Back: Troubleshooting CTI OS

View/Add tips for Release 7.5(1): Category:CTI OS, Release 7.5

Contents

Troubleshooting Checklist

Note: Troubleshooting in a CTI OS installation can be simple if you follow the correct procedure, and very difficult if done arbitrarily. Do not forget that you are dealing with a multi-component distributed system and that the source of the problem may not be the component where the symptoms are seen.

Following are the steps that you need for troubleshooting a CTI OS Installation:

Note: Steps 3 to 7 are the ?Problem Isolation? procedure used to discard any malfunction of the components in Cisco ICM.

1. Write down all the steps that reproduce the problem.
2. Write down the call flow clearly and concisely.
3. Bump the trace mask up to 0x00020A0f as minimum on CTI OS Server.
4. Bump the trace mask up to 0x00000A0f for the CTI OS Clients. (Use 0x00003E0F if you are troubleshooting Silent Monitor problems)
5. Use the hard phone and verify whether the problem repeats. If the problem no longer occurs using the hard phone, rule out the CallManager/ACD as the source of the problem.
6. Use CTITest and verify whether the problem repeats. If the problem no longer occurs using CTITest, rule out the CTI Server, PIM and OPC as the sources of the problem.
7. Use CILTest and verify whether the problem repeats. If the problem no longer occurs using the CILTest, rule out the CTI OS Server as the source of the problem.
8. Use CTI OS Desktop and verify whether the problem repeats. Using CTI OS Desktops will help isolate the problem to the phone if the problem is reproducible there.
9. Examine the CTI OS configuration registry.
10. If the problem cannot be fixed onsite, collect all logs and version numbers, and submit them to TAC. Remember to include call flow and logs for OPC, PIM, CTI Server, CTI OS Server Node, CTI Driver, CTI Client and JTAPI Gateway (if working with IPCC). Also include all comments about the troubleshooting efforts you made.

Obtaining Logs for Support

When you report a problem to Cisco, Cisco personnel will ask that you supply certain details * about the problem. You should be prepared to provide Cisco with the following details about your problem when you call:

- At exactly what time did the problem happen?
- What was the agent ID of affected agent?
- What was the device ID of affected device?
- What was the call ID of affected call?

- What was the affected agent doing prior to the failure?
- What buttons if any were pressed, and what buttons were enabled?
- Was a call in the grid at the time, and was the call on the hard phone?
- What was the call flow?

In addition, Cisco will usually require logs in order to troubleshoot a problem. It is best to collect all of the following logs for the timeframe when the problem occurred:

- CTI Toolkit
- CTI OS server
- CTI server
- PIM
- OPC
- JTAPI Gateway (only if using IPCC)

Include logs for all of the relevant servers, including both sides of a duplexed system. The following sections discuss CTI OS Server and CTI Toolkit logs and trace levels. See the *ICM Administration Guide for Cisco ICM Enterprise Edition* for information on other logs.

Taking CTI OS Server Logs

The trace log location for the server processes can be found under the following directory:

```
<drive>:\ICM\<customer_instance>\CTIOS1\logfiles
```

Files are named using the convention <process name>_yymmdd_hhmmss.ems. The date/time stamp part of the file name indicates when the file was created. The information in these files is stored in a binary format and must be read using the dumplog utility. You will need to open a DOS Command Prompt window and change to the <drive>:\ICM\<customer_instance>\CTIOS1\logfiles directory in order to use dumplog on the CTI OS Server log files. For information on how to use dumplog, refer to the ICM Administration Guide for Cisco ICM Enterprise Edition.

When reporting a problem, it is generally very helpful to provide the logs for the timeframe in which the problem occurred. This is Cisco?s ?window? into the activity that is taking place at the time of the problem. Try to provide all files that cover the needed timeframe. Do this by looking at the timestamp in the filename to find out when they were created and by looking at the modification timestamp in Windows Explorer to see the last time a given file was written to.

How to Set Trace Levels

Trace levels for the server processes can be found in the registry under:

```
 HKEY_LOCAL_MACHINE\Software\Cisco Systems, Inc.\
ICM\<Customer Instance>\CTIOS1\EMS\CurrentVersion
\Library\Processes\CTIOS\EMSTraceMask
```

Caution: The default value for the trace masks is 0x20003. Changing this value can have a serious impact on server performance. It should only be modified by experienced field personnel or at the request of Cisco support personnel.

Trace masks can be combined in order to uniquely log the messages you are interested in. For example:

- TRACE_MASK_CONNECTION | TRACE_MASK_METHOD_LOW

- ♦ logs low important method messages of the connection layer
- TRACE_MASK_EVENTFILTER | TRACE_MASK_METHOD_HIGH
  - ♦ logs high important method messages of the EventFilter component

Setting CTI OS Server Trace Levels

The default trace mask is 0x3 in all releases except in release 7.0(0) where it is set to 0x20003.

Setting the trace mask high (for example: 0xf or higher) has a big impact on both the CTI OS server performance and the call completion rate. As a result, only set the trace mask high when debugging a problem. When the required logs have been collected, turn the trace mask back to its default value.

During load testing, ensure the default trace mask is set across all the PG components, and not just the CTI OS server.

Note: The PG and CTI OS server are co-resident.
For troubleshooting purposes, set the CTI OS Server trace mask to:

- 0x0A0F for Release 6.0 and earlier
- 0x20A0F for Release 7.0, and 7.1(1)
- 0x60A0F for Release 7.1(2) and later

Setting CTI OS Client Trace Levels

The default trace mask is:

- 0x7 for Release 6.0 and earlier
- 0x40000307 in Release 7.x.x

Setting the trace mask to 0x40000307 is OK when running one, two or three agent desktops on the systems, but it impacts the system if a load test is running and CILTest is used to simulate CTI OS agents. When running load testing using CILTest, setting a low trace mask value (0x3 for example) is more appropriate.

Set the trace mask higher than 0x40000307 only when debugging a problem. When the required logs are collected, set the trace mask back to its default value.

For troubleshooting purposes, set the CTI OS Client trace mask to 0xFFF

Setting Point-to-Point Silent Monitor Trace Levels (Client only)

For troubleshooting purposes, and to get all events as described before for the client plus Silent Monitor Sniffer, Decoder events/requests, and communication of forwarded VoIP (send/receive of encapsulated VoIP in UPD stream); set the trace mask to 0x20003E0F.

Note: All CTI OS component trace settings are essentially the same. For trace masks specific to .NET CIL, refer to the CTI OS Developer?s Guide.

Trace Mask Name	Trace Mask Number	Description
TRACE_MASK_ALWAYS	0x00	Used across all CTI OS components.

		Only the necessary messages need to use this trace mask. Examples: initialization phases, important error messages, ... The message is always printed, regardless of the trace mask setting.
TRACE_MASK_CRITICAL	0x01	Used across all CTI OS components. Only the necessary messages need to use this trace mask. Examples: critical error messages, ... The message is printed only if the trace mask bit is set.
TRACE_MASK_WARNING	0x02	Used across all CTI OS components. This trace mask bit is used for warning messages. The message is printed only if the trace mask bit is set.
TRACE_MASK_EVT_REQ_HIGH	0x04	Used across all CTI OS components. This trace mask bit is used to display high important events, confs, and requests names without their parameters. The message is printed only if the trace mask bit is set.
TRACE_MASK_EVT_REQ_HIGH_PARM	0x08	The same as TRACE_MASK_EVT_REQ_HIGH, but the messages include events, confs, and requests parameters.
TRACE_MASK_EVT_REQ_AVG	0x10	Used across all CTI OS components. This trace mask bit is used to display average important events, confs, and requests names without their parameters. The message is printed only if the trace mask bit is set.
TRACE_MASK_EVT_REQ_AVG_PARM	0x20	The same as TRACE_MASK_EVT_REQ_AVG, but the messages include events, confs, and requests parameters.
TRACE_MASK_EVT_REQ_LOW	0x40	Used across all CTI OS components. This trace mask bit is used to display low important events, confs, and requests names without their parameters. The message is printed only if the trace mask bit is set.

TRACE_MASK_EVT_REQ_LOW_PARM	0x80	The same as TRACE_MASK_EVT_REQ_LOW, but the messages include events, confs, and requests parameters.
TRACE_MASK_METHOD_HIGH	0x0100	Used across all CTI OS components. This trace mask bit is used to display high visibility method entry/exit messages. The message is printed only if the trace mask bit is set.
TRACE_MASK_METHOD_HIGH_LOGIC	0x0200	Used across all CTI OS components. This trace mask bit is used to display high visibility method logic messages. The message is printed only if the trace mask bit is set.
TRACE_MASK_METHOD_AVG	0x0400	Used across all CTI OS components. This trace mask bit is used to display average visibility method entry/exit messages. The message is printed only if the trace mask bit is set.
TRACE_MASK_METHOD_AVG_LOGIC	0x0800	Used across all CTI OS components. This trace mask bit is used to display average visibility method logic messages. The message is printed only if the trace mask bit is set.
TRACE_MASK_METHOD_LOW	0x1000	Used across all CTI OS components. This trace mask bit is used to display low visibility method entry/exit messages. The message is printed only if the trace mask bit is set.
TRACE_MASK_METHOD_LOW_LOGIC	0x2000	Used across all CTI OS components. This trace mask bit is used to display low visibility method logic messages. The message is printed only if the trace mask bit is set.
TRACE_MASK_METHOD_MAP	0x4000	It is used in CTI OS server for the following components: 1. ServiceBroker component 2. ObjectMap component

		3. CtiServerDriverLib component This trace mask bit is used to display map information. These maps hold objects such as call, agent, skillgroup, arguments, and supervisor. The message is printed only if the trace mask bit is set.
TRACE_MASK_CONTROLS	0x8000	Not used.
TRACE_MASK_EVENTFILTER	0x010000	It is used in CTI OS server for the EventFilter component. This trace mask bit is used to display messages in the EventFilter component. The message is printed only if the trace mask bit is set.
TRACE_MASK_MESSAGEPASSING	0x020000	Used in CTI OS server for the IOCPConnectionMgr component (this components accepts CTI OS client connections). For Release 7.1(2) and later, this trace mask bit is used to display all the messages coming and going between CTI OS server and CTI OS client. For Releases 7.0, 7.0 SRx, 7.0 SRx ESx, and 7.1(1) this trace mask bit is used to display all the messages coming and going between CTI OS server and CTI OS client, and between CtiServer and CTI OS Server The message is printed only if the trace mask bit is set.
TRACE_MASK_CG_MESSAGEPASSING	0x040000	It is used in CTI OS server for the Driver piece of the ServiceBroker component. Valid only for Release 7.1(2) and later. This trace mask bit is used to display all the messages coming and going between CtiServer and CTI OS server (for example: events, requests and confs). The message is printed only if the trace mask bit is set.
TRACE_MASK_MESSAGEQUEUEING	0x080000	Used in CTI OS server for the MultiThread algorithm in the ServiceBroker component. It is valid only in Release 7.1(2) and later.

		This trace mask bit is used to display messages in the ServiceBroker component for the MultiThread algorithm. The message is printed only if the trace mask bit is set.
TRACE_MASK_SECURITY	0x100000	Used in the security and the connection libraries that are used in CTI OS server and CTI OS client. Valid only for Release 7.0(0) and later. This trace mask bit is used to display messages related to security in the security and the connection libraries. The message is printed only if the trace mask bit is set.
TRACE_MASK_ARGREFCOUNTING	0x400000	Used in the arguments library that is used in CTI OS server and CTI OS client. This trace mask bit is used to display messages related to the arguments reference count (Addref and Release) in the arguments library. The message is printed only if the trace mask bit is set.
TRACE_MASK_REFCOUNTING	0x800000	Used for all objects that are used in CTI OS server and CTI OS client (for example: call, agent, skillgroup, supervisor objects). This trace mask bit is used to display messages related to the object reference count (Addref and Release) in the CTI OS server and CTI OS client. The message is printed only if the trace mask bit is set.
TRACE_MASK_ARGS_METHODS	0x01000000	Used in the arguments library used in CTI OS server and CTI OS client. This trace mask bit is used to display the name of the class::method of the arguments library. The message is printed only if the trace mask bit is set.
TRACE_MASK_ARGS_LOGIC	0x02000000	Used in the arguments library used in CTI OS server and CTI OS client. This trace mask bit is used to display the logic of the methods of the arguments library.

		The message is printed only if the trace mask bit is set.
TRACE_MASK_PACKETS_METHODS	0x04000000	Used in the connection library used for the peer CTI OS server connection and for CTI OS client connection. This trace mask bit is used to display the name of the class::method of the connection library. The message is printed only if the trace mask bit is set.
TRACE_MASK_PACKETS_LOGIC	0x08000000	Used in the connection library used for the peer CTI OS server connection and for CTI OS client connection. This trace mask bit is used to display the logic of the methods of the connection library. The message is printed only if the trace mask bit is set.
TRACE_MASK_SERIALIZE_DUMP	0x10000000	Not used.
TRACE_MASK_SOCKETS_DUMP	0x20000000	Used in the connection library used for the connection between: 1. peer CTI OS servers 2. CTI OS client and CTI OS Server 3. CtiServer and CTI OS Server This trace mask bit is used to display the packets coming from the network. The message is printed only if the trace mask bit is set.
TRACE_MASK_THREADING	0x40000000	Used in the: 1. CilTest 2. CtiServerDriverLib used in CTI OS server 3. Util library used by the CTI OS server and CTI OS client This trace mask bit is used to display thread information (for example: thread name, thread status, ...). The message is printed only if the trace mask bit is set.
TRACE_MASK_CONNECTION	0x80000000	Used in all the connection layers used by CTI OS server and CTI OS client. This trace mask bit is used to display logic information of the connection layers. The message is printed only if the trace mask bit

		is set.
TRACE_LEVEL_MAJOR	0x000000ff	Not used.
TRACE_LEVEL_EVENT_REQ	0x0000ff00	Not used.
TRACE_LEVEL_METHOD	0x00ff0000	Not used.
TRACE_LEVEL_MEMORY	0xff000000	Not used.

Taking CTI Toolkit Logs

The trace log name and location for client processes can be found under the following registry keys:

```
 HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.
\CTIOS\Logging\TraceFileName
```

The default filename is CTIOSClientLog. Logfiles are created using the convention <TraceFileName>.<username>.mmdd.hhmmss.log. The files will be created in the current directory of the executing program, such as the directory into which the AgentDesktop is installed. You can provide a fully qualified path for the TraceFileName if you wish to store the files in a different location. For example, setting the value to "C:\Temp\CTIOSClientLog" would put the logfiles in the directory "C:\Temp" using the naming convention CTI OSClientLog.<username>.mmdd.hhmmss.log. Client trace files are simple ASCII text and can be opened with a conventional text editor such as Notepad.

How to Set Trace Levels

Trace levels for client processes, such as the AgentDesktop phone, can be found in the registry under:

```
 HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.
\CTIOS\Logging\TraceMask
```

 **Caution**: The default value for the trace masks is 0x40000307. Changing this value can have a serious impact on client performance. It should only be modified by experienced field personnel or at the request of Cisco support personnel.

CTI OS FAQs

This section provides answers to some frequently asked questions about CTI OS.

What is the basic CTI OS architecture?

CTI OS provides end-user CTI functionality in an ICM system. On the ICM side it connects to the CTI Server. CTI Server typically runs on a PG (Peripheral Gateway). On the end-user side CTI OS provides an agent desktop application, a supervisor desktop application, and a programming interface to develop CTI custom applications.

The CTI OS system consists of three major components

1. CTI OS Server
2. CTI Toolkit Agent Desktop
3. CTI Toolkit IPCC Supervisor Desktop (only on Cisco IPCC Enterprise).

The CTI OS Server connects to the CTI Server via TCP/IP. Depending on call and agent load (see product specification), CTI OS Server resides on the same physical machine as the CTI Server. The CTI OS server consists of the CtiosServerNode.exe executable.

In CTI Server terms, the CTI OS Server establishes an "All Events" or "Bridge" Mode connection to the CTI Server (as opposed to a "Client" Mode connection). CtiosServerNode handles CTI Toolkit connections (such as the connection to the Agent Desktop) over TCP/IP. CtiosServerNode is a "nodemanaged" component (see ICM documentation) and can therefore be started and stopped via the ICM Service Control Panel.

The main task of the CTI OS server is to do the heavy lifting of CTI messaging. It creates CTI objects (agents, calls, skillgroups, ?) and exposes these objects and selected event messages to CTI Toolkits. It also abstracts all switch specific behavior for clients, exposing the same interfaces to CTI Toolkits for all supported switches.

The CTI Toolkit Agent Desktop and CTI Toolkit IPCC Supervisor Desktop run on desktop computers and provide a user interface to CTI OS for Agents and Supervisors. The user interface includes a softphone for agentstate control, call control, handling of call context data and a chat interface. The supervisor functionality for IPCC includes monitoring and controlling agent states of monitored agents (logout, make ready), as well as barge-in and intercept functions.

The CTI Toolkit Agent Desktop and CTI Toolkit IPCC Supervisor Desktop are built upon the Client Interface Library (CIL). Developers can write custom applications using the published interfaces of CIL. The CIL is available in C++, COM (called CTIOSClient), .NET and Java, and as Active-X controls.

CTI OS supports a centralized configuration mechanism. Most parameters can be configured via the system registry on CTI OS server machine. The configuration settings will be downloaded by the CTI Toolkit application (for example, the Agent Desktop), when it connects to CTI OS server and requests them.

CTI OS will typically be installed in a duplex mode, with two CTI OS servers running in parallel. CTI OS desktop applications will randomly connect to either server and automatically fail over to the other server if the connection to their original CTI OS server fails. CTI OS can also run in a simplex mode with all clients connecting to one server (although the duplex mode is preferred because it supports fault tolerance).

Which switches are supported by CTI OS? What are PeripheralTypes?

CTI OS provides a switch-independent user interface via its softphone application. To accomplish this, some parts of the CTI OS Server must be specialized to support each switch (also referred to as Peripherals or ACDs).

To support a switch, the CTI OS system must be configured with the PeripheralID and PeripheralType of each switch. PeripheralIDs are deployment-specific, and can be found in the ICM configuration.

Peripheral Vendor (Name)	**Peripheral Type Value**
Cisco IPCC System	23
Cisco IPCC	17
Cisco IPCC Hosted Edition	17
Lucent/Avaya Definity ECS	5
Aspect Call Center ACD	1
Alcatel 4400 ACD	13
Nortel Meridian ACD	2
Nortel Symposium	16
Rockwell Spectrum ACD	7
Siemens Hicom (North American version only)	11

What is a Connection Profile?

A Connection Profile stores all of the information needed for a CTI OS Softphone to select a peripheral (switch) to log into. The information includes the Logical Name of the phone switch, the logical hostname or IP address of a CTI OS server (or pair of CTI OS Servers) that provide the service to that peripheral, and the ICM's PeripheralID that is used to track that Peripheral.

The Connection profiles are stored in the registry on the CTI OS server under HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\CTIOS\ CTIOS_<InstanceName>\ ctios1\EnterpriseDesktopSettings\All Desktops\Login\ConnectionProfiles\Name.

When the CTI OS Server is first installed, the Setup program creates a default Connection Profile called "Peripheral type name" (for example, IPCC) with information collected from the Setup prompts. You can rename the default profile to any name you like and change its properties in the registry editor (REGEDIT), and add more Connection Profiles by re-running Setup to create a new default profile. Alternatively, you can export the registry tree for your Connection Profiles to a flat (.reg) file, and edit the profiles using Notepad. Then, double-click on the .reg file to reload it into your registry.

What happens at softphone startup and login?

When the softphone starts up, the following steps are executed:

1. The CTI toolkit agent desktop or CTI toolkit supervisor desktop looks at the System registry of the local client desktop machine and reads the HostName or IP address of the Configuration machine.
2. The relevant configuration values include CTIOSA and CTIOSB and are located at HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\CTI

Desktop\CTIOS.

1. The CTI toolkit agent desktop or CTI toolkit supervisor desktop randomly connects to one of the two Configuration machines and downloads the CTI OS connection profiles and all other configuration settings.
2. These configuration settings are located in the registry of the Configuration machine under the following key HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\CTIOS\

CTIOS_<InstanceName>\ ctios1\EnterpriseDesktopSettings.

1. The CTI toolkit agent desktop or CTI toolkit supervisor desktop is now ready to accept login requests and the Login button is enabled.

When the user clicks login, the following steps are executed:

1. The login dialog is presented and the user can select one of the previously downloaded connection profiles and may enter additional required data.
2. When the user clicks OK, the softphone randomly attempts to connect with one of the two CTI OS servers defined in the user selected connection profile.
3. Upon successful connection, the client sends a SetSessionMode request to CTI OS server. This request sets a message filter for the agent ID.

4. When the CTI OS server receives a SetSessionMode request it does two things.
    1. It determines the current state of the system including the relevant peripheral and sends SetAgentModeEvent, including the status, to the client.
    2. It sends a QueryAgentStateRequest to CTI Server on behalf of the client to determine the current agent state on the switch.
5. When the Client receives SetAgentModeEvent it updates its own system status. This allows the client to inform the user if the system is offline and the login request is postponed.
6. When the client receives a QueryAgentStateConf, the client then sends a SetAgentStateRequest to login.
7. When CTI OS Server receives a login request, it snapshots the agent, logs in the agent if it is not already logged in, snapshots the agent's device, and snapshots any calls on the device. This builds the complete state of the agent.
8. Using the information obtained from the snapshots the softphone is updated to reflect the agent state and any calls. At this point the agent is fully logged in.

Why is TimeInState on the Agent Real Time Status window sometimes black and sometimes red?

If an agent remains in a certain state for longer than 10 minutes, the TimeInState column will turn red to bring this agent to the supervisor's attention. If an agent changes state, the TimeInState column will be reset to 0 and turn black again.

How can I change the update interval of SkillgroupStatistics and AgentStatistics?

The update interval for SkillgroupStatistics and AgentStatistics is set to 10 seconds by default. This means that every 10 seconds, the CTI OS server will request statistics from the CTI Server and send them to any connected Agent and Supervisor Desktops, where they will be displayed. The update interval can be changed in the system registry on the system where the CTI OS Server is installed by modifying the following keys:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\CTIOS\
CTIOS_<InstanceName>\ ctios1\Server\Agent\PollingIntervalSec

HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\CTIOS\
CTIOS_<InstanceName>\ctios1\Server\SkillGroup\
```

PollingIntervalSec

Setting the update interval to zero (0) will disable statistics entirely. This is only supported for Release 4.6.2 and later.

Besides the update interval of CTI OS, the ICM configuration has its own separate update interval to compute and store statistics (see ICM documentation). To prevent unnecessary network traffic, the CTI OS interval must not be smaller than the ICM interval.

How can I customize which columns are displayed in the callappearance, agentstatistics and skillgroupstatistics grids?

The procedure for how to customize the columns in the grids is explained in the CTI OS System Manager's Guide for Cisco Unified ICM/Contact Center Enterprise & Hosted, Release 7.5(1), Chapter 4.

How can I disable statistics minimization?

Release 4.6.2 and later: If you are not using the default statistics columns and are instead customizing your columns (as described in the Cisco ICM Software CTI OS System Manager's Guide, Chapter 5), the CTI OS server will only send updates for the statistics that you have configured. This is done to reduce network traffic. If you would like to disable this feature and receive ALL statistics for every update, set DisableStatsMinimization = 1 in the registry at the following keys:

```
 HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\CTIOS\
CTIOS_<InstanceName>\ctios1\
EnterpriseDesktopSettings\AllDesktops\Grid\AgentStatistics\
Columns\Number for agent statistics
```

and

```
HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\CTIOS\
CTIOS_<InstanceName>\ctios1\
EnterpriseDesktopSettings\All Desktops\Grid\SkillGroupStatistics\
Columns\Number for skill group statistics
```

The desktop softphone will still only display the configured columns, but it will receive ALL statistics.

Note: A defect in CTI OS versions earlier than 4.7 prevents skill group statistics from being minimized on monitor mode applications. This problem was fixed in Releases 4.7 and later. An optional DisableMonitorModeStatsMinimization setting in the SkillGroupStatistics key shown above can be used to disable minimization of skill group statistics for monitor mode applications using Release 4.7 and later.

Why does the column definition of the callappearance, agentstatistics or skillgroupstatistics grid not change after updating the registry?

The changes only become active when the CTI OS server and the client application (for example, softphone, Supervisor desktop) are shutdown and restarted.

How do I change the header and column width of a displayed column?

The procedure for how to customize the columns in the registry is explained in the *CTI OS System Manager's Guide for Cisco Unified ICM/Contact Center Enterprise & Hosted*, Release 7.5(1), Chapter 4. A column key can have Header and Width string values (plus other values as explained in the CTI OS System Manager's Guide for Cisco ICM/IPCC Enterprise & Hosted Editions).

Header: enter custom header; same as "Type" by default Width: 70 screen units by default (for example, 140 will double the standard width)

Why does CallsQNow on IPCC not display the current number of calls in the queue?

To look at the number of calls in queue on an IPCC switch, look at RouterCallsQNow. CallsQNow is supported only on legacy ACD switches.

How can one change column 1 in skillgroupstatistics?

Column 1 of the skillgroupstatistics grid always displays the skillgroupnumber and cannot be changed. The column header can be edited.

How can the network traffic caused by CTI OS statistics be reduced?

There are several ways to reduce the amount of traffic caused by the CTI OS Agent and Skillgroup Statistics messages.

- Turning off skill group or agent statistics for all agents. This can be done separately for agent and skillgroup statistics via a registry setting for each on the machine that hosts the CTI OS server. Setting the PollingIntervalSec to 0 in the registry keys listed below will disable that particular set of statistics.

    ◊ For Agent Statistics

```
HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\CTIOS\CTIOS_<InstanceName>\ctios1 \Server\Agent\Pc
```

    ◊ For Skillgroup Statistics

```
HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\CTIOS\CTIOS_<InstanceName>\ctios1 \Server\SkillGrc
```

- Expanding the update interval between statistics. The same registry keys indicated above, specify the update interval between statistics on the client in seconds if set to a value different from 0.

  For example, if PollingIntervalSec is set to 30 (default is 10) at HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\CTIOS\CTIOS_<InstanceName>\ctios1\Server\SkillGroup\, the client will see skillgroup statistics refresh every 30 seconds.
  For Agentstatistics, however, for reasons that the agent statistics is likely to change after the call, it is suggested that you continue to use the default behavior (see below), which ignores PollingIntervalSec at HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\CTIOS \CTIOS_<InstanceName>\ctios1\Server\Agent\PollingIntervalSec.

- Reducing the number of skillgroups overall and the number of skillgroups per agent. This can be accomplished via the Agent Explorer or Skillgroup Explorer config tool, which is part of the ICM Configuration Manager.
- Reducing the number of specific statistics fields being sent to the client desktop. By default, the CTI OS server only sends the statistics required for display on the CTI Toolkit. The procedure to customize which fields are displayed on the client is explained in the *CTI OS System Manager's Guide for Cisco Unified ICM/Contact Center Enterprise & Hosted*, Release 7.5(1), Chapter 4.
- Turning off statistics for some agents while leaving them on for others. In CTI OS 4.7 a new registry key was introduced to allow disabling of statistics on a per agent basis. This is done by creating a connection profile (see section on connection profiles in the CTI OS systems manager guide) for which statistics are disabled and direct some agents to use this connection profile, while others use a different connection profile with statistics enabled.

  The relevant keys are:

    ◊ DisableAgentStatistics
    ◊ DisableSkillgroupStatistics

A value of 1 indicates that the statistics are disabled for this connection profile, while a value of 0 indicates they are enabled (default).
The keys are located at:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\CTIOS\CTIOS_<InstanceName>\ctios1\EnterpriseDesktc
Name\<ConnectionProfileName>
```

- Poll for Agent statistics only at end of call. If the registry key "PollForAgentStatsAtEndCall" located at

```
HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\CTIOS\CTIOS_<InstanceName>\ctios1\Server\Agent\
```

is set to 1, PollingIntervalSec described above will be ignored and agent statistics will only be sent when a call ends. This is the recommended method (and the default behavior), since most statistics fields are only updated when a call ends. Some statistics, like TimeLoggedinToday, TimeNotReadyToday and TimeReadyToday are updated on the client independently until a new message arrives from CTI server. If PollForAgentStatsAtEndCall is set to 0, PollingIntervalSec will become effective and determine the update interval as described.

Which logs are required to diagnose a Silent Monitor issue, and what is the recommended TraceMask?

Silent Monitor is largely a client based feature and uses CTI OS server to signal the start and stop of Silent Monitor sessions as well as reporting status. See the *Obtaining Logs for Support* section above for details of how to retrieve logs.

The following logs are required to diagnose a problem:

- Client log from Supervisor Desktop (CtiosClientlog)
- Client log from Agent Desktop (CtiosClientLog)
- CTIOS server log (from both CTI OS servers in a duplexed environment, retrieved with dumplog)

The default TraceMask of 0x40000307 for CTI Toolkits and 0x20003 for CTI OS server are sufficient for high level issues. A TraceMask of 0xF0F is recommended for CTI Toolkit detailed troubleshooting and a TraceMask of 0xA0F is recommended for CTI OS server detailed troubleshooting.

What Version of CTI OS Supports Security?

The versions of CTI OS that support Security are:

- CTI OS Server 7.0
- C++ CIL and COM CIL of CTI Toolkit 7.0

Note: 1. If old CTI Toolkit tries to connect to CTI OS Server 7.0 with security ON, then the connection will fail.

Note: 2. If Java CIL or .NET CIL client try to connect to CTI OS Server 7.0 with security ON, then the connection will fail.

How do you know if Security is ON/OFF on CTI OS Server?

The value of SecurityEnabled registry key is specified in the following table:

SecurityEnabled Registry Key Value	Security Setting
0	OFF
1	ON

This SecurityEnabled registry value exists under the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems,Inc.\Ctios\CTIOS_<InstanceName>\CTIOS1\Server\Security
```

Note: If security is ON in one CTI OS Server, and this server has peers, then security must be turned ON in the peers as well.

How do you configure security on the CTI OS server and CTI Toolkit?

See the *CTI OS System Manager's Guide for Cisco Unified ICM/Contact Center Enterprise & Hosted*, Release 7.5(1) for further details on this procedure.

What files are used by Security, and where are they located?

The CTI OS files that are used by Security can be classified as CTI OS Server Security files and CTI Toolkit Security files.

The CTI OS Server Security Files: If the CTI OS Server is installed under ?<drive>:\ICM\<instance name>\<CTIOS Component name>? directory, then the following security files will be copied to ?<drive>:\ICM\<instance name>\<CTIOS Component name>\Security? directory.

CTI OS Server Security Files	Description
CtiosClient.pem (if CTI OS Server has a peer)	This file contains the CTI Toolkit certificate, and its private key.
CtiosClientPWD.dat (if CTI OS Server has a peer)	This file contains the password which is used to encrypt the CTI Toolkit private key. This password is saved in cipher text.
PrivateClientKey.pem (if CTI OS Server has a peer)	This file contains RSA private key which is used to decrypt/encrypt the password that is saved in CtiosClientPWD.dat.
CtiosServer.pem	This file contains CTI OS Server certificate, and its private key.
CtiosServerPWD.dat	This file contains the password which is used to encrypt the CTI OS Server private key. This password is saved in cipher text.
PrivateServerKey.pem	This file contains RSA private key which is used to decrypt/encrypt the password that is saved in CtiosServerPWD.dat.
MonitorPWD.dat	This file contains the monitor mode password in cipher text.
PrivateMonitorKey.pem	This file contains RSA private key which is used to decrypt/encrypt the password that is saved in MonitorPWD.dat.
CtiosDH1024.pem	This file contains Diffie-Hellman-Parameters (1024 bit).

CtiosRootCert.pem	This file contains CA certificate.
openssl.exe	This is the OpenSSL command line tool which is used when troubleshooting is necessary.
dbghelp.dll	This system DLL is needed to run openssl.exe
atl71.dll	This system DLL is needed to run openssl.exe
mfc71.dll	This system DLL is needed to run openssl.exe
mfc71d.dll	This system DLL is needed to run openssl.exe
msvcp71.dll	This system DLL is needed to run openssl.exe
msvcp71d.dll	This system DLL is needed to run openssl.exe
msvcr71.dll	This system DLL is needed to run openssl.exe
msvcr71d.dll	This system DLL is needed to run openssl.exe

CTI Toolkit Security Files: If the CTI Toolkit is installed under ?<drive>:\Program Files\Cisco Systems, Inc.\CTIOS Client? directory, then the following security files will be copied to ?<drive>:\Program Files\Cisco Systems, Inc.\CTIOS Client\Security? directory.

CTI OS Server Security Files	**Description**
CtiosClient.pem	This file contains the CTI Toolkit certificate, and its private key.
CtiosClientPWD.dat	This file contains the password which is used to encrypt the CTI Toolkit private key. This password is saved in cipher text.
CtiosRootCert.pem	This file contains CA certificate.
PrivateClientKey.pem	This file contains RSA private key which is used to decrypt/encrypt the password that is saved in CtiosClientPWD.dat.
openssl.exe	This is the OpenSSL command line tool which is used when troubleshooting is necessary.
dbghelp.dll	This system DLL is needed to run openssl.exe.
atl71.dll	This system DLL is needed to run openssl.exe.
mfc71.dll	This system DLL is needed to run openssl.exe.
mfc71d.dll	This system DLL is needed to run openssl.exe.
msvcp71.dll	This system DLL is needed to run openssl.exe.
msvcp71d.dll	This system DLL is needed to run openssl.exe.
msvcr71.dll	This system DLL is needed to run openssl.exe.
msvcr71d.dll	This system DLL is needed to run openssl.exe.

What Type of Certificate Authority (CA) was used to sign both CTI OS Server and Client Certificate Requests?

The value of CAType registry key is specified in the following table:

CAType Registry Key	**Type of CA Used**
1	Self-signed CA

2	Third-party CA

In the CTI OS Server, this registry value exists under the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\Ctios\CTIOS_<InstanceName>\CTIOS1\Server\Security
```

In CTI Toolkit, this registry value exists under the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems, Inc.\CTI Desktop\CtiOs
```

Both CTI OS Server and CTI Toolkit certificate requests need to be signed by the same CA. There are two types of CA.

- A self signed CA which is created by running ?CreateSelfSignedSetupPackage.exe? program
- A third party CA.

So, if the CTI OS Server certificate request is signed by a self signed CA, then all CTI Toolkit certificate requests must be signed by a self signed CA.

How do we display information about a Certificate?

The following table lists the options for display information for a Certificate.

Certificate Display Information	Options
Display the contents of a certificate	openssl x509 -in c CtiosClient.pem -noout -text
Display the certificate serial number	openssl x509 -in CtiosClient.pem -noout -serial
Display the certificate subject name	openssl x509 -in CtiosClient.pem -noout -subject
Display the start and expiry dates	openssl x509 -in CtiosClient.pem -noout -dates

What type of Peripheral does Multi-Tenancy/Multi-Instance CTI OS support?

Multi-Tenancy/Multi-Instance CTI OS only supports the IPCC Hosted Edition.

Can I install Multi-Tenancy/Multi-Instance CTI OS on a regular IPCC?

No. Multi-Tenancy/Multi-Instance CTI OS only supports the IPCC Hosted Edition.

Can C++ CIL, COM CIL, Java CIL, and .NET CIL Clients connect to Multi-Tenancy/Multi-Instance CTI OS Servers?

Yes.

Does Multi-Tenancy/Multi-Instance CTI OS support Siebel?

No. It does not support Siebel.

Does Multi-Tenancy/Multi-Instance CTI OS support Security?

Yes it does. Turning security ON in one of the CTI OS servers does not affect the other servers that are running on the same machine.

Note: 1. If old CTI Toolkit tries to connect to CTI OS Server 7.0 with security ON, then the connection will fail.

Note: 2. If Java CIL or .NET CIL client try to connect to CTI OS Server 7.0 with security ON, then the connection will fail.

Do all CTI OS Servers (in a Multi-Instance environment) running on the same machine use the same ListenPort?

No. Each CTI OS Server must have a unique listen port. The ?ListenPort? value of a specific CTI OS server can be found under the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems,Inc.\Ctios\CTIOS_<InstanceName>\CTIOS1\Server\connectior
```

How many CTI OS Servers (in a Multi-Instance environment) can I have installed on one machine?

Up to 10 CTI OS Servers can be installed on one machine.

Note: Multi-Instance CTI OS supports up to 10 instances, and each instance listens on a unique TCP port. Hence, all 10 ports should be accessible via a firewall.

Can I control one CTI OS Server (in a Multi-Instance environment) without affecting the others?

Yes. You can manage CTI OS Servers independently. You can stop, start, and recycle one CTI OS server without affecting the others.

Does each CTI OS Server (in a Multi-Instance environment) have its own log files?

Yes it does. The log files can be located in ?<drive>:\ICM\<instancename>\CTIOS1\logfiles? directory.

Can I change the trace mask of one CTI OS Server (in a Multi-Instance environment) without affecting the others?

Yes, there is a trace mask for every CTI OS Server. You can change the trace mask through the registry. The ?EMSTraceMask? trace mask value of a specific CTI OS server can be found under:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Cisco Systems,Inc.\ICM\CTIOS_<InstanceName>\CTIOS1\EMS\CurrentVersion\
```

Can two or more CTI OS Servers (in a Multi-Instance environment) that reside on the same machine be connected to the same CtiServer?

No. The relation between CTI OS Server and CtiServer is one to one. So, one CTI OS Server can be connected to only one CtiServer.