

Cisco CallManager Service Crash

Document ID: 19122

Introduction

Prerequisites

- Requirements

- Components Used

- Conventions

Cisco CallManager Service Crash Description

- Determine the Type of Crash

Information to Gather and Provide to Cisco Technical Support

- Unexpected Event Crash

- Lack of Resource Crash

Check Settings on Backup Utility to Avoid High CPU

- Intercluster Routing Loops Can Cause High CPU Spikes

Set Up Performance Monitor Counter Logs

Related Information

Introduction

This document provides information about a Cisco CallManager crash, how to determine if you have experienced a crash, the information to gather and provide to the Cisco Technical Support, and how to search for Cisco CallManager crash bugs that exist.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Cisco CallManager Service Crash Description

When the Cisco CallManager service (ccm.exe) crashes, you see this message in the System Event log:

```
The Cisco CallManager service terminated unexpectedly.  
It has done this 1 time. The following corrective action  
will be taken in 60000 ms. Restart the service.
```

Other messages you can see in the event of a crash are:

```
Timeout 3000 milliseconds waiting for Cisco  
CallManager service to connect.
```

The Cisco CallManager failed to start due to the following error.
The service did not respond to the start or control request in a timely fashion.

At this time, when devices like Cisco IP Phones and gateways unregister from the Cisco CallManager, users experience delayed dial tone, and/or the Cisco CallManager server freezes due to high CPU. Refer to Cisco CallManager Event Logs for event log messages not included here.

The Cisco CallManager service can crash due to one of these reasons:

1. An unexpected event occurs in the Cisco CallManager service. This crash adds an entry to the Dr.Watson log that exists and a User.dmp is generated in the folder C:\Documents and Settings\All Users\Documents\DrWatson.
2. The Cisco CallManager service does not have enough resources like CPU or memory in order to function. Generally, the CPU utilization in the server is at 100 percent at that time.

Dependent on the type of crash you experience, you need to gather different data that helps you and Cisco Technical Support determine the root cause of the crash.

Determine the Type of Crash

If you perform a search on your Cisco CallManager after the crash for a file called Drwtsn32.log and open it, look at the most recent entry in order to see if an entry for the ccm.exe has been added. Open the Dr. Watson log in Notepad, go to the bottom of the file and search for *Application exception occurred*, which takes you to the latest crash.

This is an example of header for a crash entry in drwtsn32.log.

```
Application exception occurred:
App: (pid=680)
When: 3/8/2003 @ 14:01:06.978
Exception number: e06d7363
```

Next to the date of the crash there is a PID, if that PID corresponds to the PID for ccm.exe in the task list then you know that the Cisco CallManager crashed.

The task list in the drwtsn32.log looks similar to this:

```
PID PROCESS
8 System.exe
212 SMSS.exe
240 CSRSS.exe
264 WINLOGON.exe
292 SERVICES.exe
304 LSASS.exe
424 termsrv.exe
520 svchost.exe
560 msdtc.exe
696 DLLHOST.exe
736 Ipvmsapp.exe
752 DLLHOST.exe
824 AudioTranslator.exe
848 RisDC.exe
860 LogoutService.E.exe
884 DCX500.exe
936 svchost.exe
980 LLSSRV.exe
1028 sqlservr.exe
1112 ntpd.exe
```

```
1140 rcmdsvc.exe
1172 regsvc.exe
1176 mstask.exe
1204 SNMP.exe
1244 WinMgmt.exe
1260 cpqnimgt.exe
1284 cqmgserv.exe
1296 cqmgestor.exe
1308 sysdown.exe
1372 cqmghost.exe
1524 aupair.exe
1552 sqlagent.exe
 276 svchost.exe
2400 inetinfo.exe
2412 explorer.exe
2752 sqlmangr.exe
2700 taskmgr.exe
2704 mmc.exe
 680 ccm.exe
 868 DRWTSN32.exe
```

Note: In this example, the PID = 680, which from the list, corresponds to ccm.exe.

If there is no list of the PIDs, look at the timestamp of the last entry of the drwtsn32.log and the timestamp of the error in the Event Log. See the Cisco CallManager Service Crash Description section. If they are the exact same time, it is likely that you experienced an Unexpected Event Cisco CallManager crash.

The stack trace makes a crash unique, which is why the entire drwtsn32.log file in Information to Gather and Provide to Cisco Technical Support is requested.

If the PID for the day of the crash is not ccm.exe or the timestamp does not correspond, then you most likely have run into a lack of resource crash, or a crash of another process.

Information to Gather and Provide to Cisco Technical Support

Unexpected Event Crash

If you experience an Unexpected Event crash, complete these steps in order to gather information to provide to the Cisco Technical Support.

1. Collect Cisco CallManager traces 15 minutes before and after the crash.

Traces are located at C:\Program Files\cisco\trace\ccm.

2. Collect SDL traces 15 minutes before and after the crash.

Traces are located at C:\Program Files\cisco\trace\sdl\ccm.

3. Choose **Start > Programs > Administrative Tools > Event Viewer** in order to gather System and Application Event log files from the Event Viewer.
4. Click on **System Log** and choose **Action > Save Log as** and save the log.

Do the same of the Application Log.

5. Ensure that the SdlMaxUnhandledExceptions parameter is set to **0 (zero)** for each Cisco CallManager.
6. Collect the Dr. Watson log located at C:\Documents and Settings\All Users\Documents\DrWatson. The name of the file is Drwtsn32.log.

7. Collect the User.dmp file located at C:\Documents and Settings\All Users\Documents\DrWatson.

Note: These files can be very large. Be sure to zip them before you send them to Cisco Technical Support. It is also important to note that these files hold the information the Cisco Technical Support engineer and developers need in order to determine the cause of the crash.

8. Open the Dr. Watson log in Notepad and proceed to the Determine the Type of Crash section in order to find out if your crash is a known issue.

Lack of Resource Crash

If you experience a Lack of Resource crash, complete these steps in order to gather information to provide to Cisco Technical Support.

1. Collect Cisco CallManager traces 15 minutes before and after the crash. Traces are located at C:\Program Files\cisco\trace\ccm.
2. Collect SDL traces 15 minutes before and after the crash. Traces are located at C:\Program Files\cisco\trace\sdl\ccm.
3. Collect perfmon traces if available. If they are not available, start to collect these and track memory usage and CPU usage for each process that runs on the server. See the Set Up Performance Monitor Counter Logs section in order to set up perfmon traces. These help in the event of another lack of resources crash.

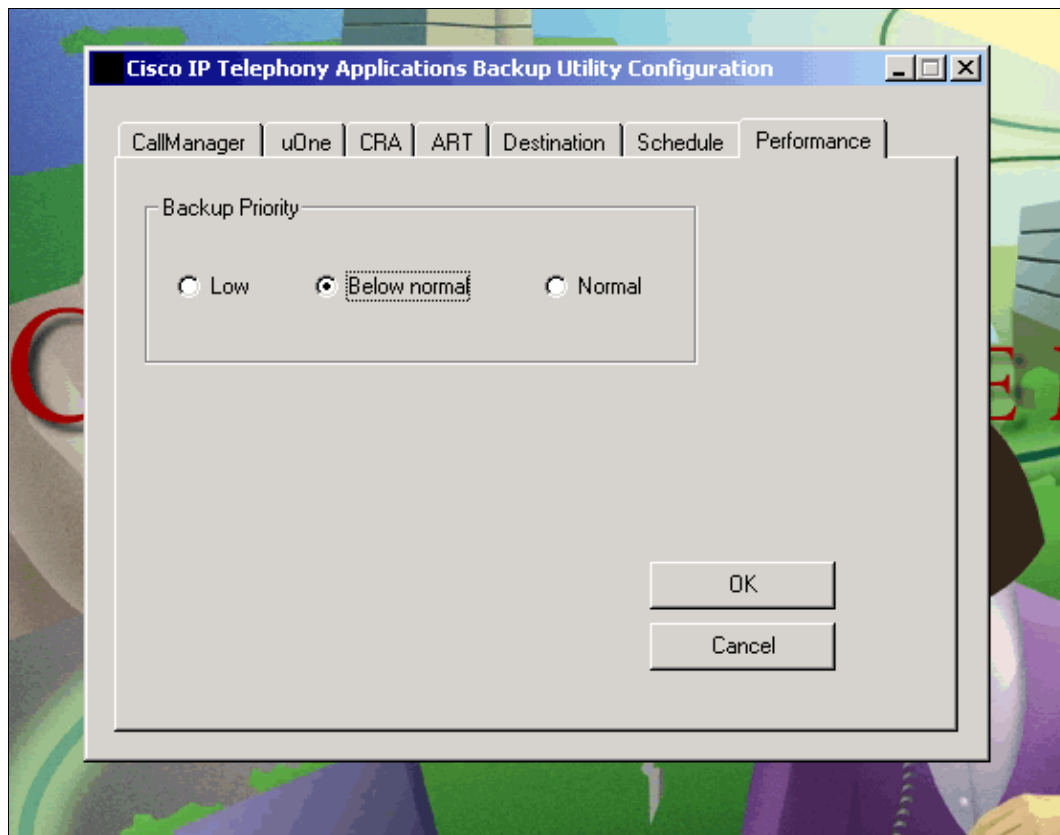
Check Settings on Backup Utility to Avoid High CPU

Ensure that you run the latest Cisco IP Telephony Applications Backup in order to avoid a system crash due to the Cisco IP Telephony Applications Backup which can run for an extended period of time at high CPU utilization. If you run Cisco CallManager 3.1(3a)spC and later or Cisco CallManager 3.2(1)spA and later, per Cisco bug ID CSCdt91655 (registered customers only) , the new Backup utility run at **low** priority by default.

You can download the latest version of Cisco IP Telephony Applications Backup from the Voice Software download page (registered customers only) under Cisco CallManager.

Note: If you perform a virus scan on the BARS staging directory C:\STI when you run the backup, you can cause CPU spikes. Disable virus scan on C:\STI in order to avoid high CPU utilizations.

Prior to this change, the previous versions used a tab called Performance in order to change the Base Priority of the process that runs the Cisco IP Telephony Applications Backup application. Change the performance to **below normal** or **low** in order to ensure that this process does not compete with other processes, which run at **normal** Base Priority, for CPU, such as CCM.exe.



Intercluster Routing Loops Can Cause High CPU Spikes

Intercluster trunk looping can be caused by a misconfigured route pattern. This can cause Cisco CallManager to run high CPU for a long period of time or crash the server. Cisco CallManager has added logic in the H.225 device (for trunk device only) in order to monitor the number of transit calls outstanding in order to solve this problem. Transit call is the call that Cisco CallManager receives the setup request for (or sends the setup request for) and does not yet receive or send the first backward message. For example, *call proceeding*, *call progress*, *alert*, *connect*, or *release complete*. Cisco Call Manager runs a five-second timer in order to monitor the transit call queue for the H.225 trunk device. If the number of the transit call queue entries is greater than a pre-defined threshold, then, for a period of time (default 30 seconds), all new incoming or outgoing call request to that H225 trunk device are rejected by sending *release completed* messages with cause code *Switch System Congestion*.

Due to this behavior of Cisco CallManager, these errors can be seen in application log of Cisco CallManager.

- **The Error** ICTCallThrottlingStart error message indicates that the Cisco CallManager cannot handle calls for the indicated H.323 device because of a route loop over the H.323 trunk.
- **The Error** ICTCallThrottlingEnd error message indicates that the Cisco CallManager resumed call handling for the indicated H.323 device (stopped due to the route loop created over the H.323 trunk).

Stop routing loop between clusters in order to avoid these errors. Refer to Cisco CallManager Loop Avoidance Guide to Best Practices for more information on Cisco CallManager Loop avoidance.

Set Up Performance Monitor Counter Logs

Complete these steps in order to gather counters for the crash in order to verify the processes that run and the amount of CPU and memory that are consumed.

1. Choose **Start > Programs > Administrative Tools > Performance**.
2. From the Performance Monitor, choose **Performance Logs > Alerts > Counter Logs**.
3. Choose **Action > New log settings** and enter a name for the counter log.
4. Under counters, choose **add**.

Use the local computer counters and make sure that you configure this directly on the Cisco CallManager that experiences the crash.

5. Under Performance Object, choose **Process**.
6. Under Select Counters, highlight **List > Select Instances**, and choose these counters and associated instances:
 - ◆ **% Processor Time / All Instances**
 - ◆ **ID Process / All Instance**
 - ◆ **Virtual Bytes / All Instances**
 - ◆ **Private Bytes / All Instances**
7. Under Sample Data Every, set the interval to **2** and the units as **seconds**.
8. From the Log Files tab, make sure that the log file type is **Text File – CSV**. Also note where these are located. The default is C:\PerfLogs.
9. Choose a log file limit of **20,000 Kb**.
10. Perform these actions from **Schedule**:
 - a. Choose **start log manually** in order to start the log.
 - b. Choose **when the 20,000 Kb log file is full** in order to stop the log.
 - c. When the log closes, choose **Start a new log file** and then click **OK**.
11. Choose the created counter log in order to start to log. Then choose **Action > Start**.

Note: Over time, if you enable these performance monitor logs, it generates a large number of files and utilizes a large amount of disk space. Therefore, it is necessary to keep an eye on this and, if it is, zip up the older logs and/or move them from the local drive.

Related Information

- [Set Up Cisco CallManager Traces for Cisco Technical Support](#)
- [Voice Technology Support](#)
- [Voice and Unified Communications Product Support](#)
- [Recommended Reading: Troubleshooting Cisco IP Telephony](#)
- [Technical Support & Documentation – Cisco Systems](#)

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2009 – 2010 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Oct 30, 2006

Document ID: 19122
