is level 1 + Monitor Item traversal	EMSUserData = (hex) f5, 7f, f6, 00, 00, 00, 00, 01, ff, ff, fe, c1, 00, 00, 00, 00, 00, 3f, ff, ff, ff, 67, cf, df, fd, ef, ff, ff, ff, ff, ff, ff, ff, ff, ff, ff, ff, fe For reference this is level 2 + locks + socket data	

Process	Level 0 (Default - Error)	Level 1 (Warning)	Level 2 (Informational)	Level 3 (Debug)	Notes
		events			
ARSPIM *	tp* precall *event csta* call_object teld_agent_state opcrequest hex=0x000000000000 0000088 0000000000000200 00000200 0000000000000000 000 07FFFFFFF060	periph* hex=0x0000 0000000000 00984 0000000000 0002000000 0200 0000000000 00000000 0007FFFFFF FFF060	autoconfig* teld* call_match_timing timer* hex=0x0000000000 000000D878 0000000000002780 0000200 0000000000000000 0000 07FFFFFFF07	lock* universal* service* threadid hex=0x00000000 00000000DFFF 80 000000000003F 80000027800 000000000000 0000 007FFFFFFF FE	Use PROCMON. When setting the new trace settings, the previous settings must be cleared, then apply the new settings.
MRPIM	EMSUserData = 0x00 Procmon: > trace mr* /off hex=0x00	EMSUserDa ta = 0x40 Procmon: > trace mr* /off > trace mr_msg_co mm_session /on hex=0x40	EMSUserData = 0x58 Procmon: > trace mr* /off > trace mr_msg_comm_ses sion /on > trace mr_*_mr /on hex=0x58	EMSUserData = 0x5F Procmon: > trace mr* /off > trace mr_msg_comm _session /on > trace mr_*_mr /on > trace mr_*_inrc /on > trace mr_*_csta /on hex=0x5F	
CTISVR	0x00	0xF0	0xF8	0xFF	Removing Callstate because 0xF8 and 0xFF already included it.
CTIOS SERVER NODE	0x00000003	0x00000A0 F	0x00020A0F	0x00060A0F	
BADIALE R	0x1F EMSUserData= FFFF	0x3F EMSUserDa ta= FFFF	0x7F EMSUserData= FFFF	0xFF EMSUserData= FFFF	Both SCCP and SIP dialers are using the same EMSTraceMask.

7.1.6 Diagnostic Framework

Table 7-6: Diagnostic Framework

Process	Level 0 (Default - Error)	Level 1 (Warning)	Level 2 (Informational)	Level 3 (Debug)	Notes
Diagnostic Framework	Info	Info	Info	Debug	Will always return Level 2 for Level 0 through 2.

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 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
```

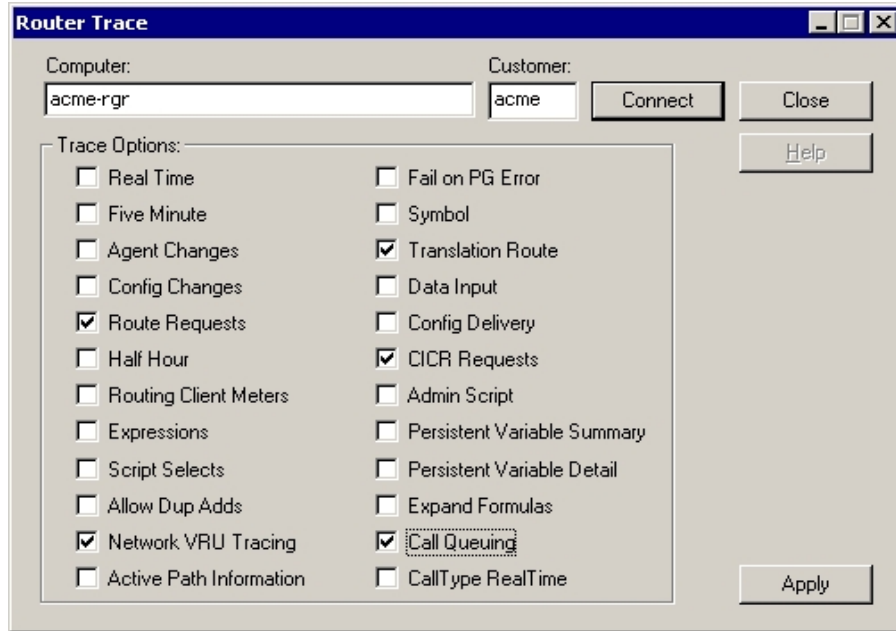


Figure 28: Router Trace Utility

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

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

Table 7-7: Setting Router Tracing

For any type of VRU	enable "Network VRU Tracing"
For NAM-CICM (Hosted)	enable "CICR Requests"
For suspected queuing issues	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.

You can "observe" specific status of call routing, call type, skill group and schedule target variables using the following RTTEST command:

```
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 (e.g. "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 /agent /routing /cstacer /rcmsg /closedcalls /inrcmsg /simplified /calls
```

7.3.2 Diagnosing Network Transfer Issues

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

7.3.3 Diagnosing Multi Media Issues

```
opctest:debug /agent /routing /cstacer /rcmsg /closedcalls /inrcmsg /task /passthru/simplified /calls
```

7.3.4 Diagnosing VRU PG Issues

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

The Default in Unified ICM/CC release 7.x is “no tracing”. The reason why is to ensure that during normal operation, tracing does not needlessly impact performance. There is no point in having tracing if the system is running fine. However, starting in release 7.5, the default tracing has been changed (increased) to minimally trace important events, albeit at a level that will not adversely impact performance. As of this date, testing is ongoing to determine the highest level possible. Current plans are:

```
opctest:debug /agent /routing /cstacer /rcmsg /closedcalls /inrcmsg /task
/passthru/simplified/ calls
```

and

```
EMSTracemask = 0x40
```

The latter being reset in the Windows registry.

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

7.4 Setting Unified CCM PIM Tracing

Resetting trace levels with the Unified Communications Manager Peripheral Interface Manager component (AKA “EAGTPIM”) is most easily accomplished using the PROCMON (process monitoring) utility. This is a command-line utility so remote desktop or local console access is required.

Table 7-8: Setting Unified CCM PIM Tracing

Command Syntax (launch)	C:> procmon <instance> <node> pim<pim number>
Example	C:> procmon acme PG1A pim1
Commands	<pre>>>>trace tp* /on >>>trace precall /on >>>trace *event /on >>>trace csta* /on</pre>
Where	
tp	Traces 3 rd party call events/messages
precall	As it implies, traces the precall events
*event	Traces JTAPI call events coming back from the JTAPI Gateway
csta*	Traces all CSTA messages going to OPC

7.4.1 ARS Gateway Registry Trace Settings

Table 7-9: Setting ARS Gateway Registry Tracing

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

7.4.1 ARS PIM Trace Settings

Table 7-10: 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 (AKA “JTAPIGW”) is most easily accomplished using the PROCMON (process monitoring) utility. This is a command-line utility so remote desktop or local console access is required.

Table 7-11: Setting Unified CM PIM Tracing

Command Syntax (launch)	C:> procmon <instance> <node> jgw<jtapigw number>
Example	C:> procmon acme PG1A jgw1
Commands	>>>trace JT_TPREQUESTS /on

	<pre>>>>trace JT_JTAPI_EVENT_USED* /on >>>trace JT_PIM_EVENT /on >>>trace JT_ROUTE_MESSAGE /on >>>trace *CONF* /on</pre>
Where	
JT_TPREQUESTS	3 rd party requests
JT_JTAPI- EVENT_USED	JTAPI events from the JTAPI Client
JT_PIM_EVENT	PIM events from JTAPI Gateway to the PIM
JT_ROUTE_MESSAGE	Route messages from the JTAPI client
CONF	Conference information – troubleshooting traces internal to JTAPI

7.6 Setting CTI Server Tracing

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

Table 7-12: 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) The value of F0 will provide sufficient tracing information to troubleshoot most issues

7.7 Setting CTI OS Tracing

Resetting trace levels with the Unified ICM/CC CTI Object Server (AKA CTI OS) is accomplished by altering the trace mask saved in the Windows registry. Use the Windows REGEDIT utility to change this numeric value.

Table 7-13: 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)

	Increasing the trace levels (other than the Default (0x3)) will impact the CTIOS Server performance. High Tracemask needs to be reverted to the default trace levels after collecting the required logs.
Levels	Level 0: 3 Level 1: 0X240A2F (Recommended default debug level.) Level 2: 0X260A2F (Recommended if multi-threaded issues are suspected.) Level 3: 0x2E0A2F (Recommended max debug trace level.)

7.8 Setting VRU PIM Tracing

Resetting trace levels with the Unified ICM/CC VRU Peripheral Interface Manager (PIM) is accomplished by altering the trace mask and user data values saved in the Windows registry. Use the Windows REGEDIT utility to change these numeric values.

Table 7-14: 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 collecting 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)
```

7.9 Setting Outbound Option Tracing

The 8.0(1) utility tools will provide centralized control for setting up each component trace level. Additionally, you can manually modify the registry key values.

7.9.1 Setting CampaignManager Tracing

Resetting trace levels with the CampaignManager of Unified ICM/CC Outbound Option can also be accomplished by altering the trace mask saved in the Windows registry. Use the Windows REGEDIT utility to change this numeric value.

Registry Key:

HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\<instance>\LoggerA\EMS\CurrentVersion\ Library\Processes\CampaignManager

Example:

HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\m3pc1\LoggerA\EMS\CurrentVersion\Library\Processes\CampaignManager

Registry	Default (0)	1	2	3
EMSTraceMask	0xFF	0xFF	0xFF	0xFF
EMSUserData	0xFFFF	0xFFFF	0xFFFF	0xFFFF

7.9.2 Setting baImport Tracing

Resetting trace levels with the baImport of Unified ICM/CC Outbound Option can also be accomplished by altering the trace mask saved in the Windows registry. Use the Windows REGEDIT utility to change this numeric value.

Registry Key:

HKLM\SOFTWARE\Cisco Systems, Inc.\ICM

Example:

HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\m3pc1\LoggerA\EMS\CurrentVersion\Library\Processes\baImport

Registry	Default (0)	1	2	3
EMSTraceMask	0xFF	0xFF	0xFF	0xFF
EMSUserData	0xFFFF	0xFFFF	0xFFFF	0xFFFF

7.9.3 Setting Dialer Tracing

Resetting trace levels for Dialer of UCC Outbound can also be accomplished by altering the trace mask saved in the Windows registry. Use the Windows REGEDIT utility to change this numeric value.

Registry Key:

HKLM\SOFTWARE\Cisco Systems, Inc.\ICM

Example:

HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\m3pc1\LoggerA\EMS\CurrentVersion\Library\Processes\Dialer

SIP dialer:

Registry	Default (0)	1	2	3
EMSTraceMask	0x1F	0x3F	0x7F	0xFF
EMSUserData	0xFFFF	0xFFFF	0xFFFF	0xFFFF

SCCP dialer:

Registry	Default (0)	1	2	3
EMSTraceMask	0xFF	0xFF	0xFF	0xFF
EMSUserData	0xFFFF	0xFFFF	0xFFFF	0xFFFF

7.10 Setting Trace File Retention Parameters

There are several Windows registry values that can be modified to adjust the trace log retention parameters, e.g. 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\

```

Example:

```
C:\icm\acme\ppla\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 this process's trace log files, 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:

Table 7-15: Registry Items

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

Note: **EMSLogFileMax** multiplied by **EMSLogFileCountMax** may be greater-than **EMSAIILogFilesMax** 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. **EMSAIILogFilesMax** 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 Platform 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
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 - 100%)	< 60%	60% - 80%	> 80%
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%
<p>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.</p> <p>This counter is typically a good indicator of whether there is sufficient RAM installed in the server. 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.</p>						

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 Platform 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.

Table 8-2: Platform Diagnostic Counters Values

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 8-3: Performance Counters - Diagnostics

Component	Counter Name	Type	Units (Range)
Processor	% Processor Time (_Total)	Int32	Percentage (0 – 100%)
<p>% 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</p>			

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 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 Platform Diagnostic Counters

8.3.1 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 8-4: 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

8.3.2 Logger / Administration & Data Server / 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 8-5: Diagnostic Counters - Logger, Administration & Data Server, 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

8.3.3 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 8-6: 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	cscowawdb ¹	Transactions / sec
SQLServer:Databases	cscowhds 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 “cscow” is the Unified ICM/CC instance name.

8.3.4 WebView

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

Table 8-7: 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

Performance Object	Instance	Counter Name
Process	java	Virtual Bytes
Process	java	Working Set
Web Service	_Total	Current Connections

8.4 Component-Specific Counters

8.4.1 Router

Table 8-8: Router Performance Counters

Performance Object: Cisco ICM Router		
Counter Instance: "{ICM Instance Name}" – if multiple instances installed		
Always ON?	Counter Name	Description
Y	Agents Logged On ¹	The number of (contact center) agents currently logged on.
Y	Calls In Progress ¹	The number of calls currently in progress (being controlled by the CCE application).
Y	Calls/sec ¹	The (calculated) inbound call rate measured in the number of calls received per second.
Y	Calls In Queue	The number of calls queued in all network Voice Response Units (VRUs), from the router's perspective, including those calls that are in the process of transferring to the VRU for queuing.
Y	Calls In Router	Number of active calls in the Router, including the calls sent to VRU for treatment or queuing and the calls the Router is waiting for response from the routing client.
N	Router State Size	The current router state size - the total size of all of the state transfer objects in router memory; this size is measured in kilobytes. After one router side goes out of service, when it returns in-service, the router state is transferred from the surviving router side to the returning router side.
N	Messages Processed/sec	The number of MDS messages router processed. By default, this counter is disabled.
N	Bytes Processed/sec	The rate of the data bytes the router processed. By default, this counter is disabled.
N	Avg Process Time/Message (ms)	The average time (in milliseconds) the router spends processing a MDS message.
N	Max Process Time(ms)	The maximum time (in milliseconds) the router spends processing a MDS message.

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 counter 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.

8.4.2 Logger

Table 8-9: Logger Performance Counters

Performance Object: Cisco ICM Logger		
Counter Instance: "{ICM Instance Name}" – if multiple instances installed		
Always ON?	Counter Name	Description
Y	Number of DB Write Records	The number of database writes (records/rows) in the historical logger process that is written to the database at the time the counter is polled.
Y	DB Write Average Time	The average database write time expresses the average amount of time, in 100 nanosecond units, required to write data to a table in the central controller database. This value represents the average time per write of the write operations that occurred in the past second. This object is a good indicator of contention for database access.
Y	DB Write Records Processed	The number of records processed – written to the database – in the Historical Logger Process in the past second.

8.4.3 Administration & Data Server

Table 8-10: Administration & Data Server Real-Time Counter

Performance Object: Cisco ICM Distributor RealTime		
Counter Instance: {Instance Name} ADS#		
Always ON?	Counter Name	Description
Y	Agent Queue Depth	The queue depth – number of pending write transactions – for the Agent table in the Real-Time Client process.
Y	Agent DB Write Average Time	The average time – in units of 100 ns – for the Real-Time Client process to write an Agent table transaction within the past 1 second interval.
Y	Agent DB Write Records Processed	The number of Agent table records written by the Real-Time Client process in the past 1 second interval.
Y	Agent Skill Group Queue Depth	The queue depth – number of pending write transactions – for the Agent Skill Group table in the Real-Time Client process.
Y	Agent Skill Group DB Write Average Time	The average time – in units of 100 ns – for the Real-Time Client process to write an Agent Skill Group table transaction within the past 1 second interval.
Y	Agent Skill Group DB Write	The number of Agent Skill Group table records

	Records Processed	written by the Real-Time Client process in the past 1 second interval.
Y	Skill Group Queue Depth	The queue depth – number of pending write transactions – for the Skill Group table in the Real-Time Client process.
Y	Skill Group DB Write Average Time	The average time – in units of 100 ns – for the Real-Time Client process to write an Skill Group table transaction within the past 1 second interval.
Y	Skill Group DB Write Records Processed	The number of Skill Group table records written by the Real-Time Client process in the past 1 second interval.
Y	CallType Queue Depth	The queue depth – number of pending write transactions – for the CallType table in the Real-Time Client process.
Y	CallType DB Write Average Time	The average time – in units of 100 ns – for the Real-Time Client process to write an CallType table transaction within the past 1 second interval.
Y	CallType DB Write Records Processed	The number of CallType table records written by the Real-Time Client process in the past 1 second interval.
Y	Route Queue Depth	The queue depth – number of pending write transactions – for the Route table in the Real-Time Client process.
Y	Route DB Write Average Time	The average time – in units of 100 ns – for the Real-Time Client process to write an Route table transaction within the past 1 second interval.
Y	Route DB Write Records Processed	The number of Route table records written by the Real-Time Client process in the past 1 second interval.
Y	Service Queue Depth	The queue depth – number of pending write transactions – for the Service table in the Real-Time Client process.
Y	Service DB Write Average Time	The average time – in units of 100 ns – for the Real-Time Client process to write an Service table transaction within the past 1 second interval.
Y	Service DB Write Records Processed	The number of Service table records written by the Real-Time Client process in the past 1 second interval.

Table 8-11: Administration & Data Server Replication Counters

Performance Object: Cisco ICM Distributor Replication		
Counter Instance: {Instance Name} Distributor #		
Always ON?	Counter Name	Description
Y	DB Write Average Time	The average time – in units of 100 nanoseconds – for database write operations in the HDS Replication process during the past 1 second interval.
Y	DB Write Records Processed	The number of records written by the HDS Replication process in the past 1 second interval.

8.4.4 PG – OPC

Table 8-12: PG - OPC Counters

Performance Object: Default: Cisco ICM OPC Optionally Enabled: Cisco ICM OPC (Optional)		
Counter Instance: “{Instance Name} PG#A/B” (Ex: “acme PG3A”)		
Always ON?	Counter Name	Description
Y	Call Count	Number of Calls that are currently active.
N	Agent Count	An Agent is a specific individual who receives calls through the peripheral. This counter will provide the information on the number of Agents that are configured in the system.
N	Skill Group Count	A skill group is a group of agents who share a common set of skills and who can, therefore, all handle specific types of calls. Each skill group contains one or more agents. If supported by the peripheral, each agent can be a member of more than one skill group. This counter gives the number of various Skill groups available for the agents to login.
N	Services Count	A service is a type of processing the caller requires. A peripheral might have services defined for sales, technical support, or opening new accounts. Each service has one or more skill groups whose members can provide the service. Each skill group can be associated with more than one service. This counter gives the number of services that are configured to process the calls.
Y	Logged-In Agent Count	This counter gives the number of agents that have logged on. This does not necessarily indicate that the agents are ready to accept calls.

Y	Ready Agent Count	Number of Agents that are logged on and are ready to accept calls.
N	Not-Ready Agent Count	Number of Agents that are logged on, but occupied with task other than accepting incoming calls.
Y	Talking Agent Count	Number of Agents currently talking on Inbound or Outbound calls.
N	Held Agent Count	Number of Agents that are inactively participating in a call.
N	Work-Ready Agent Count	Agents occupied with work associated with the last call. This implies that agent is no longer connected to the call and will be ready to receive additional calls when they exit this state.
N	Work-Not-Ready Agent Count	Agents occupied with work associated with the last call. This implies that agent is no longer connected to the call. These Agents will not be ready to receive additional calls when they exit this state.
N	Logged-Out Agent Count	Number of Agents that are logged out of the system. This count will help in validating the statistics if there are any state mismatches.
N	None-State Call Count	This count will give the number of calls for which a call object was created but no activity.
N	Null-State Call Count	This count will give the number of calls that has no relationship between the call and device.
N	Initiated Call Count	This count will give the number of calls for which the device has requested for a service. Often this is the "dialing" state.
N	Alerting Call Count	This count will give the number of calls for which the device is in alerting (ringing) state. This indicates that a call wishes to become connected to a device.
Y	Connected Call Count	This count will give the number of calls for which the device is actively participating in the call.
N	Held Call Count	This count will give the number of calls for which the device is inactively participating in the call.
N	Queued Call Count	This count will give the number of calls for which the normal state progression has been stalled. This state generally refers to two conditions but can apply to others as well. One condition is when a device is trying to establish a Connection with a call, and the process is stalled. The second condition is when a call tries to establish a Connection with a device and that process is stalled.
N	Failed Call Count	This count will give the number of calls for which the normal state progression has been aborted. This state generally refers to the condition when a device tries to

		become Connected to a call or a call tries to become Connected to a device and the attempt fails. Failed can result because of failure to connect the calling device and call, failure to connect the called device and call, failure to create the call, and other reasons.
--	--	--

8.4.5 PG – Communications Manager (EA) PIM

Table 8-13: PG - CM PIM Counters

Performance Object: Default: Cisco ICM CMPIM		
Optionally Enabled: Cisco ICM CMPIM (Optional)		
Counter Instance: “{Instance Name} PG#A/B PIM#” (Ex: “acme PG3A PIM1”)		
Always ON?	Counter Name	Description
N	Agent Count	Number of agents that are currently configured in system.
N	Calls per sec	Number of incoming calls per second.
Y	Call Count	Number of calls that are in progress.
N	Invalid Call Count	Number of calls that are not in any of the valid call states.
N	Messages per second	Number of call events, agent events exchanged per second between the JTAPI Gateway and CM PIM.
N	Messages sent	Number of call events, agent events, and CSTA messages sent today.
N	Messages sent past 5	Number of call events, agent events, and CSTA messages sent past 5 seconds.

8.4.6 PG – VRU PIM

Table 8-14: PG - VRU PIM Counters

Performance Object: Cisco ICM VRUPIM		
Counter Instance: “{Instance Name} PG#A/B PIM#” (Ex: “acme PG3A PIM3”)		
Always ON?	Counter Name	Description
Y	Calls At VRU	Calls at VRU is the number of calls that are currently at the Voice Response Unit (VRU). For a VRU that only uses a Call Routing Interface, this value is zero.
N	Messages To VRU/sec	Messages To VRU/sec is the rate at which messages are sent to the Voice Response Unit (VRU). This counter is only active when enabled in ICM registry.
N	Messages From VRU/sec	Messages From VRU/sec is the rate at which messages are received from the Voice Response Unit (VRU). This counter is only active when enabled in ICM registry.
N	Bytes To VRU/sec	Bytes To VRU/sec is the rate at which bytes are sent to the

		Voice Response Unit (VRU). This counter is only active when enabled in ICM registry.
N	Bytes From VRU/sec	Bytes From VRU/sec is the rate at which bytes are received from the Voice Response Unit (VRU). This counter is only active when enabled in ICM registry.
Y	New Calls/sec	New Calls/sec is the rate at which new calls arriving at the Voice Response Unit (VRU). New calls are calls not under ICM script control when arriving at a Service Control VRU.
Y	Pre-Routed Calls/Sec	Pre-Routed Calls/sec is the rate at which Pre-Routed calls are arriving at Voice Response Unit (VRU). Pre-Routed calls are calls under ICM script control when arriving at a Service Control VRU.
Y	Connection Resets	Connection Resets is the number of times the TCP connection between ICM and the Voice Response Unit has been changed from an “established” state to a “closed” state since the application was started.

8.4.7 CTI Server

Table 8-15: CTI Server Counters

Performance Object: Default: Cisco ICM CTISVR		
Optionally Enabled: Cisco ICM CTISVR (Optional)		
Counter Instance: “{Instance Name} CG#A/B” (Ex: “acme CG3A”)		
Always ON?	Counter Name	Description
N	Reported Call Count	Number of calls that are already reported to the CTI clients.
N	Active Call Count	Number of calls that are currently in progress.
N	Deactivated Call Count	Number of calls that are not currently active and will be eventually cleared.
N	Cleared Call Count	Number of calls that no longer exist in the system.
N	Private Call Count	Number of calls that are privately tracked by CTI Server and which are not reported to OPC.
Y	Logged-In Agent Count	Agents that have logged on. This does not necessarily indicate that they are ready to accept calls.
Y	Ready Agent Count	Number of Agents that are logged on and are ready to accept calls.
N	Not-Ready Agent Count	Number of Agents that are logged on, but occupied with task other than accepting incoming calls.
Y	Talking Agent Count	Number of Agents currently talking on Inbound or Outbound calls.
N	Held Agent Count	Number of Agents that are inactively participating in a call.
N	Work-Ready Agent	Agents occupied with work associated with the last call. This

	Count	implies that agent is no longer connected to the call and will be ready to receive additional calls when they exit this state.
N	Work-Not-Ready Agent Count	Agents occupied with work associated with the last call. This implies that agent is no longer connected to the call. These agents will not be ready to receive additional calls when they exit this state.
N	Logged-Out Agent Count	The number of Agents that are logged out of the system. This count will help in validating the statistics if there are any state mismatches.
Y	Sessions Unknown	The number of sessions for which there is no socket connection made yet.
N	Sessions Opening	The number of sessions that are in the process of setting up a connection.
Y	Sessions Open	The number of sessions that has been successfully setup.
N	Sessions Closing	The number of sessions that are in the process of tear down.
Y	Sessions Closed	The total number of sessions that are terminated by the CTI Server.
Y	Sessions Failed	The number of sessions that failed due to various reasons like missing heartbeat, open request timeout, session inactivity etc. These timers are configurable parameters in CTI Server.
Y	Total Sessions	The total number of sessions maintained by CTI Server.

8.4.8 CTI OS Server

Table 8-16: CTI OS Server Counters

Performance Object: Cisco ICM CTI OS		
Counter Instance: CTI OS Name		
Always ON?	Counter Name	Description
Y	CTI OS Active Client Connections	The number of CTI OS Active Client Mode Desktop Connections. This value indicates the total number of desktops connected to the CTIOS server. The number of desktops connected to the A and B side of CTIOS determine the total desktops connected through this instance of CTI OS server.
Y	CTI OS Active Monitor Mode Connections	The number of CTI OS Active Monitor Mode Desktop Connections. CTIOS only supports two monitor mode connections per each CTI OS server. This value indicates how many monitor mode connections are in use. Once there are two in use further monitor mode connection attempts are rejected.
Y	CTI OS Active Calls	The total number of active calls being tracked by CTI OS. This value shows how many calls

		are currently being handled by CTI OS. This value should go up and down based on the call arrival rate and the agent call completion rate.
Y	CTI OS Configured Skill Groups	The total number of configured skill groups being tracked by CTI OS. This value should match the number of skill groups configured for the PG that this CTI OS is associated.
Y	CTI OS Configured Teams	The total number of configured Teams being tracked by CTI OS. This value should match the number of teams configured for the PG that this CTI OS is associated.
Y	CTI OS Configured Agents	The total number of configured Agents being tracked by CTI OS. This value should match the number of Agents configured for the PG that this CTI OS is associated.
Y	CTI OS Active Conferences	The total number of active Conferences being tracked by CTI OS. This value indicates the number of multi-party calls that are in progress at any one given time in CTI OS.
Y	CTI OS Call Count	The total number of calls handled by CTI OS. This value only increases and shows the total number of calls processed by CTI OS since it last started. This value should increase at the same rate as the calls per second being shown by the router.
Y	CTI OS Conference Count	The total number of Conferences performed by CTI OS. This value only increases and shows the total number of calls that were conferenced since CTI OS last started. The conference count should be a small percentage of total calls.
Y	CTI OS Transfer Count	The total number of Transfers performed by CTI OS. This value only increases and shows the total number of calls that were transferred since CTI OS last started. The transfer count should be a small percentage of total calls.
Y	CTI OS Call Failed Count	The total number of Calls that failed reported to CTI OS. This value shows the total number of calls that failed via a failure event being reported to CTI OS. If this count begins to rise the log file should be captured to gather more specific information on the failure events.
Y	CTI OS CTI Message Receive Rate	The rate at which CTI OS receives messages from CTI Server per second. This value is an indicator to total load on the system. Increases are not really a problem unless the CTI OS Service Broker Queue Size also begins to increase.

Y	CTI OS CTI Message Send Rate	The rate at which CTI OS sends messages to CTI Server per second. This value is an indicator of total load on the system. If it increases it indicate the CTI OS server is under a heavy request load from the desktop clients.
Y	CTI OS Service Broker Queue Size	The number of messages queued in the CTI OS Service Broker queue. This value is a good load indicator for CTI OS. If it increases it suggests that CTI OS is not keeping up with the incoming message rate from CTI Server. A review of the configuration may be necessary to understand why CTI OS is not able to keep up with event handling from CTI Server.
N	CTI OS Call Object Count	The total number of CTI OS call objects that are active. This value shows how many CTI OS Call objects were created since it last started. This value should go up and down and may reach a steady state when the number of calls being completed by agents equals the call arrival rate.
N	CTI OS Connection Object Count	The total number of active CTI OS connection objects. This value shows how many CTI OS connection objects were created since it last started. This value should go up and down and may reach a steady state when the number of calls being completed by agents equals the call arrival rate.
N	CTI OS Argument Object Count	The total number of active CTI OS argument objects. This value shows how many CTI OS argument objects were created since it last started. This value shall be quite large, go up and down and may reach a steady state when the number of calls being completed by agents equals the call arrival rate.
N	CTI OS Device Object Count	The total number of active CTI OS devices. This value shows how many CTI OS device objects were created since it last started. This value should mainly stay constant while CTI OS runs.
N	CTI OS Agent Object Count	The total number of CTI OS agent objects. This value shows how many CTI OS agent objects were created since it last started. This value should stay constant while CTI OS runs unless agents are added or deleted.
N	CTI OS Skill group Object Count	The total number of CTI OS skill group objects. This value shows how many CTI OS skill group objects were created since it last started. This value should stay constant while CTI OS runs unless skill groups are added or deleted.

N	CTI OS Supervisor Object Count	The total number of CTI OS Supervisor objects. This value shows how many CTI OS supervisor objects were created since it last started. This value should stay constant while CTI OS runs unless supervisors are added or deleted.
N	CTI OS Team Object Count	The total number of CTI OS Team objects. This value shows how many CTI OS team objects were created since it last started. This value should stay constant while CTI OS runs unless teams are added or deleted.
N	CTI OS Total Objects Created Count	The total count of all objects created by CTI OS. This value shows how many CTI OS objects were created since it last started. This value shall only increase and shall grow very large as CTI OS up time increases.
N	CTI OS Total Objects Deletion Count	The total count of all objects deleted by CTI OS. This value shows how many CTI OS objects were deleted since it last started. This value shall only increase and shall grow very large as CTI OS up time increases. It shall never equal the total objects created count as some objects are never deleted after being created by CTI OS like agent, device, team and skill group objects.
N	CTI OS Active Object Count	The total count of all objects created by CTI OS that are active. This value shows how many CTI OS objects are currently allocated since it last started. If this value begins to increase it would indicate that a memory leak is occurring in CTI OS. The specific object counters shall show which object is not being released.
N	CTI OS CLIENT Send Message Rate	The rate at which CTI OS sends messages to Clients per second. This value shows the number of messages, per second, that CTI OS is delivering messages to CTI OS desktops. As this value increases it indicates that CTI OS server is being placed under an increasing load. A review of the configuration as its' related to agents, skill groups and teams may be necessary.
N	CTI OS CLIENT Receive Message Rate	The rate at which CTI OS receives messages from Clients per second. This value shows the number of messages, per second, that are being received from the CTI OS desktops. As this value increases it indicates that CTI OS is being placed under an increasing request load from the desktops.

8.4.9 Outbound Option Campaign Manager

Table 8-17: Outbound Option Campaign Manager Counters

Performance Object: Cisco ICM CampaignMgr		
Counter Instance: “{Instance Name}”		
Always ON?	Counter Name	Description
Y	DB Space Utilization	The Campaign Manager and Import processes share a private database on the Side A Logger. This shows what percentage of allocated space in the database is currently utilized. An administrator should start paying attention when this value exceeds eighty percent.
Y	Queue Depth	The Campaign Manager is a multithreaded process. There is one main dispatch thread that is involved in most processing. Queue Depth indicates how many messages are queued to this internal dispatch thread. By default, the Campaign Manager will crash when this value exceeds 10,000 messages in queue.
Y	Average Queue Time	The Campaign Manager is a multithreaded process. There is one main dispatch thread that is involved in most processing. This shows what is the average time spent in the main dispatch thread queue in milliseconds.
Y	Do Not Call Number Count	The Campaign Manager manages a global list of phone numbers used to prevent block dialing. This list is stored in memory. Each record uses 17 bytes of memory. This counter shows how many do not call entries are currently in memory.
Y	Active Dialer Count	The Campaign Manager process feeds several Dialer components which do all of the dialing of customers for outbound campaigns. This counter indicates how many Dialers are currently registered to the Campaign Manager.

8.4.10 Outbound Option Import

Table 8-18: Outbound Option Import Counters

Performance Object: Cisco ICM Import		
Counter Instance: “{Instance Name}”		
Always ON?	Counter Name	Description
Y	Records Imported Today	The Outbound Option Import process imports customer records which contain phone numbers used by the Campaign Manager and Dialer to find available customers for a campaign. This counter tracks how many records were imported today.

8.4.11 Outbound Option Dialer

Table 8-19: Outbound Option Dialer Counters

Performance Object: Cisco ICM Dialer		
Counter Instance: “{Instance Name}”		
Always ON?	Counter Name	Description
Y	Queue Depth	The Dialer is a multithreaded process which communicates between threads using inter thread messaging. This indicates how many messages are currently queued up for the main dispatch thread. By default, the Dialer process will be restarted when this value exceeds 10,000 messages.
Y	Average Queue Time	The Dialer is a multithreaded process that communicates between threads using messaging. There is one main dispatch thread that is involved in most processing. This shows what is the average time spent in queue.
Y	Talking Agents	For an agent campaign, the Dialer replaces calls to customers and transfers those customers to agents. This counter indicates how many agents are currently talking in the monitored campaign skill group.
Y	Busy Port (Customer) Count	The port is the unit on the Dialer that places calls to reserve agents and to contact customers. This counter tracks how many ports are currently busy trying to contact customers.
Y	Busy Port (Reservation) Count	The port is the unit on the Dialer that places calls to reserve agents and to contact customers. This counter tracks how many ports are currently busy reserving agents.
Y	Idle Port Count	The port is the unit on the Dialer that places calls to reserve agents and to contact customers. This counter tracks how many ports are currently idle.
Y	Call Attempt Count	The Dialer attempts to contact customers and transfer them to reserved agents or an available IVR. This counter tracks how many customer attempts were placed today. It will not include preview calls that were rejected or skipped.
Y	Abandoned Call Count	When a customer is contacted and an agent is not available to take the call, the call can be dropped or sent to the IVR for prompting and queuing. When either of these conditions occurs, the call is counted as abandoned. In a transfer to IVR campaign, a call will be dropped and counted as abandoned if the configured IVR port limit is exceeded.

Y	Reservation Call Count	The dialer places calls to agents to reserve them for use while attempting to contact available customers. This counter tracks how many reservation calls were placed today.
Y	Answering Machine Call Count	A campaign can be enabled to differentiate between live voice and answering machines. This counter tracks how many answering machines were detected today.
Y	Customer Answered Call Count	A campaign can be enabled to differentiate between live voice and answering machines. If answering machine detection (AMD) is enabled for a campaign this counter will be incremented when live voice is detected. If AMD is disabled, then all connected calls that are not FAX will be identified as live voice. Direct Preview calls will be identified as voice or AMD by the agent. This counter is reset daily at midnight.
Y	Customer Not Answered Call Count	The Dialer attempts to contact customers. This counter tracks how many attempts resulted in no answer condition. This counter is reset daily.
Y	Error Call Count	The Dialer attempts to contact customers. This counter tracks how many attempts resulted in a network error condition which includes no ring-back, no dial tone, and call disconnected from the network before ring no answer time out was exceeded.
Y	Number of attempted calls per second	This counter tracks how many calls per second the dialer is placing rounded to the nearest integer. If the dialing rate is too high, it can result in network congestion on the voice network which can result in inefficient dialing.

8.4.12 Message Delivery Service

Table 8-20: MDS Client Counters

Performance Object: Cisco ICM MDSCLIENT		
Counter Instance: “{Instance Name}”		
Always ON?	Counter Name	Description
N	Client Handle ID	Handle for this MDS client. It is used to uniquely identify the MDS client connected to the MDS process.
N	Now Message Received	Number of messages received by the MDS client per second.
N	Now Message Sent	Number of messages sent by the MDS client per second.

N	Now Bytes Received	Number of bytes received by the MDS client per second.
N	Now Bytes Sent	Number of bytes sent by the MDS client per second
N	Current Buffers Memory Allocated	Total number of bytes used by all currently allocated buffers.
N	Current Buffers Allocated	Total number of buffers currently allocated from buffer pool.
N	Buffers Allocation Requests/sec	Number of buffers allocated per second.
N	Buffers Free Requests/sec	Number of buffers freed per second.
N	Current Buffers Memory Limit	Maximum amount of memory allowed to be allocated for buffers for this process.
N	Initial Buffers Memory Limit	Amount of memory limit reserved for buffers for this process.
N	SendClientQ Current Depth	Current number of messages in the MDS Client Send Queue.
N	SendClientQ Now Messages In/sec	Total number of messages added to the MDS Client Send Queue per second.
N	SendClientQ Now Messages Out/sec	Total number of messages removed from the MDS Client Send Queue per second.
N	SendClientQ Now Bytes In/sec	Total number of bytes added for all messages to the MDS Client Send Queue per second.
N	SendClientQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the MDS Client Send Queue per second.
N	SendClientQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the MDS Client Send Queue per second.
N	SendClientQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the MDS Client Send Queue.
N	SendClientQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the MDS Client Send Queue will experience.

Table 8-21: MDS Process Client Counters

Performance Object: Cisco ICM MDSPROCCLIENT		
Counter Instance: "{Instance name}"		
Always ON?	Counter Name	Description
N	Client Handle ID	Handle for this MDS client. It is used to uniquely identify the MDS client connected to the MDS process.

N	Total MDS Client Connects	Total number of times the MDS client has connected to the MDS process.
N	Total MDS Client Disconnects	Total number of times the MDS client has disconnected from the MDS process.
N	Now Message Received from Client	Number of messages received from the MDS client per second.
N	Now Message Sent to Client	Number of messages sent to the MDS client per second.
N	Now Bytes Received from Client	Number of bytes received from the MDS client per second.
N	Now Bytes Sent to Client	Number of bytes sent to the MDS client per second.
N	ToClientQ Current Depth	Current number of messages in the MDS Send Client Queue.
N	ToClientQ Now Messages In/sec	Total number of messages added to the MDS Client Send Queue per second.
N	ToClientQ Now Messages Out/sec	Total number of messages removed from the MDS Client Send Queue per second.
N	ToClientQ Now Bytes In/sec	Total number of bytes added for all messages to the MDS Client Send Queue per second.
N	ToClientQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the MDS Client Send Queue per second.
N	ToClientQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the MDS Client Send Queue per second.
N	ToClientQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the MDS Client Send Queue.
N	ToClientQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the MDS Client Send Queue will experience.

Table 8-22: MDS Process Counters

Performance Object: Cisco ICM MDSPROC		
Counter Instance: “{Instance Name}”		
Always ON?	Counter Name	Description
N	Current Buffers Memory Allocated	Total number of bytes used by all currently allocated buffers.
N	Current Buffers Allocated	Total number of buffers currently allocated from buffer pool.
N	Buffers Allocation	Number of buffers allocated per second.

	Requests/sec	
N	Buffers Free Requests/sec	Number of buffers freed per second.
N	Current Buffers Memory Limit	Maximum amount of memory allowed to be allocated for buffers for this process.
N	Initial Buffers Memory Limit	Amount of memory limit reserved for buffers for this process.
N	Synch Messages Ordered/sec	Number of messages ordered by the MDS synchronizer per second.
N	Synch MDS Duplicates/sec	Number of duplicate MDS messages detected by the synchronizer per second.
N	Synch DMP Duplicates/sec	Number of duplicate DMP messages detected by the synchronizer per second.
N	LocalHighInQ Current Depth	Current number of messages in the Local High Incoming Queue.
N	LocalHighInQ Now Messages In/sec	Total number of messages added to the Local High Incoming Queue per second.
N	LocalHighInQ Now Messages Out/sec	Total number of messages removed from the Local High Incoming Queue per second.
N	LocalHighInQ Now Bytes In/sec	Total number of bytes added for all messages to the Local High Incoming Queue per second.
N	LocalHighInQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the Local High Incoming Queue per second.
N	LocalHighInQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the Local High Incoming Queue per second.
N	LocalHighInQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the Local High Incoming Queue.
N	LocalHighInQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the Local High Incoming Queue will experience.
N	LocalMedInQ Current Depth	Current number of messages in the Local Medium Incoming Queue.
N	LocalMedInQ Now Messages In/sec	Total number of messages added to the Local Medium Incoming Queue per second.
N	LocalMedInQ Now Messages Out/sec	Total number of messages removed from the Local Medium Incoming Queue per second.
N	LocalMedInQ Now Bytes In/sec	Total number of bytes added for all messages to the Local Medium Incoming Queue per second.
N	LocalMedInQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the Local Medium Incoming Queue per second.
N	LocalMedInQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the Local

		Medium Incoming Queue per second.
N	LocalMedInQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the Local Medium Incoming Queue.
N	LocalMedInQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the Local Medium Incoming Queue will experience.
N	LocalLowInQ Current Depth	Current number of messages in the Local Low Incoming Queue.
N	LocalLowInQ Now Messages In/sec	Total number of messages added to the Local Low Incoming Queue per second.
N	LocalLowInQ Now Messages Out/sec	Total number of messages removed from the Local Low Incoming Queue per second.
N	LocalLowInQ Now Bytes In/sec	Total number of bytes added for all messages to the Local Low Incoming Queue per second.
N	LocalLowInQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the Local Low Incoming Queue per second.
N	LocalLowInQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the Local Low Incoming Queue per second.
N	LocalLowInQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the Local Low Incoming Queue.
N	LocalLowInQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the Local Low Incoming Queue will experience.
N	RemoteHighOutQ Current Depth	Current number of messages in the Remote High Output Queue.
N	RemoteHighOutQ Now Messages In/sec	Total number of messages added to the Remote High Output Queue per second.
N	RemoteHighOutQ Now Messages Out/sec	Total number of messages removed from the Remote High Output Queue per second.
N	RemoteHighOutQ Now Bytes In/sec	Total number of bytes added for all messages to the Remote High Output Queue per second.
N	RemoteHighOutQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the Remote High Output Queue per second.
N	RemoteHighOutQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the Remote High Output Queue per second.
N	RemoteHighOutQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the Remote High Output Queue.
N	RemoteHighOutQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the Remote High Output Queue will experience.
N	RemoteMedOutQ Current Depth	Current number of messages in the Remote Medium Output Queue.

N	RemoteMedOutQ Now Messages In/sec	Total number of messages added to the Remote Medium Output Queue per second.
N	RemoteMedOutQ Now Messages Out/sec	Total number of messages removed from the Remote Medium Output Queue per second.
N	RemoteMedOutQ Now Bytes In/sec	Total number of bytes added for all messages to the Remote Medium Output Queue per second.
N	RemoteMedOutQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the Remote Medium Output Queue per second.
N	RemoteMedOutQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the Remote Medium Output Queue per second.
N	RemoteMedOutQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the Remote Medium Output Queue.
N	RemoteMedOutQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the Remote Medium Output Queue will experience.
N	RemoteLowOutQ Current Depth	Current number of messages in the Remote Low Output Queue.
N	RemoteLowOutQ Now Messages In/sec	Total number of messages added to the Remote Low Output Queue per second.
N	RemoteLowOutQ Now Messages Out/sec	Total number of messages removed from the Remote Low Output Queue per second.
N	RemoteLowOutQ Now Bytes In/sec	Total number of bytes added for all messages to the Remote Low Output Queue per second.
N	RemoteLowOutQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the Remote Low Output Queue per second.
N	RemoteLowOutQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the Remote Low Output Queue per second.
N	RemoteLowOutQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the Remote Low Output Queue.
N	RemoteLowOutQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the Remote Low Output Queue will experience.
N	LocalHighOrderQ Current Depth	Current number of messages in the Local High Order Queue.
N	LocalHighOrderQ Now Messages In/sec	Total number of messages added to the Local High Order Queue per second.
N	LocalHighOrderQ Now Messages Out/sec	Total number of messages removed from the Local High Order Queue per second.
N	LocalHighOrderQ Now Bytes In/sec	Total number of bytes added for all messages to the Local High Order Queue per second.
N	LocalHighOrderQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the Local High Order Queue per second.

N	LocalHighOrderQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the Local High Order Queue per second.
N	LocalHighOrderQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the Local High Order Queue.
N	LocalHighOrderQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the Local High Order Queue will experience.
N	LocalMedOrderQ Current Depth	Current number of messages in the Local Medium Order Queue.
N	LocalMedOrderQ Now Messages In/sec	Total number of messages added to the Local Medium Order Queue per second.
N	LocalMedOrderQ Now Messages Out/sec	Total number of messages removed from the Local Medium Order Queue per second.
N	LocalMedOrderQ Now Bytes In/sec	Total number of bytes added for all messages to the Local Medium Order Queue per second.
N	LocalMedOrderQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the Local Medium Order Queue per second.
N	LocalMedOrderQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the Local Medium Order Queue per second.
N	LocalMedOrderQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the Local Medium Order Queue.
N	LocalMedOrderQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the Local Medium Order Queue will experience.
N	LocalLowOrderQ Current Depth	Current number of messages in the Local Low Order Queue.
N	LocalLowOrderQ Now Messages In/sec	Total number of messages added to the Local Low Order Queue per second.
N	LocalLowOrderQ Now Messages Out/sec	Total number of messages removed from the Local Low Order Queue per second.
N	LocalLowOrderQ Now Bytes In/sec	Total number of bytes added for all messages to the Local Low Order Queue per second.
N	LocalLowOrderQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the Local Low Order Queue per second.
N	LocalLowOrderQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the Local Low Order Queue per second.
N	LocalLowOrderQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the Local Low Order Queue.
N	LocalLowOrderQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the Local Low Order Queue will experience.

N	RemoteHighOrderQ Current Depth	Current number of messages in the Remote High Order Queue.
N	RemoteHighOrderQ Now Messages In/sec	Total number of messages added to the Remote High Order Queue per second.
N	RemoteHighOrderQ Now Messages Out/sec	Total number of messages removed from the Remote High Order Queue per second.
N	RemoteHighOrderQ Now Bytes In/sec	Total number of bytes added for all messages to the Remote High Order Queue per second.
N	RemoteHighOrderQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the Remote High Order Queue per second.
N	RemoteHighOrderQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the Remote High Order Queue per second.
N	RemoteHighOrderQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the Remote High Order Queue.
N	RemoteHighOrderQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the Remote High Order Queue will experience.
N	RemoteMedOrderQ Current Depth	Current number of messages in the Remote Medium Order Queue.
N	RemoteMedOrderQ Now Messages In/sec	Total number of messages added to the Remote Medium Order Queue per second.
N	RemoteMedOrderQ Now Messages Out/sec	Total number of messages removed from the Remote Medium Order Queue per second.
N	RemoteMedOrderQ Now Bytes In/sec	Total number of bytes added for all messages to the Remote Medium Order Queue per second.
N	RemoteMedOrderQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the Remote Medium Order Queue per second.
N	RemoteMedOrderQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the Remote Medium Order Queue per second.
N	RemoteMedOrderQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the Remote Medium Order Queue.
N	RemoteMedOrderQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the Remote Medium Order Queue will experience.
N	RemoteLowOrderQ Current Depth	Current number of messages in the Remote Low Order Queue.
N	RemoteLowOrderQ Now Messages In/sec	Total number of messages added to the Remote Low Order Queue per second.
N	RemoteLowOrderQ Now Messages Out/sec	Total number of messages removed from the Remote Low Order Queue per second.
N	RemoteLowOrderQ Now Bytes In/sec	Total number of bytes added for all messages to the Remote Low Order Queue per second.

N	RemoteLowOrderQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the Remote Low Order Queue per second.
N	RemoteLowOrderQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the Remote Low Order Queue per second.
N	RemoteLowOrderQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the Remote Low Order Queue.
N	RemoteLowOrderQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the Remote Low Order Queue will experience.
N	TDHighQ Current Depth	Current number of messages in the Timed Delivery High Queue.
N	TDHighQ Now Messages In/sec	Total number of messages added to the Timed Delivery High Queue per second.
N	TDHighQ Now Messages Out/sec	Total number of messages removed from the Timed Delivery High Queue per second.
N	TDHighQ Now Bytes In/sec	Total number of bytes added for all messages to the Timed Delivery High Queue per second.
N	TDHighQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the Timed Delivery High Queue per second.
N	TDHighQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the Timed Delivery High Queue per second.
N	TDHighQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the Timed Delivery High Queue.
N	TDHighQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the Timed Delivery High Queue will experience.
N	TDMedQ Current Depth	Current number of messages in the Timed Delivery Medium Queue.
N	TDMedQ Now Messages In/sec	Total number of messages added to the Timed Delivery Medium Queue per second.
N	TDMedQ Now Messages Out/sec	Total number of messages removed from the Timed Delivery Medium Queue per second.
N	TDMedQ Now Bytes In/sec	Total number of bytes added for all messages to the Timed Delivery Medium Queue per second.
N	TDMedQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the Timed Delivery Medium Queue per second.
N	TDMedQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the Timed Delivery Medium Queue per second.
N	TDMedQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the Timed Delivery Medium Queue.

N	TDMedQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the Timed Delivery Medium Queue will experience.
N	TDLowQ Current Depth	Current number of messages in the Timed Delivery Low Queue.
N	TDLowQ Now Messages In/sec	Total number of messages added to the Timed Delivery Low Queue per second.
N	TDLowQ Now Messages Out/sec	Total number of messages removed from the Timed Delivery Low Queue per second.
N	TDLowQ Now Bytes In/sec	Total number of bytes added for all messages to the Timed Delivery Low Queue per second.
N	TDLowQ Now Bytes Out/sec	Total number of bytes removed for all the messages from the Timed Delivery Low Queue per second.
N	TDLowQ Now Traffic Intensity	Ratio of the number of messages added to the number of messages removed from the Timed Delivery Low Queue per second.
N	TDLowQ Avg. Queue Response Time [ms]	Average time in milliseconds a message waits in the Timed Delivery Low Queue.
N	TDLowQ 90% Queue Response Time [ms]	The response time in milliseconds that 90% of all messages passing through the Timed Delivery Low Queue will experience.
N	Output Waits	Total number of times output from critical client (Route or OPC) waited for ACK from MDS peer.
N	Average Output Wait Time	Average number of milliseconds MDS output waits to receive an ACK message from MDS peer.
N	Private Net Min RTT	Minimum time it took MDS to send a message over the private network and receive an ACK response from MDS peer.
N	Private Net Avg RTT	Average time it took MDS to send a message over the private network and receive an ACK response from MDS peer.
N	Private Net Max RTT	Maximum time it took MDS to send a message over the private network and receive an ACK response from MDS peer.

8.4.13 QoS

Table 8-23: Cisco ICM QoS

Performance Object: Cisco ICM QoS		
Counter Instance: “{Instance Name}”		
Always ON?	Counter Name	Description
N	High BytesSent/sec	High BytesSent/sec is the number of bytes per second sent to the other side over high priority connection.
N	High MsgsSent/sec	High MsgsSent/sec is the number of messages sent to the other side over high priority connection.
N	High BytesRcvd/sec	High BytesRcvd/sec is the number of bytes received from the other side over high priority connection.
N	High MsgsRcvd/sec	High MsgsRcvd/sec is the number of messages received from the other side over high priority connection.
N	High LocalRttMean	High LocalRttMean is the mean Round Trip Time in milliseconds of high priority messages as measured by local node.
N	High LocalRttStdDev	High LocalRttStdDev is the standard deviation of Round Trip Time of high priority messages as measured by local node.
N	High RemoteRttMean	High RemoteRttMean is the mean Round Trip Time in milliseconds of high priority messages as measured by remote node.
N	High RemoteRttStdDev	High RemoteRttStdDev is the standard deviation of Round Trip Time of high priority messages as measured by remote node.
N	High Xmit NowQueueDepth	High Xmit NowQueueDepth is the current number of messages in the transmit queue for high priority traffic.
N	High Xmit MaxQueueDepth	High Xmit MaxQueueDepth is the maximum number of message observed in the transmit queue for high priority traffic.
N	High Xmit NowBytesQueued	High Xmit NowBytesQueued is the current number of bytes in the retransmit queue for high priority traffic.
N	High Xmit MaxBytesQueued	High Xmit MaxBytesQueued is the maximum number of bytes observed in the retransmit queue for high priority traffic.
N	High TotalQoSReallocations	High TotalQoSReallocations is the total number of times QoS resources had to be reallocated for high priority connection because actual usage has exceeded previous allocation over defined threshold levels.
N	Med BytesSent/sec	Med BytesSent/sec is the number of bytes per second sent to the other side over medium priority connection.

N	Med MsgsSent/sec	Med MsgsSent/sec is the number of messages sent to the other side over medium priority connection.
N	Med BytesRcvd/sec	Med BytesRcvd/sec is the number of bytes received from the other side over medium priority connection.
N	Med MsgsRcvd/sec	Med MsgsRcvd/sec is the number of messages received from the other side over medium priority connection.
N	Med LocalRttMean	Med LocalRttMean is the mean Round Trip Time in milliseconds of medium priority messages as measured by local node.
N	Med LocalRttStdDev	Med LocalRttStdDev is the standard deviation of Round Trip Time of medium priority messages as measured by local node.
N	Med RemoteRttMean	Med RemoteRttMean is the mean Round Trip Time in milliseconds of medium priority messages as measured by remote node.
N	Med RemoteRttStdDev	Med RemoteRttStdDev is the standard deviation of Round Trip Time of medium priority messages as measured by remote node.
N	Med Xmit NowQueueDepth	Med Xmit NowQueueDepth is the current number of messages in the transmit queue for medium priority traffic.
N	Med Xmit MaxQueueDepth	Med Xmit MaxQueueDepth is the maximum number of message observed in the transmit queue for medium priority traffic.
N	Med Xmit NowBytesQueued	Med Xmit NowBytesQueued is the current number of bytes in the retransmit queue for medium priority traffic.
N	Med Xmit MaxBytesQueued	Med Xmit MaxBytesQueued is the maximum number of bytes observed in the retransmit queue for medium priority traffic.
N	Med TotalQoSReallocations	Med TotalQoSReallocations is the total number of times QoS resources had to be reallocated for medium priority connection because actual usage has exceeded previous allocation over defined threshold levels.
N	Low BytesSent/sec	Low BytesSent/sec is the number of bytes per second sent to the other side over low priority connection.
N	Low MsgsSent/sec	Low MsgsSent/sec is the number of messages sent to the other side over low priority connection.
N	Low BytesRcvd/sec	Low BytesRcvd/sec is the number of bytes received from the other side over low priority connection.
N	Low MsgsRcvd/sec	Low MsgsRcvd/sec is the number of messages received from the other side over low priority connection.
N	Low LocalRttMean	Low LocalRttMean is the mean Round Trip Time in milliseconds of low priority messages as measured by local node.

N	Low LocalRttStdDev	Low LocalRttStdDev is the standard deviation of Round Trip Time of low priority messages as measured by local node.
N	Low RemoteRttMean	Low RemoteRttMean is the mean Round Trip Time in milliseconds of low priority messages as measured by remote node.
N	Low RemoteRttStdDev	Low RemoteRttStdDev is the standard deviation of Round Trip Time of low priority messages as measured by remote node.
N	Low Xmit NowQueueDepth	Low Xmit NowQueueDepth is the current number of messages in the transmit queue for low priority traffic.
N	Low Xmit MaxQueueDepth	Low Xmit MaxQueueDepth is the maximum number of message observed in the transmit queue for low priority traffic.
N	Low Xmit NowBytesQueued	Low Xmit NowBytesQueued is the current number of bytes in the retransmit queue for low priority traffic.
N	Low Xmit MaxBytesQueued	Low Xmit MaxBytesQueued is the maximum number of bytes observed in the retransmit queue for low priority traffic.
N	Low TotalQoSReallocations	Low TotalQoSReallocations is the total number of times QoS resources had to be reallocated for low priority connection because actual usage has exceeded previous allocation over defined threshold levels.

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?

Let’s look at a simple 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:

 - a. Hardware Level: resources on a single server
 - b. Component Level: resources associated with a single application or a single application component (e.g. Unified ICM/CC Router) on a multi-application or multi-component server
 - c. 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

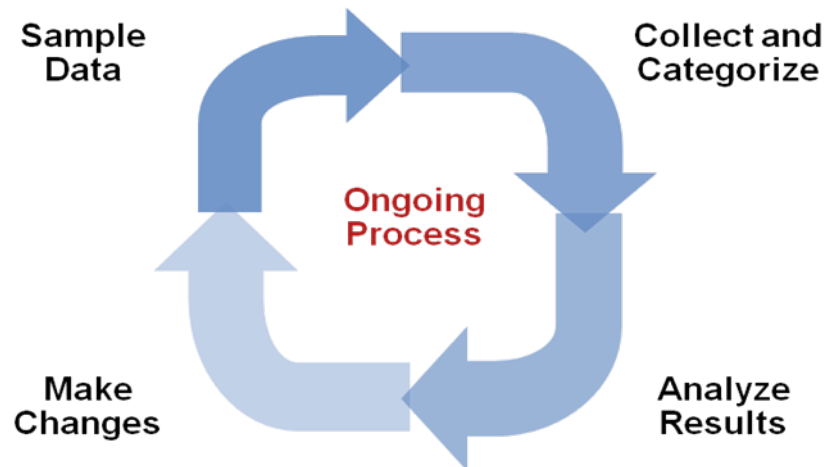


Figure 29: Capacity Planning Process

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 (e.g. Monday vs. Wednesday); busier days of the month (e.g. last business day of the month) or busier weeks of the year (e.g. 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:

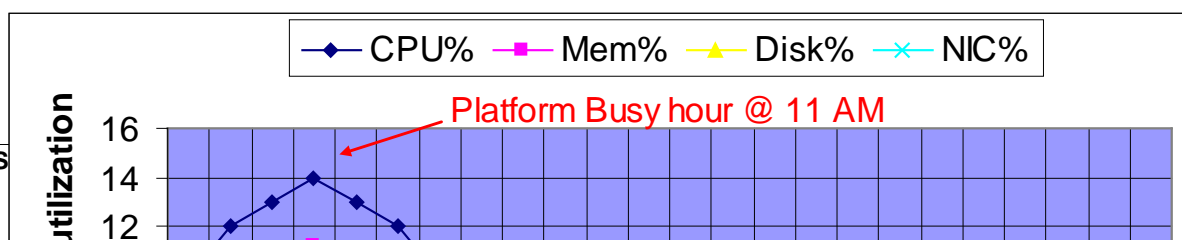


Figure 30: Graph of Samples to Find Busy Hour

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 Administration & Data Server
 - Clustering over WAN – any change to WAN characteristics

It's 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 (e.g. 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 Administration & Data Servers (real time feeds)
- Number of concurrent reporting users

9.3.3 Traffic Load

Examples of impacting traffic load changes:

- **Inbound call rate**

Let's say 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’s 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 95th 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 $\geq 100\%$ 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

Table 9-1: Calculating CPU Utilization

$\overline{CPU}_{\rho}(t_n) = \frac{CPU_{95\%}(t_n)}{CPU_{Sat}} * 100$	
CPU _{95%}	Measurement Counter: Processor – % Processor Time (_Total)
CPU _{Sat}	Maximum threshold: 60%
Indicator	Counter: System – Processor Queue Length Threshold: 2

9.4.2 Calculating Memory Utilization

Table 9-2: Calculating Memory Utilization

$Mem_{Sat} = Mem_{physical} * .8$ $\overline{Mem}_{\rho}(t_n) = \frac{Mem_{95\%}(t_n)}{Mem_{Sat}} * 100$	
Mem _{95%}	Measurement Counter: Memory – Committed Bytes
Mem _{Sat}	Threshold: 80% (of physical memory)
Indicator Counters	Counter: Memory – Available Mbytes Threshold: < 20% Counter: Memory – Memory – Pages / sec Threshold: 20% Counter: Paging File – % Usage Threshold: 80%

9.4.3 Calculating Disk Utilization

Table 9-3: Calculating Disk Utilization

$\overline{Disk}_{\rho}(t_n) = \frac{DT_{95\%}(t_n)}{DT_{Sat}} * 100$	
DT _{95%}	Measurement Counter: Processor – % Processor Time (_Total)
DT _{Sat}	Maximum threshold: 50%
Indicator	Counter: Physical Disk – Avg. Disk Queue Length Threshold: 1.5

9.4.4 Calculating NIC Utilization

Table 9-4: Calculating NIC Utilization

$NIC_{Sat} = NIC_{physical} * .03 \quad \overline{NIC}_{\rho}(t_n) = \frac{NIC_{95\%}(t_n)}{NIC_{Sat}} * 100$	
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:

$$\overline{UTIL}_{\rho} = \text{MAX}(\overline{CPU}_{\rho}[t], \overline{Mem}_{\rho}[t], \overline{Disk}_{\rho}[t], \overline{NIC}_{\rho}[t])$$

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 Unified ICM/CC Diagnostic Tools

10.1 Diagnostic Framework

10.1.1 Overview

Beginning with the 8.0 release, Unified ICM/CCE/CCH servers have implemented a new web-based service called the Diagnostic Framework, which is used to collect (and sometimes set) diagnostic information for that server. The Diagnostic Framework service is a REST-like service that accepts requests over HTTPS, gathers information from the system, and responds in the form of an XML response message. It can collect a variety of data, such as process logs, current trace values, network status, perfmon values, etc. The service can also be used to collect log files from the server. For a complete list of the capabilities, see the Diagnostic Framework API section, 10.1.5.

You can use the Diagnostic Framework in four (4) ways. They are briefly listed here, and described in more detail, in section 10.1.4

1. For Unified CCE deployments, the primary access method will be through the Analysis Manager, which serves as a solution-wide serviceability portal.
2. Unified CCE deployments can also use the Unified Communication diagnostic clients' command line interface (CLI).
3. Each Diagnostic Framework service also includes an HTML-based web user interface that provides access to the complete list of the API commands.
4. The API can also be accessed directly through a browser.

For more information on how to access the service, see the Usage section, 10.1.4.

10.1.2 Installation and Configuration

The Diagnostic Framework service is installed as part of the ICM/CCE/CCH release 8.0 software by the ICM-CCE-CCH installer (henceforth, called the Unified ICM installer for short). No additional installation or configuration steps are required. You may optionally choose to customize the service if needed, such as change the port number, certificate, logging level, etc. as explained in the following sections.

10.1.2.1 Service Registration and Dependencies

Diagnostic Framework is a .NET based web service. It is registered in the Windows service control by the Unified ICM installer¹. The service files are laid down under the following folder:

```
<ICM_Drive>:\icm\serviceability\diagnostics
```

The Diagnostic Framework service can be started or stopped from the Windows service control panel.

The service is registered under the following name:

```
Cisco ICM Diagnostic Framework
```

The Diagnostic Framework is hosted on top of the HTTP service built in the Windows Server 2003 kernel. It does not require IIS or any other web server to be installed. The Diagnostic Framework utilizes the Windows HTTP SSL service to provide secure

¹ The Unified ICM installer detects and installs the appropriate .NET version.

communications between the server and the client. Therefore, you must enable the HTTP SSL service before starting the Diagnostic Framework service. The Unified ICM installer configures this dependency in the Windows service control panel to automatically start the HTTP SSL service when the Diagnostic Framework service is started.

Note: The Diagnostic Framework or HTTP SSL service does not require IIS. However, if IIS is installed, the HTTP SSL service adds a dependency on the IIS service. Therefore, In order for HTTP SSL and the Diagnostic Framework to work, you must start IIS.

10.1.2.2 Service Port Configuration

The Diagnostic Framework listens on TCP port 7890.

If needed, you can change the port number. To change the port number you must update the Diagnostic Framework service configuration file and the certificate registration with Windows. You must change the port number on the CLI and Analysis Manager clients too. Additionally, you will need to change the port number on every other Unified ICM server where other instances of the Diagnostics Framework are running.

Note: Consider changing the port number only if absolutely necessary. To change the TCP port, follow these steps:

1. Stop the Diagnostic Framework service through the Windows service control.
2. Open a command prompt and change directory to
`<ICM_Drive>:\icm\serviceability\diagnostics\bin`
3. Run the following command and confirm from the output that the certificate binding with the current port is valid.
`DiagFwCertMgr /task:ValidateCertBinding`

Tip: To learn more about the *DiagFwCertMgr* utility, see the Certificate Management section, 10.1.3.3.

4. Note down the thumbprint of the certificate in use. You will need the thumbprint while registering the certificate with a different port, later. You can access it either from the output of the above command or from the following registry value:
`HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\Serviceability\DiagnosticFramework\CertUsedByDiagFwSvc`
5. In the same command window, run the following command to remove the certificate binding from the current port
`DiagFwCertMgr /task:UnbindCert`
6. Launch Notepad and open the service configuration file:
`<ICM_Drive>:\icm\serviceability\diagnostics\bin\DiagFwSvc.exe.config`
Tip: You may want to make a copy of this configuration file before making any changes to it.
7. Under Services section, locate the following statement and modify the port number printed after *localhost*: from 7890 to your desired number.

```
<add baseAddress="https://localhost:7890/icm-
dp/rest/AnalysisManager" />
```

Do not modify the syntax of the URL.

8. Save the file and quit Notepad.
9. Open a command prompt and change directory to

```
<ICM_Drive>:\icm\serviceability\diagnostics\bin
```
10. Run the following command to bind the certificate to the new port number:

```
DiagFwCertMgr/task:BindCertFromStore/certhash:<hash of the
certificate noted above>
```

The utility reads the port number from the service configuration file.
11. Read the output and confirm that the above command completed successfully.
 Optionally, run the `DiagFwCertMgr/task:ValidateCertBinding` command again to verify the changes to the port number binding.
12. Restart the Diagnostic Framework service.
13. If you have configured the Windows Firewall, you'll need to make sure the new port has been opened in the firewall configuration.

10.1.2.3 *Installing or updating 3rd Party Certificate*

During installation, the Diagnostic Framework generates a self signed certificate with its name set to the server hostname. The self signed certificate can be replaced with a trusted 3rd party signed certificate. For additional details, see the Certificate Management section, 10.1.3.3.

10.1.2.4 *Diagnostic Framework Log Files and Logging Level*

The Diagnostic Framework log files are created in the following folder:

```
<ICM_Drive>:\icm\serviceability\diagnostics\logs
```

The Diagnostic Framework uses the industry-standard log4net library to create and manage its log files. There is a configuration file which controls the names of the log files, how large they can get, how many rollover files are kept, the logging level, etc.

The default logging level is 'INFO', and it should be sufficient for most cases. You need not change the logging level unless directed by the TAC.

You can change the log level by editing the following file:

```
<ICM_Drive>:\icm\serviceability\diagnostics\config\log4net.
config
```

and changing the 'level' tag value to "DEBUG" (or "WARN", "ERROR", or "FATAL").

```
<root>
  <level value="INFO" />
  <appender-ref ref="RollingFileAppender" />
</root>
```

10.1.2.5 Diagnostic Framework Service Resources Requirements

10.1.2.5.1 Reduced Priority

The Diagnostic Framework service executes at a Below Normal priority so as to avoid adversely impacting server/application performance while running.

10.1.2.5.2 Changing Service CPU Threshold

Some CPU-intensive APIs of the Diagnostic Framework will first check the overall system CPU utilization value (%CPU), and will not start the request if the %CPU value is greater than a threshold value.

These APIs are:

- LogMgr commands
- TraceMgr commands
- ConfigMgr command

There are a few registry keys that control this behavior. Look in the following Windows Registry Key:

HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\Serviceability\DiagnosticFramework

And you will see the following:

Table 10-1: CPU Threshold

Registry key	Default Value	Description
CPUThresholdSample	5	To get a more accurate reading of the %CPU, multiple readings are taken. This value says how many samples should be read
CPUThresholdDelay	2	The number of milliseconds to wait between each sample taken
CPUThresholdPercent	60	The percent value to compare the current %CPU to. If the %CPU is greater than this value, the API will not start, and will return an error telling the user that the server is too busy, and to try the command later.

10.1.2.5.3 Changing Maximum Number of Concurrent Requests

The Diagnostic Framework service is designed to handle up to 20 concurrent web requests. The system has been tested under load to work with this configuration. However, due to special circumstances if there is a need to lower the number of concurrent requests, then you can modify the value of `maxConcurrentCalls` property in the service configuration file.

1. Stop the Diagnostic Framework service.
2. Launch Notepad and open the file:
`<ICM_Drive>:\icm\serviceability\diagnostics\bin\DiagFwSvc.exe.config`
Tip: You may want to make a copy of this configuration file prior to making any changes to it.
3. Locate the following property and change the value to any number below 20:
`<serviceThrottling maxConcurrentCalls="20" />`
4. Save the file and quit Notepad.
5. Restart the Diagnostic Framework service.

Caution: Do not increase the value beyond 20. It may lead to unexpected results during peak call volume.

10.1.3 Security

The Diagnostic Framework provides the infrastructure to establish a secure connection between the service and its clients. It uses HTTP basic authentication over SSL to authenticate, authorize and encrypt the connection. A valid Diagnostic Framework user account is required to access the service. Connections are not session oriented; the connection is maintained from the receipt of a request until the response is sent.

For service provider deployments, the Diagnostic Framework service is ICM instance aware, and can control access based on instance data requested.

10.1.3.1 Authentication, Authorization and Auditing

The Diagnostic Framework service integrates with Windows as well as Active Directory to provide user management and access control. The Diagnostic Framework allows two sets of users:

1. *A local Windows user who is a member of the local Windows security group called `ICMDiagnosticFrameworkUsers` on the server where the service exists:* This group is created by the Unified ICM installer and is initially empty; so by default, no local users have access to the service. The administrator on the server can make any local user a member of this group and provide access to Diagnostic Framework service. To add a user to the `ICMDiagnosticFrameworkUsers` group, use the Computer Management tool under Administrative Tools.
2. *A trusted domain user who is a member of the `CONFIG` domain security group of the `ICM/CCE/CCH` instance being accessed:* An `ICM/CCE/CCH SETUP` user or domain administrator can make any trusted user a member of the instance `CONFIG` group. Nested membership is allowed too; as a result the `SETUP` users and domain administrator also have access to the service. To add a user to the instance `CONFIG` group use the Active Directory Users and Computers tool or `ICM/CCE/CCH User List` tool. Access to domain users is configurable. By default, all direct and nested members of the `CONFIG`

group have access to the service. However, if needed access to domain users can be disabled as follows:

- a. Stop the Diagnostic Framework service.
- b. Launch Notepad and open the file:
`<ICM_Drive>:\icm\serviceability\diagnostics\bin\DiagFwSvc.exe.config`
Tip: You may want to make a copy of this configuration file prior to making any changes to it.
- c. Locate the following property and change the value from 1 to 0:
`<add key="DomainAuthorizationEnabled" value="1" />`
- d. Save the file and quit Notepad.
- e. Restart the Diagnostic Framework service.

Note: A Diagnostic Framework user does not require administrative privileges on the server to access the service.

The user authentication, validating username and password, is managed by Windows or Active Directory. Therefore, all valid or invalid login attempts are logged in the Windows Event Viewer (provided that logon/logoff auditing is enabled). The user authorization, validating group membership and optionally ICM instance access, is managed by the Diagnostic Framework service. Hence, all authorization requests can be audited through the Diagnostic Framework logs.

Note: A user may be a valid Windows or Active Directory user but may not be a member of the required security groups for access to Diagnostic Framework service. As a result, even though the user may pass authentication, it may not pass authorization.

Since the Diagnostic Framework user is managed by Windows or by Active Directory, the user is subjected to the password policies of the server or the domain. Always follow best practices and set strong password policies. For more information on system hardening and password policies refer to the Security Best Practices Guide for ICM/CCE/CCH 8.0.

10.1.3.1.1 Special Consideration for Servers with Multiple ICM Instances

This section applies to environments similar to service providers, who have multiple ICM instances on each server.

The domain user is authorized against the CONFIG domain security group of the ICM instance. If there are multiple instances on the server, then the service needs to know which instance security group to authorize against. *Therefore, on a multiple ICM instance server, the ICM instance name must be passed as one of the parameters for each request when authorizing a domain user.* If an instance name parameter is not passed then the domain user authorization will fail. The local user is free from this requirement since there is only one local group per server. Furthermore, when a domain user is used to access the service, the response is crafted only for the specific instance that user belongs to. However, when a local user tries to access the service, the response includes information for all instances on that server. This gives service providers flexibility to access control information collection for a one or all instances.

On a single instance server, the instance name is not required when accessing any API. Since there is only one instance on the server, the domain user is authorized against the CONFIG domain security group of that instance.

The table below summarizes the all authorization combinations. Remember that domain authorization can be completely disabled through the service configuration file.

Table 10-2: Domain Authorization Combination

ICM Instances on Server	User Type	Instance Name Provided	Authorization Criteria	Response Content on Successful Authorization
Multiple	Domain	No	Fail authorization, user must provide instance name in request	HTTP 403 – Access Forbidden
Multiple	Domain	Yes	Authorize against the instance name provided by user	Data for instance requested
Multiple	Local	No	Authorize against local group	Data for all instances
Multiple	Local	Yes	Authorize against local group	Data for instance requested
Single	Domain	No	Automatically detect the instance name and authorize against it.	Data for instance installed
Single	Domain	Yes	Authorize against the instance name provided by user. If the instance name is invalid then authorization will fail.	Data for instance installed
Single	Local	No	Authorize against local group	Data for instance installed
Single	Local	Yes	Authorize against local group	Data for instance installed

10.1.3.2 Encryption

Diagnostic Framework uses SSL to secure the HTTP connection between the server and the client. This secures both the credentials as well as data exchanged. To establish the SSL connection, a self-signed certificate is created by ICM-CCE-CCH installer and used during connection negotiation. Since the certificate is self signed, you will notice a warning issued by the browser about the invalidity of the certificate trust. Diagnostic Framework allows replacing the self signed certificate with a trust 3rd party certificate. See the Certificate Management section below for more details.

10.1.3.3 Certificate Management

The ICM-CCE-CCH installer creates a self signed certificate and stores it in the Windows Local Computer Personal certificate store with the friendly name “Cisco ICM Diagnostic Framework service certificate”. The installer then binds this certificate to the Windows HTTP service on the Diagnostic Framework service port, which by default is TCP 7890. Recall that Diagnostic Framework service is hosted on top of the Windows HTTP service.

Therefore, this certificate is used by Windows HTTP service to establish a secure HTTPS channel (HTTP over SSL) whenever the Diagnostic Framework service is accessed. The Unified ICM installer uses the Diagnostic Framework Certificate Manager Utility to create and bind the self signed certificate.

Depending on the nature of business and the network access layout of the site, a self signed certificate may provide sufficient security for accessing the service from within the trusted intranet. However, if you plan to access the service from outside the trusted network then it is highly recommended to replace the self signed certificate with a trusted 3rd party certificate to provide improved security².

When accessing the service with the self signed certificate for the first time from Internet Explorer, you will see a warning about the validity of the certificate. If you are certain that the server is authentic then you may choose to accept the certificate and store it on the client machine to avoid future warnings.

If you wish to replace the server certificate with a trusted 3rd party certificate or modify the port to which a certificate is bound, you **must** use the Diagnostic Framework Certificate Manager utility.

10.1.3.3.1 Diagnostic Framework Certificate Manager Utility

The Diagnostic Framework Certificate Manager utility is a command line utility used to manage certificate creation and binding for the Diagnostic Framework service. It is installed at:

```
<ICM_Drive>:\icm\serviceability\diagnostics\bin\DiagFwCertMgr.exe
```

The utility can perform the following tasks:

- Create self signed certificate.
- Store the certificate in Local Computer Personal certificate store.
- Bind a certificate to Windows HTTP service on a given port.
- Remove a certificate binding from the Windows HTTP service on a given port.
- Delete the self signed certificate created by itself from the Local Computer Personal certificate store.
- Validate the certificate binding to HTTP service for Diagnostic Framework service.

The following section explains the usage of the utility:

```
DiagFwCertMgr /task:<task_name>  
    [/port:<port_number>]  
    [/certhash:<certificate_thumbprint>]  
    [/logpath:<logfile_path>]
```

Where

/task: specifies the task to be performed.

² A self signed certificate cannot guarantee the authenticity of the hosting server. Since the client is unaware of the server authenticity, the client should exercise caution when sharing the user credentials with such server. A malicious user may setup a rogue server with a self signed certificate, claiming to be a legit server, and use it to steal user credentials from the client. Always use trusted certificates to authenticate servers when accessing outside your trusted network.

/port: specifies the port number used by the service; this is optional as the port number is automatically read from the service configuration file (DiagFwSvc.exe.config).

/certhash: specifies the SHA-1 thumbprint of the certificate; required only when binding a specific certificate, which exists in the certificate store, to a port.

/logpath: specifies the path where the log file should be created; by default it is the current folder.

The following table explains each task:

Table 10-3: Diagnostic Framework Certificate Manager Utility Tasks

Task	Description
CreateAndBindCert	Creates a self signed certificate in the local computer personal certificate store and binds it with HTTP service on the given port. [Used by ICM-CCE-CCH Install]
BindCertFromStore	Looks up the certificate provided by /certhash argument in certificate store and binds it with the HTTP service on the given port.
UnbindCert	Removes the certificate binding from the specified port, does not modify any certificate in the store
UnbindAndDeleteCert	Removes the certificate binding from the specified port. Also, deletes the self signed certificate created by CreateAndBindCert option. [Used by ICM-CCE-CCH Uninstall]
ValidateCertBinding	Verifies the certificate binding on the specified port and confirms its presence in the local computer certificate store.

Diagnostic Framework Certificate Manager utility stores the thumbprint (SHA-1 hash) of the self signed certificate created by the utility and the certificate used by the Diagnostic Framework service in the registry at the following location respectively:

HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\Serviceability\
DiagnosticFramework\SelfSignedCertCreatedForDiagFwSvc

HKLM\SOFTWARE\Cisco Systems, Inc.\ICM\Serviceability\
DiagnosticFramework\CertUsedByDiagFwSvc

Unless the certificate used by the service is changed manually, both registry values will be the same.

10.1.3.3.2 Using a Trusted Third Party Certificate

Replacing the certificate used by the Diagnostic Framework service involves two tasks, first to **import** the new certificate in the Local Computer Personal certificate store and second to **bind** it with the TCP port used by the service.

Import Certificate: Use the MMC Certificates snap-in to import a certificate in the Local Computer Personal certificate store as explained in the section “*Import the Certificate into the Local Computer Store*” of the Microsoft KB article 816794 – “HOW TO: Install Imported Certificates on a Web Server in Windows Server 2003”.

<http://support.microsoft.com/kb/816794>

Caution: Do not follow the instructions in the next section “*Assign the Imported Certificate to the Web Site*”. Diagnostic Framework does not use IIS web server. It is hosted on top of Windows HTTP service. You must use the DiagFwCertMgr utility to bind this certificate to the Windows HTTP service.

Bind Certificate: Follow the instructions below to bind the certificate added to the Windows HTTP service using the DiagFwCertMgr utility:

1. Open MMC Certificates snap-in and note down the thumbprint of the certificate that needs to be used with the Diagnostic Framework service.
2. Stop the Diagnostic Framework service via the Windows service control.
3. Open a command prompt and change directory to
`<ICM_Drive>:\icm\serviceability\diagnostics\bin`
4. In the command window, run the following command to remove the current certificate binding from the port:
`DiagFwCertMgr /task:UnbindCert`
5. Run the following command to bind the new certificate to the service:
`DiagFwCertMgr /task:BindCertFromStore /certhash:<hash of the certificate noted above>`
The utility reads the port number from the service configuration file.
6. Read the output and confirm that the above command completed successfully. Optionally, run the `DiagFwCertMgr /task:ValidateCertBinding` command to verify the changes to the certificate binding.
7. Restart the Diagnostic Framework service.

10.1.4 Usage

There are four ways to access the diagnostic data provided by the framework:

10.1.4.1 Accessing the Diagnostic Framework through the Analysis Manager

The Analysis Manager is part of the Real Time Monitoring client Tool (TROT) which resides on Unified CM. TROT is not a web-based tool, rather it is a thick client tool where you must

download from Unified CM and install on a server. TROT includes menus for the Analysis Manager. You can access the Analysis Manager functions from the tool. See the sample screen:

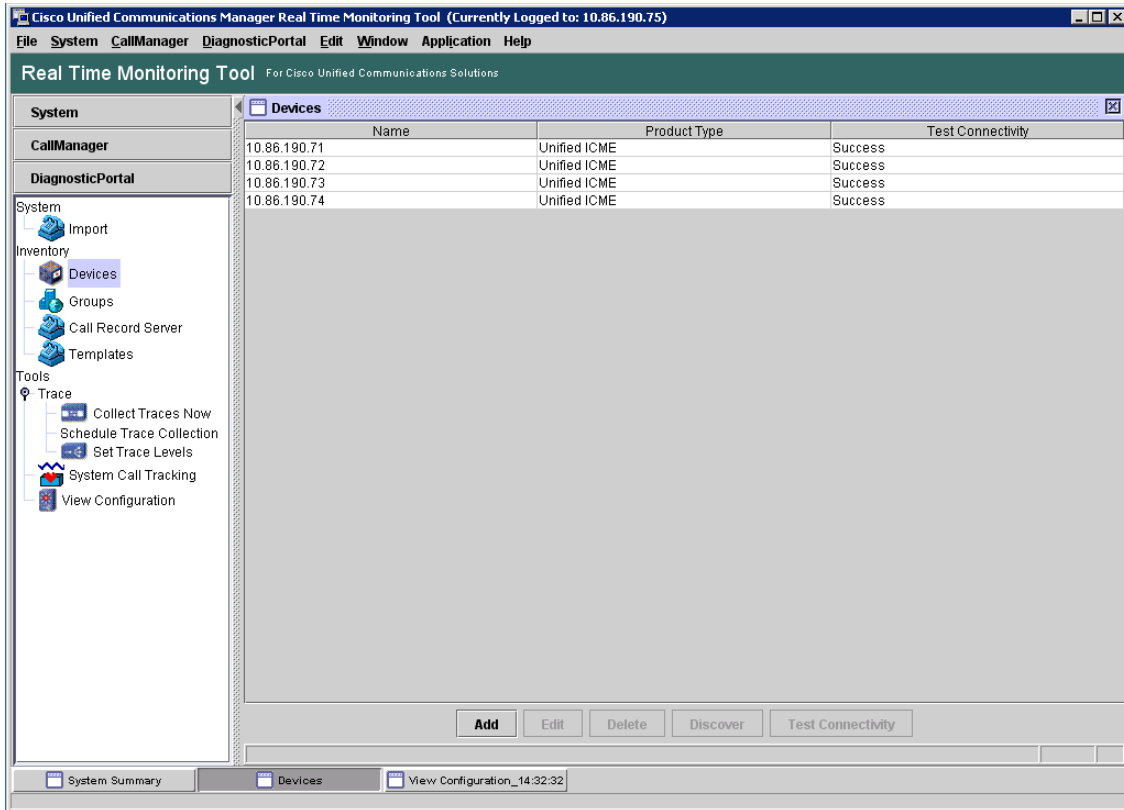


Figure 31: Real Time Monitoring Tool

For more information on how to use the Analysis Manager, see the *Cisco Unified Analysis Manager User Guide*.

10.1.4.2 Accessing the Diagnostic Framework through the Unified System CLI

The Diagnostic Framework can also be accessed through a Command Line Interface (CLI). The CLI access utility is installed on every Unified ICM machine at the following location:

```
<ICM_Drive>:\icm\serviceability\wsccli\runwsccli.bat
```

Use a DOS command shell to run this batch file, and it will setup everything needed to access the Diagnostic Framework through the CLI.

A shortcut is included to the Unified ICM menu to provide quick access to the CLI. Also, you can access Unified CLI from: **Start->Programs->Cisco Unified CCE Tools->Unified System CLI**. A new DOS Window opens with an initial prompt for your credentials (username and password).

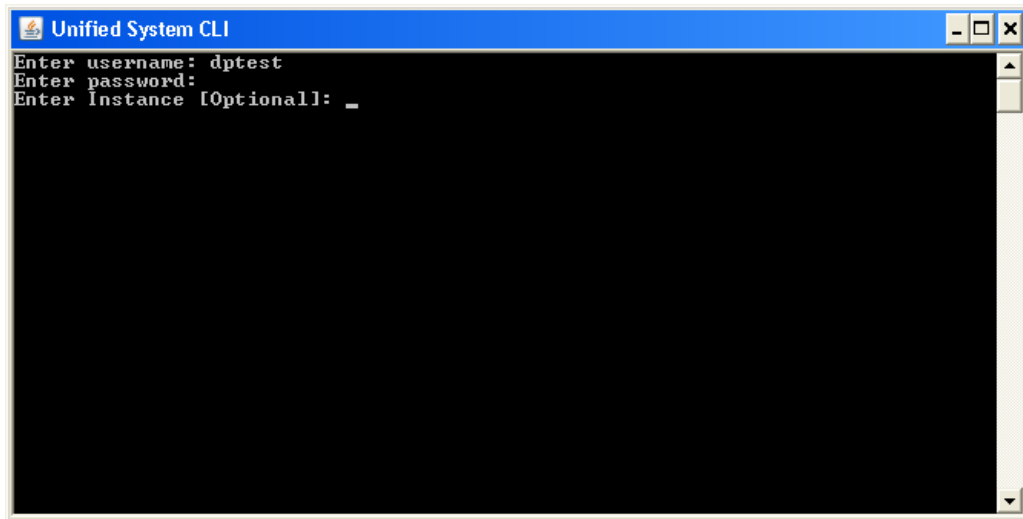


Figure 32: Using Unified System CLI from Command Prompt

On authentication, you can use the CLI from this window, as explained in the following section:

The CLI allows an optional user input named Instance. In Unified CCE environments, you don't need to enter anything. In a Hosted environment, you will need to enter the Instance in order to access the diagnostic data for that particular instance only. See the section (Special Consideration for Servers with Multiple ICM Instances) for additional details.

10.1.4.2.1 Unified CLI Architecture

Note: This picture is from a CVP perspective only, and doesn't specify the Diagnostic Framework directly. But the Diagnostic Framework is what ICM uses under-the-hood!

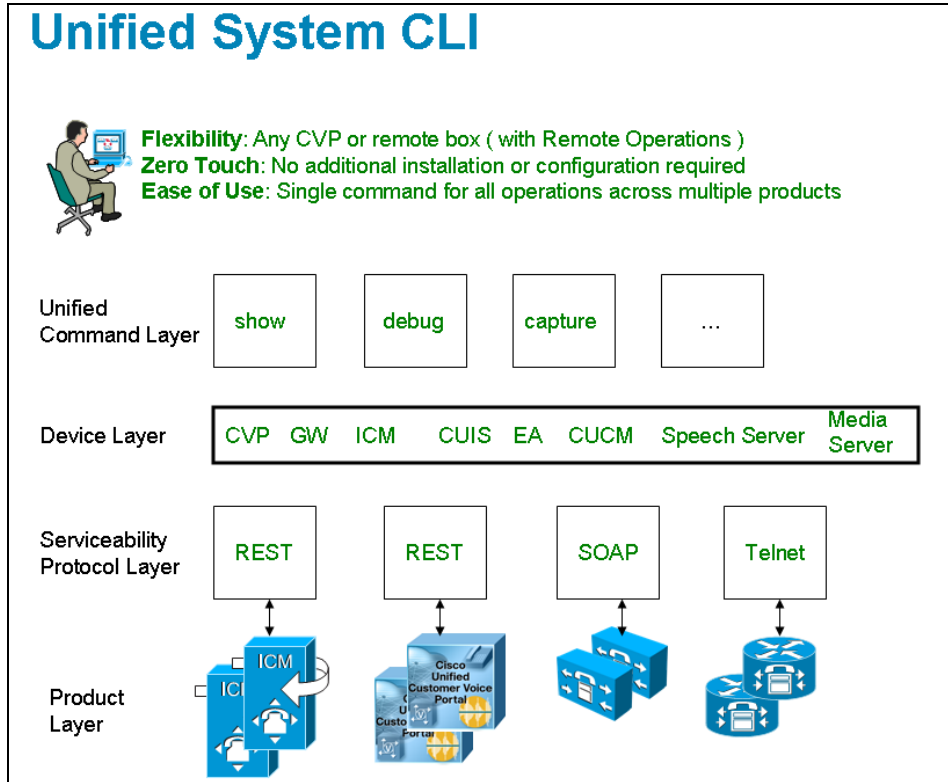


Figure 33: Unified CLI Architecture

An user can perform the following tasks using Unified CLI:

- Run a single command (in system mode) on any Unified ICM system to gather information about all supported solution components.
- In system mode, you can optionally provide the seed device(s) in WSC_CLI_DIR\conf directory or give a flat CSV file with a device list.
- System mode allows the CLI to recursively go to each supported box in the background and run the same command that was executed by the user in system mode. User can optionally limit the system command to be executed only on certain device group or list of servers. Device group is automatically populated based on device type (CVP, ICM, IOS, EA as an example), device IP/hostname wildcard (LOC-1*, 10.86.129.* as an example for branch office deployments), or the CSV file in WSC_CLI_DIR\conf directory.
- The system command can also be executed by prefixing the "system" on any regular command. For example, "system show all" or typing "system" and executing the commands exactly like a regular CLI for interactive mode.

10.1.4.2.2 Unified System CLI Usability

- System CLI is automatically installed on all Unified ICM systems as part of the infrastructure, so there is no additional installation required.
- System CLI can be executed as a Windows scheduled job or a Unix Cron job. Single command for all operations across multiple products and servers.
- All the commands available in non-system mode for a local system will also be available in system mode. The command syntax remains the same in system mode. There is an additional option to limit the system command option to certain device group, device type or list of servers.
- In system mode, when you seek help for using the "?" character after the keyword component or subcomponent is inputted, the list of components that outputs maybe quite large due to the fact that it is an aggregated list of all the possible component types on all the unique server types.
- The Master list is defined by the unique "Name", "ProductType". If there are multiple components for the purpose of co-location, the internal list will still contain one entry because there is only one WebServices manager running at the specified port.
- System CLI runs on a low priority, so it will only use the IDLE CPU on the System. It should not affect the Call Processing even if it gets executed on a system running under load. The response time varies depending on the load of the system you are running and the server response time. The response time when there is no running load should be below 5 seconds for each server for simple operations like "version", "license", "debug" and "perf". The response time when there is no running load for "platform" should be below 10 seconds for each server. However, the response time cannot be determined for command like "trace", "log", "sessions", "and all "tech-support" and that can vary depending on the data transferred by the server.
- There are no specific timeouts on the System CLI client and it is controlled by the server.
- Error code and error description during failure conditions occur from the server side. System CLI will display the error message arriving from server. The possible error codes are specified and described in the DP REST API specification.

10.1.4.2.3 Extensibility

- System CLI is not a tool but an extensible platform to build several analysis toolkits. The CLI library can be embedded or used within the analysis engine to do post processing of the data (normalized). System CLI can be used by common scripting tools like Perl etc to create custom logic

10.1.4.2.4 Command Syntax

The common CLI syntax matches closely with **Cisco IOS gateway** CLI commands. In cases where specific commands or parameters are not available in IOS gateway, the syntax attempts to match the Unified CM platform CLI commands for consistency.

The following tables list and describe the CLI commands that are available for the diagnostics purpose:

Notes:

1. If you do not specify component/sub-component, then the list includes all the installed components/sub-components on the server.
2. The command output on screen will not include binary data.

Table 10-4: CLI Commands

Command (Verb)	Noun	Description
show	all	Aggregation of output for all the supported nouns and specific to the verb "show".
show	component	Lists the currently installed components on the server.
show	configuration	Lists the application configuration.
show	debug	Shows the current debug levels.
show	license	Shows the license/port information.
show	log	Shows the logs.
show	perf	Shows the performance information.
show	platform	Shows the platform information.
show	sessions	Shows the current active sessions/calls. [Not supported by Unified ICM]
show	tech-support	Shows system information for Tech-Support Note: This command is exactly the same as "show all".
show	trace	Shows the traces.
show	version	Shows system hardware and software status and version.
show	devices	Shows information of devices that are known to the CLI.
debug	level	Sets the specific debug level.
help		Shows the help information
quit		Quits the CLI
capture		Captures the network packets [Not supported by Unified ICM]

There are more detailed examples at the following location:

<http://cvp.cisco.com/display/CVPDE/Common+CLI>

Note: The following features – filter and match -- of the CLI will not be supported for trace files because the framework is returning a zip file that contained not just the text file. For those two features, CLI is expecting a plain text file.

Following is the system mode syntax:

Note: There are product specific extensions that can be added, **however** any extensions **must** be reviewed by this common cross-product team for **clarity and consistency**.


Table 10-5: System Mode Syntax

Command (Verb)	Noun	Description
system		Enter the interactive system mode of the CLI. Use quit/exit command to exit the system mode.

The system command can also be executed by prefixing the "system" on any regular command for non-interactive mode. For example, "system show all"

Table 10-6: System Commands

Noun	Parameters and Options	Description
show all	<p>[component <i>component(s)</i>] [subcomponent <i>subcomponent(s)</i>] [filter <i>noun(s)</i>] [absdatetime <i>startdatetime enddatetime</i>] [reltime <value> minutes/hours/days/weeks/months] [match <string value>] [<output modifier>] [group <i>group(s)</i>] [server <i>server(s)</i>] [sysmatch <string value>] [devicetype <product type>]</p> <p>Note: The options highlighted in blue color above are included to command(s) in system mode.</p> <p>where</p> <p>Options</p> <ul style="list-style-type: none"> • group - Narrows the output to selected group(s) only. • server - Narrows the output to selected server(s) only. • sysmatch - Match a particular string as specified by <string value> <p>Note The command notifies about a possible impact to system performance and asks you if you want to continue.</p>	<p>Aggregation of output for all the supported nouns and specific to the verb "show".</p> <p>Example-1: admin:system admin(system):show all redirect dir c:\system-tech-support [server-1] server-1 show all Output [server-2] server-2 show all Output [server-3] server-3 show all Output [server-4] server-4 show all Output [server-5] server-5 show all Output [server-6] server-6 show all Output</p> <p>Output is saved to "c:\system-tech-support\clioutput0.zip"</p> <p>Example-2:</p>

Noun	Parameters and Options	Description
	<p style="text-align: center;">  Warning Because running this command can affect system performance, Cisco recommends that you run the command during off-peak hours. </p>	<p>Assuming Group:Branch-1 contains server-2, server-3 and Group:Branch-2 contains server-5, server-6</p> <pre>admin:system admin(system): show all group Branch1 Branch2 redirect dir c:\system-tech-support [server-2] server-2 show all Output [server-3] server-3 show all Output [server-5] server-5 show all Output [server-6] server-6 show all Output</pre> <p>Output is saved to "c:\system-tech-support\clioutput0.zip"</p> <p>Example-3:</p> <pre>admin:system admin(system):show all server server-1 server-6 redirect dir c:\system-tech-support [server-1] server-1 show all Output [server-6] server-6 show all Output</pre> <p>Output is saved to "c:\system-tech-support\clioutput0.zip"</p> <p>Example-4:</p>

Noun	Parameters and Options	Description
		<p>Assuming that server-2, server-3, server-5 are in subnet 10.86.129.xxx</p> <pre>admin:system admin(system):show all group Branch1 Branch2 sysmatch "10.86.129*" redirect dir c:\system-tech-support [server-2] server-2 show all Output [server-3] server-3 show all Output [server-5] server-5 show all Output</pre> <p>Output is saved to "c:\system-tech-support\clioutput0.zip"</p> <p>Example-5:</p> <pre>admin:system show all redirect ftp://vpalawat:password/SR609140000 [server-1] server-1 show all Output [server-2] server-2 show all Output [server-3] server-3 show all Output [server-4] server-4 show all Output [server-5] server-5 show all Output [server-6] server-6 show all Output</pre>

Noun	Parameters and Options	Description
		<p>Output is saved to "ftp-sj.cisco.com\incoming\SR609140000-0.zip"</p> <p>Output is saved to "ftp-sj.cisco.com\incoming\SR609140000-1.zip"</p> <p>Example-6:</p> <p>Assuming that devices configured in OAMP are CVP[server-5], IOS [server-2, server-3], UCM [server-4] and ICM [server-1] .</p> <pre>admin:system admin(system):show all devicetype cvp ios redirect dir c:\system-tech-support [server-2] server-2 show all Output of ProductType [ios] [server-3] server-3 show all Output of ProductType [ios] [server-5] server-5 show all Output of ProductType [cvp]</pre> <p>Output is saved to "c:\system-tech-support\clioutput0.zip"</p>

10.1.4.2.5 Automated Command Execution

CLI or System CLI commands can be executed automatically using the following mechanism:

- Create a batch file with the commands given below as an example

```
REM VERSION-COLLECTION
```

```
echo system show version redirect dir c:\test\ > clicmds.txt
echo exit >> clicmds.txt
type clicmds.txt | wsccli.bat inplace nointeractive
"user:wsmadmin" "passwd:<password>"
```

- To define a multiple component and sub-component filter, use double quotes as follows:

```
REM CONFIG-COLLECTION
echo show config comp CallServer subc "SIP|ICM" redirect dir
c:\test\ > clicmds.txt
echo exit >> clicmds.txt
type clicmds.txt | wsccli.bat inplace nointeractive
"user:wsmadmin" "passwd:<password>"
```

- Automated trace collection on CVP servers using a scheduled job:

```
REM TRACE-COLLECTION
echo show trace device cvp redirect dir c:\test\ > clicmds.txt
echo exit >> clicmds.txt
type clicmds.txt | wsccli.bat inplace nointeractive
"user:wsmadmin" "passwd:<password>"
```

- Automated script can be invoked from a Windows scheduled job for automated tasks.

Note: Because running the automated commands and non-interactive mode can affect system performance, Cisco recommends that you run the command during off-peak hours.

10.1.4.2.6 Import File Syntax

The file to be imported is named devices.csv located at:

```
<ICM_Drive>:\icm\serviceability\wsccli\conf
```

A sample file named devices-sample.csv is provided. Add the devices to this file, and then restart Unified System CLI to load those devices.

10.1.4.2.6.1 Devices CSV File Syntax

```
#####
# Sample CSV file for importing devices. File name should be devices.csv
# The file should be located at WSC_CLI_DIR/conf folder
#
# The possible values for Product Type are given below:
#
# * UCM      - For Unified CM
# * CVP      - For Unified CVP
# * ICM      - For Unified ICME, ICM
# * UCCX     - For Unified CCX
# * IOS      - For IOS Gateway
# * EA       - For Unified Expert Advisor
# * CUIC     - For Unified IC
# * CUP      - For Unified Presence ( that includes the SIP Proxy )
#####
#
# The column assignments are as follows:
#
```

```
# HOSTNAME          -- Mandatory
# DESCRIPTION
# PRODUCT_TYPE     -- Mandatory
# GROUP
# USERNAME
# PASSWORD
# PORT_NUMBER      -- Mandatory
# ENABLE_PASSWORD
# IS_SEED_SERVER
#
HOSTNAME, DESCRIPTION, PRODUCT_TYPE, GROUP, USERNAME, PASSWORD,
PORT_NUMBER, ENABLE_PASSWORD, IS_SEED_SERVER
#10.86.129.109, IOS GW, IOS, Location_1, cisco, cisco, 23, cisco,,
```

10.1.4.2.7 Device, Protocol and Command Mapping Table

Following is the mapping table for device type, command and serviceability protocol created in WSC_CLI_DIR/conf folder:

Table 10-7: Device, Protocol and Command Mapping

	CVP	ICM	EA	CUIS	Speech Server	Media Server	Trace Server	IOS GW	CUCM	UCCX
capture	REST	✘	✘	✘	REST	REST	REST	✘	✘	✘
config	REST	REST	REST	REST	✘	✘	✘	TELNET	✘	✘
debug	REST	REST	REST	REST	✘	✘	✘	TELNET	SOAP	REST
license	REST	REST	REST	REST	?	✘	✘	TELNET	SOAP	REST
log	REST	REST	✘	✘	REST	REST	REST	✘	✘	✘
perf	REST	REST	✘	✘	REST	REST	REST	TELNET	✘	✘
platform	REST	REST	SOAP	SOAP	REST	REST	REST	TELNET	SOAP	SOAP
sessions	REST	✘	✘	✘	✘	✘	✘	TELNET	✘	✘
trace	REST	REST	SOAP REST	SOAP REST	?	?	?	TELNET	SOAP	SOAP REST
version	REST	REST	SOAP REST	SOAP REST	REST	REST	REST	TELNET	SOAP	SOAP REST

✘ -- Not supported ? -- Unknown

- CLI will have the master list of all devices from seed server(s). It will run the system command on each device recursively based on the protocol supported in this release and according to the mapping table given above.
- Master list is defined by the unique "Name", "ProductType". If there are multiple devices for the purpose of co-location, the internal list will still contain one entry for a product

type because there is only one WebServices manager running at the specified port.

- CLI will also pull the component/sub-component list from all the devices to create a master list dynamically.
- The CLI output will be in the structure of [Server]/[Type]/clioutput . There will be a single (or multiple zip in case exceeding the size of zip file of 1GB) zip file created for the aggregate response from all servers.

10.1.4.2.8 Mapping of System CLI commands to IOS CLI commands

Table 10-8: Mapping of System CLI commands to IOS CLI commands

System CLI	IOS CLI	Notes
"show config"	"show running-config"	
"show version"	"show version"	
"show license"	"show license"	
"show perf"	"show call resource voice stat" "show memory statistics" "show processes cpu history" "show processes memory sorted" "show voice dsp group all" "show voice dsp voice"	
"show debug"	"show debug"	
"show log"	N/A	
"show sessions"	"show call active voice compact"	
"show tech-support"	"show tech-support" <Everything else given above>	
"show trace"	"show logging"	
"show platform"	"show diag"	
"debug"	0 no debug all 1 - deb ccsip err deb cch323 err deb voip app vxml err deb http client err deb mrcp err deb rtsp err deb h225 asn1 err deb h245 asn1 err 2 - debug isdn q931 debug h225 events debug h245 events debug voip ccapi inout	

	debug vtsp events 3 - debug ccsip messages debug h225 q931 debug h225 asn1 debug h245 asn1	
--	---	--

Note: This mapping table is available in the configuration file, so that mapping can be altered easily.

10.1.4.2.9 Logs

All logs generated by the CLI process will be under the following directory:

```
<ICM_Drive>:\icm\serviceability\wsccli
```

10.1.4.3 Accessing the Diagnostic Framework via the built-in User Interface

In order for an end user to easily harness the functionality of the Diagnostic Framework, a built-in, web-based menu utility will allow a user to interact with the framework through their browser. The single API command, GetMenu, will generate an HTML page which can be used to interactively create framework requests and view their replies from the Diagnostic Framework in the same page for the specified server.

Users who do not have access to the Analysis Manager will be able to use this command to gather data from the Diagnostic Framework, without having to know all of the API URLs and parameter values. The GetMenu command will recognize and support machines with multiple instances [Hosted environment] installed. Since this GetMenu command is built directly into the Diagnostic Framework, no special client side files or installations are needed in order to access it. The command will be accessible from any machine with a compatible browser (i.e. Internet Explorer).

The entry point for the menu utility is through the GetMenu command within the Diagnostic Framework. Following is an example request:

https://<UCCE-server>:<port>/icm-dp/rest/AnalysisManager/GetMenu

Where <UCCE-server> is the host name or IP address of the desired server, and <port> is the access port (usually 7890).

Following is a sample screen:

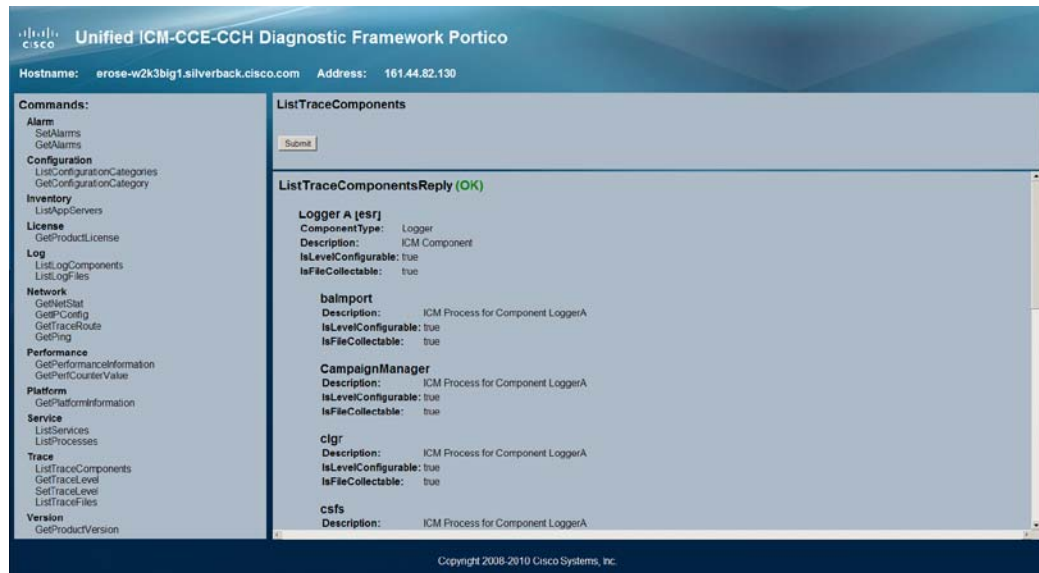


Figure 34: Unified ICM-CCE-CCH Diagnostic Framework

Most of the commands will return simple XML data; this menu utility will do some XML parsing and display the results. A few of these commands will create links to allow the user to download the returned files.

Note: For DownloadTraceFiles, you may download the zip file only once, because the file is unique for each request. Once the file is downloaded, the link will be disabled and the zip file will be deleted from the server. However, you may simply do another request with the same parameters; it will generate another zip file with exactly the same contents but with a different file name.

10.1.4.4 Accessing the Diagnostic Framework through a Browser

Since the Diagnostic Framework is a XML/HTTP based REST-style RPC referred as “RPC-Hybrid” interface, it is possible to access the Diagnostic Framework commands directly through a browser (Internet Explorer). To access the commands from a browser, type the full URL of the desired command, at the browser address location.

For example, the following URL:

<https://<UCCE-Server>:<port>/icm-dp/rest/AnalysisManager/GetTraceLevel?Component=Component/Subcomponent>

The IE browser will display the data in XML or may ask you to save the file if you are downloading the file. See API section for more examples of the URL.

The complication with this technique is that there are many APIs, and many of them contain various parameters which need to be specified properly.

10.1.5 Diagnostic Framework API

The Diagnostic Interface supports the following commands:

10.1.5.1 *GetTraceLevel*

The Diagnostic Framework supports four levels of trace configuration based on level of trace detail and performance impact; the Diagnostic Framework translates the following levels to component- or process-specific trace level settings:

Table 10-9: Trace Levels

Trace Level	Description
0	Product/component install default, should have no/minimal performance impact
1	Less detailed trace messages, small performance impact
2	More detailed trace messages, medium performance impact
3	If the trace level does not match any pre-defined levels (i.e., a manually configured, specific trace mask), Diagnostic Framework returns "custom (99)".

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/GetTraceLevel?Component=Component/Subcomponent>

Reply example:

```
<?xml version="1.0" encoding="UTF-8"?>
<dp:GetTraceLevelReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/analysismanager">
<dp:Schema Version="1.0"/>
<dp:Trace Level="0"/>
</dp:GetTraceLevelReply>
```

10.1.5.2 *SetTraceLevel*

Please see *GetTraceLevel* above for details about the actual trace level values.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/SetTraceLevel?Component=Component/Subcomponent&Level=1>

Reply example:

```
<?xml version="1.0" encoding="UTF-8"?>
<dp:SetTraceLevelReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/analysismanager">
<dp:Schema Version="1.0"/>
</dp:SetTraceLevelReply>
```

10.1.5.3 *ListTraceComponents*

Lists all possible application components that produce trace files.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/ListTraceComponents>

Reply example:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```

<dp:ListTraceComponentsReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/analysismanager">
<dp:Schema Version="1.0" />
<dp:TraceComponentList>
<dp:TraceComponent Name="Logger A" ComponentType="Logger" Description="ICM
Component" IsLevelConfigurable="true" IsFileCollectable="true">
  <dp:TraceComponentList>
    <dp:TraceComponent Name="baImport" Description="ICM Process for Component
LoggerA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="CampaignManager" Description="ICM Process for
Component LoggerA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="clgr" Description="ICM Process for Component
LoggerA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="csfs" Description="ICM Process for Component
LoggerA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="cw2kFeed" Description="ICM Process for Component
LoggerA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="dtp" Description="ICM Process for Component
LoggerA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="hlgr" Description="ICM Process for Component
LoggerA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="nm" Description="ICM Process for Component
LoggerA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="nmm" Description="ICM Process for Component
LoggerA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="rcv" Description="ICM Process for Component
LoggerA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="rpl" Description="ICM Process for Component
LoggerA" IsLevelConfigurable="true" IsFileCollectable="true" />
  </dp:TraceComponentList>
</dp:TraceComponent>
<dp:TraceComponent Name="Router A" ComponentType="Router" Description="ICM
Component" IsLevelConfigurable="true" IsFileCollectable="true">
  <dp:TraceComponentList>
    <dp:TraceComponent Name="agi" Description="ICM Process for Component
RouterA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="ccag" Description="ICM Process for Component
RouterA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="dba" Description="ICM Process for Component
RouterA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="dbw" Description="ICM Process for Component
RouterA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="mds" Description="ICM Process for Component
RouterA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="nm" Description="ICM Process for Component
RouterA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="nmm" Description="ICM Process for Component
RouterA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="nms" Description="ICM Process for Component
RouterA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="rtr" Description="ICM Process for Component
RouterA" IsLevelConfigurable="true" IsFileCollectable="true" />
    <dp:TraceComponent Name="rts" Description="ICM Process for Component
RouterA" IsLevelConfigurable="true" IsFileCollectable="true" />
  </dp:TraceComponentList>
</dp:TraceComponent>
<dp:TraceComponent Name="Cisco ICM Diagnostic Framework" Description="Cisco
ICM Diagnostic Framework" IsLevelConfigurable="true"
IsFileCollectable="true" />
<dp:TraceComponent Name="Web Setup" Description="Web Setup"
IsLevelConfigurable="true" IsFileCollectable="true" />
</dp:TraceComponentList>
</dp:ListTraceComponentsReply>

```

10.1.5.4 ListTraceFiles

Lists trace files for that application component/subcomponent during the FromDate and ToDate parameters [which are in UTC].

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/ListTraceFiles?Component/Subcomponent&Fromdate=0&ToDate=0>

Reply example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<dp:ListTraceFilesReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/analysismanager">
<dp:Schema Version="1.0"/>
<dp:TraceFileList>
  <dp:FileProperty Name="TraceFile1.TXT" Date="1212347735" Size="1000000"/>
  <dp:FileProperty Name="TraceFile2.TXT" Date="1212347835" Size="1000000"/>
  <dp:FileProperty Name="TraceFile3.TXT" Date="1212347935" Size="1000000"/>
</dp:TraceFileList>
</dp:ListTraceFilesReply>
```

- Optional URL parameter "Type" is applicable only for components that generate multiple trace types.
- URL parameters "FromDate" and "ToDate" are used to specify time range of trace files requested by user. It is required for ICM components to supply these parameters.
- Attribute "Date" specifies file modification time in UTC.
- Attribute "Size" specifies file size in bytes.

10.1.5.5 DownloadTraceFile

Download the trace files that were returned by the ListTraceFiles API.

Note: Only one file may be requested at a time.

However, for trace files, there will always be just one zip file (included trace files, capture files, and others) returned by ListTraceFiles API. Only one download request is needed.

Note: Subsequent download request with the same filename will be returned with an error because once the file is downloaded, it is deleted from the server.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/DownloadTraceFile?Component=Component/Subcomponent&File=TraceFile1.txt>

Reply:

There are four possible replies:

- The server streams the specified file unzipped over the existing HTTP connection. Content (MIME) type will be defined by the app server as "application/text".
- The server streams the specified file zipped over the existing HTTP connection. Content (MIME) type will be defined by app server as "application/zip".
- The server streams the specified file gzipped over the existing HTTP connection. Content (MIME) type will be defined by app server as "application/x-gzip".
- In case of error, app server replies error condition in following XML format (MIME type "application/xml"):

```
<?xml version="1.0" encoding="UTF-8" ?>
<dp:DownloadTraceFileReply ReturnCode="1" ErrorString="File TraceFile1.txt
not found." />
"xmlns:dp="http://www.cisco.com/vtg/analysismanager">
```

10.1.5.6 ListLogComponents

Lists all possible application components that produce log files.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/ListLogComponents>

Reply example:

```
<?xml version="1.0" encoding="utf-8" ?>
<dp:ListLogComponentsReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/analysismanager">
<dp:Schema Version="1.0" />
<dp:LogComponentList>
  <dp:LogComponent Name="ICM Installation and Upgrade" Description="ICM
Installation and Upgrade logs" />
  <dp:LogComponent Name="ICMDBA" Description="ICM DBA logs" />
  <dp:LogComponent Name="Performance Counter" Description="Performance
Counter Logs" />
  <dp:LogComponent Name="Active Directory" Description="Logs for
troubleshooting Active Directory issues." />
  <dp:LogComponent Name="Cisco ICM Diagnostic Framework Install"
Description="Cisco ICM Diagnostic Framework Install Logs" />
  <dp:LogComponent Name="Cisco ICM Diagnostic CLI" Description="Cisco ICM
Diagnostic CLI Logs" />
  <dp:LogComponent Name="Dr Watson" Description="Dr.Watson logs" />
  <dp:LogComponent Name="Cisco Security Agent" Description="Cisco Security
Agent logs" />
  <dp:LogComponent Name="Security Hardening" Description="Security Hardening
logs" />
  <dp:LogComponent Name="Webview Job Scheduler" Description="Webview Job
Scheduler logs" />
  <dp:LogComponent Name="Cisco CCBU Support Tools" Description="Support
Tools logs" />
  <dp:LogComponent Name="Web Setup" Description="Web Setup troubleshooting
and audit logs" />
  <dp:LogComponent Name="Web Agent Re-skilling" Description="Web Agent Re-
skilling troubleshooting logs" />
</dp:LogComponentList>
</dp:ListLogComponentsReply>
```

10.1.5.7 ListLogFiles

Lists log files for that application component/subcomponent during the FromDate and ToDate parameters [which are in UTC].

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/ListLogFiles?Component=Component/Subcomponent&FromDate=0&ToDate=0>

Reply example:

```
<?xml version="1.0" encoding="UTF-8"?>
<dp:ListLogFilesReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/analysismanager">
<dp:Schema Version="1.0"/>
<dp:LogFileList>
  <dp:FileProperty Name="LogFile1.txt" Date="1212347735" Size="1000000"/>
  <dp:FileProperty Name="LogFile2.txt" Date="1212347835" Size="1000000"/>
  <dp:FileProperty Name="LogFile3.txt" Date="1212347935" Size="1000000"/>
</dp:LogFileList>
</dp:ListLogFilesReply>
```

10.1.5.8 DownloadLogFile

Download the log files that were returned by the ListLogFiles API.

Note: Only one file may be requested at a time.

In the case of downloading the log files, a user may request subsequent download with the same filename, the exact same file will be returned. This is different from the trace file because we are not deleting the log file from the server.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/DownloadLogFile?Component=Component/Subcomponent&File=LogFile1.txt>

Reply:

There are four possible replies:

- The server streams the specified file unzipped over the existing HTTP connection. Content (MIME) type will be defined by the app server as "application/text".
- The server streams the specified file zipped over the existing HTTP connection. Content (MIME) type will be defined by app server as "application/zip".
- The server streams the specified file zipped over the existing HTTP connection. Content (MIME) type will be defined by app server as "application/x-gzip".
- In case of error, app server replies error condition in following XML format (MIME type "application/xml"):

```
<?xml version="1.0" encoding="UTF-8" ?>
<dp:DownloadLogFileReply ReturnCode="1" ErrorString="File LogFile1.txt not
found." xmlns:dp="http://www.cisco.com/vtg/analysismanager">
```

10.1.5.9 ListAppServers

Lists the applications and application components installed on the target server.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/ListAppServers>

Reply example:

```
<?xml version="1.0" encoding="utf-8" ?>
<dp:ListAppServersReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/anaysismanager">
  <dp:Schema Version="1.0" />
  <dp:AppServerList>
    <dp:AppServer Name="buzzards-bay" ProductType="ICM"
      ProductComponentType="Logger A" />
    <dp:AppServer Name="buzzards-bay" ProductType="ICM"
      ProductComponentType="Router A" />
    <dp:AppServer Name="buzzards-bay" ProductType="ICM"
      ProductComponentType="Cisco ICM Diagnostic Framework"
    />
  </dp:AppServerList>
</dp:ListAppServersReply>
```

- <AppServer> has following optional attributes
 - "ProductType"- for product to reply topology information. Needs to be one of the following (CVP, UCCX, CUCM, UCCE, EA, IOS).
 - "ProductComponentType" – component type within a product. For example, Router, PG, etc.

10.1.5.10 ListConfigurationCategories

Lists the configuration categories available on this application server.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/ListConfigurationCategories>

Reply example:

```
<?xml version="1.0" encoding="utf-8" ?>
<dp:ListConfigurationCategoriesReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/analysismanager">
  <dp:Schema Version="1.0" />
  <dp:ConfigurationCategoryList>
    <dp:ConfigurationCategory Name="DumpCfg"
      Description="ConfigurationCategory
      for DumpCfg; Instance=ipcc8" />
    <dp:ConfigurationCategory Name="ExportICMCfg"
      Description="ConfigurationCategory for ExportICMCfg; Instance=ipcc8"
    />
    <dp:ConfigurationCategory Name="ConfigExport"
      Description="ConfigurationCategory for ConfigExport; Instance=ipcc8"
    />
    <dp:ConfigurationCategory Name="Registry"
      Description="ConfigurationCategory
      for Registry; Instance=ipcc8" />
  </dp:ConfigurationCategoryList>
</dp:ListConfigurationCategoriesReply>
```

10.1.5.11 GetConfigurationCategory

Retrieve configuration information based on category.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/GetConfigurationCategory?Category=???>

Categories are: “DumpCfg”, “ExportICMCfg”, “ConfigExport” and “Registry”

Reply example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<dp:GetConfigurationCategoryReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/analysismanager">
<dp:Schema Version="1.0" />
</dp:GetConfigurationCategoryReply>
```

The requested configuration data is returned as a zip file.

10.1.5.12 *GetProductVersion*

Fetches the version of the application(s) installed on the target server.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/GetProductVersion>

Reply example:

```
<?xml version="1.0" encoding="utf-8" ?>
<dp:GetProductVersionReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/analysismanager">
<dp:Schema Version="1.0" />
<dp:ProductVersion Name="ICM" Major="8" Minor="0" Maintenance="1"
VersionString="8.0(1) BuildNumber=26380" />
</dp:GetProductVersionReply>
```

10.1.5.13 *GetProductLicense*

Get license information for application(s) installed on target server.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/GetProductLicense>

Reply example:

```
<?xml version="1.0" encoding="utf-8" ?>
<dp:GetProductLicenseReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/analysismanager">
<dp:Schema Version="1.0" />
<dp:LicenseList>
<dp:License>
<dp:PropertyList>
<dp:Property Name="License" Value="Unified ICM/UCCE does not have any
license information." />
</dp:PropertyList>
</dp:License>
</dp:LicenseList>
</dp:GetProductLicenseReply>
```

10.1.5.14 *GetPlatformInformation*

Fetches server and operating system platform details.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/GetPlatformInformation>

Reply example:

```
<?xml version="1.0" encoding="utf-8" ?>
<dp:GetPlatformInformationReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/analysismanager">
<dp:Schema Version="1.0" />
<dp:PlatformInformation>
<dp:PropertyList>
  <dp:Property Name="Host Name" Value="BUZZARDS-BAY" />
  <dp:Property Name="OS Platform" Value="Win32NT" />
  <dp:Property Name="OS Service Pack" Value="Service Pack 2" />
  <dp:Property Name="OS Version" Value="5.2.3790.131072" />
  <dp:Property Name="OS Version String" Value="Microsoft Windows NT 5.2.3790
Service Pack 2" />
  <dp:Property Name="System Directory" Value="C:\WINDOWS\system32" />
  <dp:Property Name="User Domain Name" Value="SILVERBACK" />
  <dp:Property Name="Common Language Runtime Version" Value="2.0.50727.3053"
/>
  <dp:Property Name="Admin Password Status" Value="3 []" />
  <dp:Property Name="Daylight Time In Effect" Value="True []" />
  <dp:Property Name="User Name" Value="[unavailable]" />
  <dp:Property Name="Computer Manufacturer" Value="HP" />
  <dp:Property Name="Model" Value="ProLiant DL380 G5" />
  <dp:Property Name="Number Of Processors" Value="[unavailable]" />
  <dp:Property Name="Total Physical Memory" Value="2145230848" />
  <dp:Property Name="Boot Device" Value="\Device\HarddiskVolume1" />
  <dp:Property Name="Build Number" Value="3790" />
  <dp:Property Name="Build Type" Value="Multiprocessor Free" />
  <dp:Property Name="Caption" Value="Microsoft(R) Windows(R) Server 2003,
Standard Edition" />
  <dp:Property Name="Current Time Zone" Value="-240" />
  <dp:Property Name="IS OS a Debug version?" Value="False" />
  <dp:Property Name="Free Physical Memory" Value="653648" />
  <dp:Property Name="Free Virtual Memory" Value="2724228" />
  <dp:Property Name="Install Date" Value="Friday, February 13, 2009 3:03:50
PM" />
  <dp:Property Name="Large System Cache" Value="1 []" />
  <dp:Property Name="Locale Code" Value="0409" />
  <dp:Property Name="OS Manufacturer" Value="Microsoft Corporation" />
  <dp:Property Name="Max Process Memory Size" Value="2097024" />
  <dp:Property Name="OS Name" Value="Microsoft Windows Server 2003 Standard
Edition|C:\WINDOWS|\Device\Harddisk0\Partition1" />
  <dp:Property Name="Number Of Processes" Value="66" />
  <dp:Property Name="Number Of Users" Value="10" />
  <dp:Property Name="ServicePackMajorVersion" Value="2" />
  <dp:Property Name="ServicePackMinorVersion" Value="0" />
  <dp:Property Name="System Directory" Value="C:\WINDOWS\system32" />
  <dp:Property Name="System Drive" Value="C:" />
  <dp:Property Name="Total Virtual Memory" Value="4044744" />
  <dp:Property Name="Total Visible Memory" Value="2094952" />
  <dp:Property Name="Windows Directory" Value="C:\WINDOWS" />
</dp:PropertyList>
</dp:PlatformInformation>
</dp:GetPlatformInformationReply>
```

10.1.5.15 GetNetStat

Execute a NETSTAT command remotely on the target server and return the results.

Request:

[https://<server>:<port>/icm-dp/rest/AnalysisManager/GetNetStat?Arguments="-an"](https://<server>:<port>/icm-dp/rest/AnalysisManager/GetNetStat?Arguments=)

Reply:

Returns a text file with the output from the command execution.

10.1.5.16 GetIPConfig

Execute an IPCONFIG command remotely on the target server and return the results.

Request:

[https://<server>:<port>/icm-dp/rest/AnalysisManager/GetIPConfig?Arguments="/all"](https://<server>:<port>/icm-dp/rest/AnalysisManager/GetIPConfig?Arguments=)

Reply:

Returns a text file with the output from the command execution.

10.1.5.17 GetTraceRoute

Execute a TRACERT command remotely on the target server and return the results.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/GetTraceRoute>

Reply:

Returns a text file with the output from the command execution.

10.1.5.18 GetPing

Execute a PING command remotely on the target server and return the results.

Request:

[https://<server>:<port>/icm-dp/rest/AnalysisManager/GetPing?Arguments="n.n.n.n"](https://<server>:<port>/icm-dp/rest/AnalysisManager/GetPing?Arguments=)

Reply:

Returns a text file with the output from the command execution.

10.1.5.19 ListProcesses

Lists application processes running on the target server.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/ListProcesses>

Reply example:

```
<?xml version="1.0" encoding="utf-8" ?>
<dp:ListProcessesReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/analysismanager">
<dp:Schema Version="1.0" />
<dp:ServiceList>
  <dp:Service Name="Cisco CCBU Support Tools NodeAgent">
    <dp:ProcessList>
      <dp:ProcessProp Name="appserver.exe" Description="appserver" />
    </dp:ProcessList>
  </dp:Service>
  <dp:Service Name="Logger A">
    <dp:ProcessList>
      <dp:ProcessProp Name="nodeman.exe" Description="nodeman" />
      <dp:ProcessProp Name="nmm.exe" Description="nmm" />
      <dp:ProcessProp Name="configlogger.exe" Description="configlogger" />
      <dp:ProcessProp Name="csfs.exe" Description="csfs" />
      <dp:ProcessProp Name="cw2kfeed.exe" Description="cw2kfeed" />
      <dp:ProcessProp Name="histlogger.exe" Description="histlogger" />
      <dp:ProcessProp Name="recovery.exe" Description="recovery" />
      <dp:ProcessProp Name="replication.exe" Description="replication" />
    </dp:ProcessList>
  </dp:Service>
  <dp:Service Name="Router A">
    <dp:ProcessList>
      <dp:ProcessProp Name="nodeman.exe" Description="nodeman" />
      <dp:ProcessProp Name="nmm.exe" Description="nmm" />
      <dp:ProcessProp Name="ccagent.exe" Description="ccagent" />
      <dp:ProcessProp Name="dbagent.exe" Description="dbagent" />
      <dp:ProcessProp Name="mdsproc.exe" Description="mdsproc" />
      <dp:ProcessProp Name="router.exe" Description="router" />
      <dp:ProcessProp Name="rtsvr.exe" Description="rtsvr" />
      <dp:ProcessProp Name="testsync.exe" Description="testsync" />
    </dp:ProcessList>
  </dp:Service>
  <dp:Service Name="Cisco ICM Diagnostic Framework">
    <dp:ProcessList>
      <dp:ProcessProp Name="DiagFwSvc.exe" Description="DiagFwSvc" />
    </dp:ProcessList>
  </dp:Service>
</dp:ServiceList>
</dp:ListProcessesReply>
```

10.1.5.20 ListServices

Lists application services running on the target server.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/ListServices>

Reply example:

```
<?xml version="1.0" encoding="utf-8" ?>
<dp:ListServicesReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/AnalysisManager">
<dp:Schema Version="1.0" />
<dp:ServiceList>
  <dp:Service Name="Cisco CCBU Support Tools NodeAgent"
Description="Provides
  Support Tools communication support and processing" Status="Running"
  StartupType="Auto" LogOnAs="LocalSystem" />
```

```
<dp:Service Name="Cisco ICM ipcc8 LoggerA" Description="Cisco ICM ipcc8
  LoggerA" Status="Running" StartupType="Auto"
  LogOnAs="SILVERBACK.CISCO.COM\IPCC8-LOGGERA-77B585" />
<dp:Service Name="Cisco ICM ipcc8 RouterA" Description="Cisco ICM ipcc8
  RouterA" Status="Running" StartupType="Auto" LogOnAs="LocalSystem" />
<dp:Service Name="Cisco ICM Diagnostic Framework" Description="Provides a
  web-based diagnostic service for Cisco Unified ICM, Contact Center
  Enterprise application." Status="Running" StartupType="Auto"
  LogOnAs="silverback\w2003admin" />
</dp:ServiceList>
</dp:ListServicesReply>
```

10.1.5.21 GetPerformanceInformation

Get a set of System and Application Performance Counters for the specified server.

Request:

[rformanceInformation](#)

Reply example :

```
<?xml version="1.0" encoding="utf-8" ?>
<dp:GetPerformanceInformationReply ReturnCode="0"
  xmlns:dp="http://www.cisco.com/vtg/analysismanager">
  <dp:Schema Version="1.0" />
  <dp:PerformanceInformation>
  <dp:PropertyList>
    <dp:Property Name="Memory/Memory Page Faults/sec" Value="29.93962" />
    <dp:Property Name="Process(_Total)/Handle Count" Value="20386" />
    <dp:Property Name="Processor(_Total)/% Processor Time" Value="13.63913" />
    <dp:Property Name="Memory/Total Memory" Value="1.399697E+09" />
    <dp:Property Name="System/Threads" Value="1165" />
    <dp:Property Name="Memory/Memory Pages/Sec" Value="3.654335" />
    <dp:Property Name="System/Processor Queue" Value="0" />
    <dp:Property Name="System/Processes" Value="73" />
    <dp:Property Name="Cisco ICM Logger(ipcc8 LoggerA)/DB Write Average Time"
  Value="0" />
    <dp:Property Name="Cisco ICM Logger(ipcc8 LoggerA)/DB Write Records
  processed" Value="0" />
    <dp:Property Name="Cisco ICM Router(ipcc8 RouterA)/Calls/sec" Value="0" />
    <dp:Property Name="Cisco ICM Router(ipcc8 RouterA)/Agents Logged On"
  Value="0" />
    <dp:Property Name="Cisco ICM Router(ipcc8 RouterA)/Calls In Progress"
  Value="0" />
    <dp:Property Name="Cisco ICM Router(ipcc8 RouterA)/Calls In Queue"
  Value="0" />
    <dp:Property Name="Cisco ICM Router(ipcc8 RouterA)/Router State Size(KB)"
  Value="0" />
    <dp:Property Name="Cisco ICM Router(ipcc8 RouterA)/Messages Processed/sec"
  Value="0" />
    <dp:Property Name="Cisco ICM Router(ipcc8 RouterA)/Bytes Processed/sec"
  Value="0" />
    <dp:Property Name="Cisco ICM Router(ipcc8 RouterA)/Avg Process
  Time/Message (ms)" Value="0" />
    <dp:Property Name="Cisco ICM Router(ipcc8 RouterA)/Max Process Time(ms)"
  Value="0" />
    <dp:Property Name="Cisco ICM Router(ipcc8 RouterA)/Calls In Router"
  Value="0" />
  </dp:PropertyList>
</dp:PerformanceInformation>
</dp:GetPerformanceInformationReply>
```

10.1.5.22 GetPerfCounterValue

Get the current value of a performance counter from the target server.

Request:

[https://<server>:<port>/icm-dp/rest/AnalysisManager/GetPerfCounterValue?CategoryName=Processor&CounterName="% Processor Time"&PerfInstance="_ Total"](https://<server>:<port>/icm-dp/rest/AnalysisManager/GetPerfCounterValue?CategoryName=Processor&CounterName=)

Reply example:

```
<?xml version="1.0" encoding="utf-8" ?>
<dp:GetPerfCounterValueReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/analysismanager">
<dp:Schema Version="1.0" />
<dp:PerformanceInformation>
<dp:PropertyList>
  <dp:Property Name="CategoryName" Value="Processor" />
  <dp:Property Name="CounterName" Value="% Processor Time" />
  <dp:Property Name="InstanceName" Value="_Total" />
  <dp:Property Name="BaseValue" Value="0" />
  <dp:Property Name="CounterFrequency" Value="0" />
  <dp:Property Name="CounterTimeStamp" Value="0" />
  <dp:Property Name="CounterType" Value="Timer100NsInverse" />
  <dp:Property Name="RawValue" Value="203276171875" />
  <dp:Property Name="NextValue" Value="0.003199898" />
  <dp:Property Name="SystemFrequency" Value="2333380000" />
  <dp:Property Name="TimeStamp" Value="48917923479390" />
  <dp:Property Name="TimeStamp100nSec" Value="128929442042854145" />
</dp:PropertyList>
</dp:PerformanceInformation>
</dp:GetPerfCounterValueReply>
```

10.1.5.23 GetAlarms

Retrieves up to 25 of the most recent alarms generated by Unified CCE.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/GetAlarms?Severity=#?Count=##>

“Severity” and “Count” are optional parameters.

Severity may be a numeric value between 1 and 3 (1=Informational, 2=Warning, 3=Error) – returns all alarms with a severity greater-than or equal-to the specified severity.

Count may be a numeric value between 1 and 25 – returns a maximum of the specified number of alarms.

Reply example:

```
<?xml version="1.0" encoding="utf-8" ?>
<dp:GetAlarmsReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/analysismanager">
<dp:Schema Version="1.0" />
<dp:AlarmList>
  <dp:Alarm DateTime="Jul 24, 2009 15:41:41 +0000" Type="Clear" Id="1028104"
Severity="1" Instance="ipcc8" Component="4_5_BERKSHIRE_ICM\ipcc8\LoggerB"
SubComponent="nm" Message="ICM\ipcc8\LoggerB Node Manager started. Last
shutdown was due to system shutdown." />
  <dp:Alarm DateTime="Jul 24, 2009 15:41:27 +0000" Type="Clear" Id="10500FF"
Severity="1" Instance="ipcc8" Component="24_1_B_hlgr" SubComponent="rtr"
Message="Side B hlgr process is OK." />
</dp:AlarmList>
</dp:GetAlarmsReply>
```

```

    <dp:Alarm DateTime="Jul 24, 2009 15:42:37 +0000" Type="Clear" Id="10500FF"
    Severity="1" Instance="ipcc8" Component="24_1_B_clgr" SubComponent="rtr"
    Message="Side B clgr process is OK." />
    <dp:Alarm DateTime="Jul 24, 2009 15:41:27 +0000" Type="Clear" Id="10500FF"
    Severity="1" Instance="ipcc8" Component="24_1_B_clgr" SubComponent="rtr"
    Message="Side B clgr process is OK." />
    <dp:Alarm DateTime="Jul 24, 2009 15:41:14 +0000" Type="Clear" Id="10F8004"
    Severity="1" Instance="ipcc8" Component="6_1_BERKSHIRE_B_PG01"
    SubComponent="ccag" Message="Device PG01 path changing to idle state." />
    <dp:Alarm DateTime="Jul 24, 2009 15:41:14 +0000" Type="Clear" Id="102C107"
    Severity="1" Instance="ipcc8" Component="4_1_BERKSHIRE_ICM\ipcc8\RouterB"
    SubComponent="nm" Message="ICM\ipcc8\RouterB Node Manager started. Last
    shutdown was for reboot after failure of critical process." />
    <dp:Alarm DateTime="Jul 24, 2009 15:41:13 +0000" Type="Clear" Id="10500FF"
    Severity="1" Instance="ipcc8" Component="24_1_B_rts" SubComponent="rtr"
    Message="Side B rts process is OK." />
    <dp:Alarm DateTime="Jul 24, 2009 15:41:12 +0000" Type="Clear" Id="10500FF"
    Severity="1" Instance="ipcc8" Component="24_1_B_rtr" SubComponent="rtr"
    Message="Side B rtr process is OK." />
    <dp:Alarm DateTime="Jul 24, 2009 15:41:12 +0000" Type="Clear" Id="10500FF"
    Severity="1" Instance="ipcc8" Component="24_1_B_tsyrr" SubComponent="rtr"
    Message="Side B tsyrr process is OK." />
    <dp:Alarm DateTime="Jul 24, 2009 15:41:12 +0000" Type="Clear" Id="10500FF"
    Severity="1" Instance="ipcc8" Component="24_1_B_csfs" SubComponent="rtr"
    Message="Side B csfs process is OK." />
    <dp:Alarm DateTime="Jul 24, 2009 15:41:12 +0000" Type="Clear" Id="10500FF"
    Severity="1" Instance="ipcc8" Component="24_1_B_rcv" SubComponent="rtr"
    Message="Side B rcv process is OK." />
    <dp:Alarm DateTime="Jul 24, 2009 15:41:12 +0000" Type="Clear" Id="10500FF"
    Severity="1" Instance="ipcc8" Component="24_1_B_dba" SubComponent="rtr"
    Message="Side B dba process is OK." />
    <dp:Alarm DateTime="Jul 24, 2009 15:42:20 +0000" Type="Clear" Id="10500FF"
    Severity="1" Instance="ipcc8" Component="24_1_B_rtr" SubComponent="rtr"
    Message="Side B rtr process is OK." />
    <dp:Alarm DateTime="Jul 24, 2009 15:42:20 +0000" Type="Clear" Id="10500FF"
    Severity="1" Instance="ipcc8" Component="24_1_B_tsyrr" SubComponent="rtr"
    Message="Side B tsyrr process is OK." />
    <dp:Alarm DateTime="Jul 24, 2009 15:42:20 +0000" Type="Clear" Id="10500FF"
    Severity="1" Instance="ipcc8" Component="24_1_B_csfs" SubComponent="rtr"
    Message="Side B csfs process is OK." />
    <dp:Alarm DateTime="Jul 24, 2009 15:42:20 +0000" Type="Clear" Id="10500FF"
    Severity="1" Instance="ipcc8" Component="24_1_B_rcv" SubComponent="rtr"
    Message="Side B rcv process is OK." />
    <dp:Alarm DateTime="Jul 24, 2009 15:42:20 +0000" Type="Clear" Id="10500FF"
    Severity="1" Instance="ipcc8" Component="24_1_B_dba" SubComponent="rtr"
    Message="Side B dba process is OK." />
    <dp:Alarm DateTime="Jul 24, 2009 15:42:18 +0000" Type="Clear" Id="1040023"
    Severity="1" Instance="ipcc8" Component="5_1_0" SubComponent="mds"
    Message="Communication with peer Synchronizer established." />
    <dp:Alarm DateTime="Jul 24, 2009 15:37:55 +0000" Type="Clear" Id="1028103"
    Severity="1" Instance="ipcc8"
    Component="4_4_WACHUSETT_ICM\ipcc8\Distributor" SubComponent="nm"
    Message="ICM\ipcc8\Distributor Node Manager started. Last shutdown was by
    operator request." />
    <dp:Alarm DateTime="Jul 24, 2009 15:37:41 +0000" Type="Clear" Id="102C110"
    Severity="2" Instance="ipcc8"
    Component="3_4_WACHUSETT_ICM\ipcc8\Distributor_uaw" SubComponent="nm"
    Message="ICM\ipcc8\Distributor node process uaw successfully reinitialized
    after restart." />
    <dp:Alarm DateTime="Jul 24, 2009 15:37:40 +0000" Type="Clear" Id="102C10A"
    Severity="2" Instance="ipcc8"
    Component="3_4_WACHUSETT_ICM\ipcc8\Distributor_uaw" SubComponent="nm"

```



```

Message="ICM\ipcc8\Distributor node restarting process uaw after having
delayed restart for 1 seconds." />
  <dp:Alarm DateTime="Jul 24, 2009 15:37:39 +0000" Type="Raise" Id="102C10F"
Severity="2" Instance="ipcc8"
Component="3_4_WACHUSETT_ICM\ipcc8\Distributor_uaw" SubComponent="nm"
Message="Process uaw on ICM\ipcc8\Distributor is down after running for 30
seconds. It will restart after delaying 1 second for related operations to
complete." />
  <dp:Alarm DateTime="Jul 24, 2009 15:37:39 +0000" Type="Raise" Id="102C10E"
Severity="3" Instance="ipcc8"
Component="3_4_WACHUSETT_ICM\ipcc8\Distributor_uaw" SubComponent="nm"
Message="Process uaw on ICM\ipcc8\Distributor went down for unknown reason.
Exit code 0x1. It will be automatically restarted." />
  <dp:Alarm DateTime="Jul 24, 2009 15:37:14 +0000" Type="Clear" Id="102C111"
Severity="1" Instance="ipcc8"
Component="3_4_WACHUSETT_ICM\ipcc8\Distributor_rpl" SubComponent="nm"
Message="ICM\ipcc8\Distributor node process rpl successfully started." />
  <dp:Alarm DateTime="Jul 24, 2009 15:37:13 +0000" Type="Clear" Id="102C111"
Severity="1" Instance="ipcc8"
Component="3_4_WACHUSETT_ICM\ipcc8\Distributor_rtc" SubComponent="nm"
Message="ICM\ipcc8\Distributor node process rtc successfully started." />
</dp:AlarmList>
</dp:GetAlarmsReply>

```

10.1.5.24 SetAlarms

Turns Unified CCE alarming OFF or ON. Turning alarming OFF is useful during maintenance windows to prevent flooding at the management station.

Request:

<https://<server>:<port>/icm-dp/rest/AnalysisManager/SetAlarms?State=ON/OFF>

Reply example:

```

<?xml version="1.0" encoding="utf-8" ?>
<dp:SetAlarmsReply ReturnCode="0"
xmlns:dp="http://www.cisco.com/vtg/analysismanager">
<dp:Schema Version="1.0" />
</dp:SetAlarmsReply>

```

10.1.6 Diagnostic Framework Troubleshooting

The Diagnostic Framework is self contained and does not require any additional configuration other than assigning users. In case you encounter any issues with the service, refer to the following table:

Table 10-10: Diagnostic Framework Troubleshooting

Issue	Troubleshooting / Remedy
Diagnostic Framework service won't start	<p>Check if required service HTTP SSL (and IIS, when installed) is started without any errors. Check Windows Event log for errors and resolve any issues with the required service(s).</p> <p>Make sure none of the configuration files is missing.</p> <p>Check Event Viewer and Diagnostic Framework log file for any initialization errors.</p>
Cannot access any API from the client, such as Internet Explorer	<p>Confirm the base URL is correct; compare it with the URL in the service configuration file DiagFwSvc.exe.config.</p> <p>Confirm the API used is valid; try accessing the built in GetMenu API.</p> <p>Make sure the API is accessed using HTTPS.</p> <p>Make sure the credentials used as valid, check Windows Event log for any authentication errors and Diagnostic Framework log for any authorization errors.</p> <p>Use DiagFwCertMgr utility to validate the certificate binding to the port in use. Recreate or rebind the certificate if any issues were found.</p> <p>If using Internet Explorer, clear the cache and restart the browser.</p> <p>Verify that the Windows Firewall is either turned off, or that it has been configured with the ICM Security Wizard, which ensures that a proper exception is in place for the Diagnostic Framework to work.</p>
Some commands work, and others don't seem to work.	<p>Make sure you're using an approved browser. Currently only IE 6 and IE 7 are approved. Most commands work via Firefox, but we've seen problems when trying to return files</p>

10.2 DUMPLOG

Using the DUMPLOG Utility's 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 10-11: APPNAME and TAGS Used in DUMPLOG Trace Output

Field	Description
APPNAME	PRODUCT_COMPONENT_MESSAGECATEGORY PRODUCT - always ICM COMPONENT – such as Router MESSAGECATEGORY – such as ProcessSynchronization
TAGS	Acceptable tags are: [comp=%s] - component name including side, such as Router A [pname=%s] - process name, such as rtr [iid=%s] - instance name, such as ipcc [sev=%s] – severity, such as info 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-rgrla
cccaLoggerRouterSideBName.0.2 = cc-rgrla
cccaLoggerDuplexPairName.0.2 = cc-rgrla
cccaLoggerHDSReplication.0.2 = 0
cccaDistAwSide.0.4 = sideA(1)
cccaDistAwType.0.4 = standard(0)
cccaDistAwAdminSiteName.0.4 = cc-rgrla
cccaDistAwRouterSideAName.0.4 = cc-rgrla
cccaDistAwRouterSideBName.0.4 = cc-rgrla
cccaDistAwLoggerSideAName.0.4 = cc-rgrla
cccaDistAwLoggerSideBName.0.4 = cc-rgrla
cccaDistAwDuplexPairName.0.4 = cc-rgrla
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-rgrla
cccaPgRouterSideBName.0.3 = cc-rgrla
cccaPgDuplexPairName.0.3 = cc-rgrla
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 second.
				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 rrttest. 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 customer's configuration size has grown larger. The alarm limit can be raised with the rsetting 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 router 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.	

				<p>Logger or HDS on the specified TCP/IP connection and port number is either out of service or communication has broken.</p> <p>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.</p>
118C040	Single-State Raise	Warning	LGR MISSING NETWORK ROUTING CLIENT	<p>Found %1 records with DateTime greater than current Central Controller Time %2 in %3 table. Check and correct the errors.</p> <p>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.</p> <p>No action is required.</p>
118C04F	Raise	Warning	LGR HDS RUN BEHIND	<p>HDS Running Behind: %1 is running behind its logger %2 by %3 minutes.</p> <p>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.</p> <p>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.</p>
12B001F	Clear	Error	APPGW APPGW	<p>Application Gateway has connected with the host. Application Gateway ID = %1</p> <p>The application gateway is now connected to the host process.</p> <p>No action is required.</p>
12B0020	Raise	Error	APPGW APPGW	<p>The external database has disconnected from the Application Gateway (ID = %1). Routing may be impacted.</p> <p>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.</p> <p>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.</p>
12E8006	Clear	Informational	CTI SESSION	<p>CONNECTION MONITOR SERVICE: Enterprise CTI session established by Client %1 (%2) at %3.</p> <p>An Enterprise CTI session has been opened by ClientID %1 (Signature %2) from IP address %3.</p> <p>No action is required.</p>
12E8007	Raise	Warning	CTI SESSION	<p>CONNECTION MONITOR SERVICE: Enterprise CTI session closed by Client %1 (%2) at %3.</p> <p>The Enterprise CTI session with ClientID %1 (Signature %2) at IP address %3 has been closed by the client.</p> <p>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.</p>
12E8008	Raise	Error	CTI SESSION	<p>CONNECTION MONITOR SERVICE: Enterprise CTI session terminated with Client %1 (%2) at %3.</p> <p>The Enterprise CTI session with ClientID %1 (Signature %2) at IP address %3 has been terminated by the Enterprise CTI Server.</p> <p>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.</p>
12E800C	Clear	Informational	CTI NORMAL OBJECT EVENT	<p>Client:%1 Object:%2 Normal Event Report: %3</p>

				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.
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.
1438004	Clear	Error	BA IMPORT	The import for table [%1] has been successful.
				An import has completed successfully.
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.
1438006	Raise	Error	BA IMPORT	Failed to build dialing list from table [%1].
				A Dialing list could not be populated from the specified table.
1438008	Raise	Error	BA DIALER NETWORK	Could not connect to Campaign Manager.
				Either the Campaign Manager is not running or a network connection can't be established due to connectivity issues.
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.
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.
1438011	Raise	Error	BA CTI	Blended Agent CTI Server connection on computer [%1] is down.
				The Blended Agent CTI Server connection has been terminated.
1438012	Clear	Error	BA CTI	Blended Agent CTI Server connection on computer [%1] is active.
				Blended Agent CTI Server connection is active.
1438017	Raise	Error	CAMPAIGN MANAGER IMPORT	Process is down on computer [%1].
				Process is not running on the specified computer.
1438018	Clear	Error	CAMPAIGN MANAGER IMPORT	Process is up on computer [%1].
				Process is running on the specified computer.

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.