

# Cómo identificar un reinicio del Cisco CallManager como una caída o servicio apague

## Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenciones](#)

[La diferencia entre el Cisco CallManager causa un crash y apaga](#)

[Desperfectos](#)

[Apaga](#)

[Cómo señalar las caídas del Cisco CallManager al Soporte técnico de Cisco](#)

[Información Relacionada](#)

## [Introducción](#)

Este documento describe la diferencia entre una caída del Cisco CallManager y un servicio apague. El documento también proporciona el procedimiento para señalar una caída del Cisco CallManager y para permitir al [Soporte técnico de Cisco](#) para resolver problemas el problema.

## [prerrequisitos](#)

### [Requisitos](#)

No hay requisitos específicos para este documento.

### [Componentes Utilizados](#)

La información que contiene este documento se basa en estas versiones de software:

- Cisco CallManager 3.x y 4.0

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

### [Convenciones](#)

Consulte [Convenciones de Consejos Técnicos de Cisco](#) para obtener más información sobre las convenciones sobre documentos.

# La diferencia entre el Cisco CallManager causa un crash y apaga

## Desperfectos

Un bug en el código del Cisco CallManager causa las caídas del CallManager. Hay tres tipos principales de caídas:

- Violaciones de acceso
- Divide By Zero
- Excepciones desconocidas

Las caídas generan las entradas del Dr.Watson, que se añaden al final del fichero al final del archivo existente del Dr.Watson. Las caídas también generan los archivos del user.dmp. La ubicación de estos archivos es usuarios de C:\Documents and Settings\All \ documentos \ DrWatson.

El nombre del archivo del Dr.Watson, que es un archivo de texto, es drwtsn32.log.

Elija el **drwtsn32** de la ventana del funcionamiento para configurar las configuraciones.

## Cómo leer un archivo del Dr.Watson

Complete estos pasos para leer el archivo del Dr.Watson:

1. Busque para la palabra “cuando”, en minúsculas, y encuentre la fecha y hora en la cual el problema ocurrió.El archivo del Dr.Watson registra todas las caídas de la aplicación. Algunos expedientes de caída pueden no ser caídas del Cisco CallManager. Los ejemplos de los expedientes de caída que no son caídas del Cisco CallManager incluyen RisDC.exe y aupair.exe.
2. Después de que usted localice la fecha y hora del problema, localice el número de proceso del identificador (PID), y busque la lista de tareas para determinar que la aplicación causó un crash.La lista de tareas aparece en el ejemplo en este paso.En este ejemplo, la aplicación que causó un crash tiene un PID de 752, y el nombre de la aplicación es

SCAN32.exe:Application exception occurred:

```
App: (pid=752) When: 9/1/2000 @ 10:23:40.836 Exception number:
c0000005 (access violation) *----> System Information <----* Computer Name: CISCO-
8VCUWBLUJ User Name: SYSTEM Number of Processors: 1 Processor Type:
x86 Family 6 Model 8 Stepping 3 Windows 2000 Version: 5.0 Current Build:
2195 Service Pack: None Current Type: Uniprocessor Free Registered
Organization: Cisco Systems Inc. Registered Owner: Cisco User *----> Task List <---
-* 0 Idle.exe 8 System.exe 144 smss.exe 168 csrss.exe 164 winlogon.exe 216
services.exe 228 lsass.exe 336 ibmpmsvc.exe 380 svchost.exe 424 svchost.exe 576
regsvcs.exe 592 MSTask.exe 924 Explorer.exe 992 cmd.exe 972 msiexec.exe 928 tp4mon.exe
856 ibmpmsvc.exe 852 ltmsg.exe 408 RunDll32.exe 428 RunDll32.exe 328 PDirect.exe 620
TP98.exe 968 tphkmgr.exe 948 PRPCUI.exe 668 AUTOCHK.exe 744 tponscr.exe 868 KIX32.exe
520 spoolsv.exe 1164 Avsynmgr.exe 1136 VsStat.exe 1192 Vshwin32.exe 1224 Mcshield.exe 1024
MCUPDATE.exe 752 SCAN32.exe 1292 drwtsn32.exe 0 _Total.exe
```

3. Si la caída es una caída del Cisco CallManager, observe el número de la excepción para determinar el tipo de caída.**Nota:** Rutee al equipo de desarrollo apropiado una caída de la aplicación que no sea una caída del Cisco CallManager, en caso necesario.Application exception occurred:

```
App: (pid=752)
```

When: 9/1/2000 @ 10:23:40.836

Exception number: **c0000005 (access violation)** En este ejemplo, el número de la excepción es **c0000005**, que es una violación de acceso. Esta violación de acceso significa que la aplicación intentó a memoria de acceso fuera del Límite de memoria de la aplicación que el conjunto del sistema operativo. Otro tipo de desperfecto posible es divisoria por cero. Pues este ejemplo muestra, el número de la excepción para la divisoria por cero es

c0000094:Application exception occurred:

App: (pid=1564)

When: 1/7/2003 @ 13:16:15.051

Exception number: **c0000094 (divide by zero)** El tipo de desperfecto puede también ser excepción desconocida. Pues este ejemplo muestra, número de excepción para una excepción desconocida es **e06d7363**:Application exception occurred:

App: (pid=2784)

When: 12/10/2002 @ 09:02:58.202

Exception number: **e06d7363** Cuando usted determina si una caída es una violación de acceso, divisoria por cero, o excepción desconocida, usted puede corresponder con una caída con un bug Cisco existente. Si usted no encuentra ningún emparejamiento, el ingeniero de desarrollo tiene un buen comienzo para determinar qué ocurrió.

4. Busque bajo **cuando** sección del archivo para la palabra **INCIDENTE** para comenzar a definir la "firma" de la caída. **Nota:** El **INCIDENTE** aparece con mayúsculas. Esta sección del **INCIDENTE** del archivo contiene seis pedazos de información importante, que son: El número del hilo que experimentó el problema El contenido de los registros para este hilo a la hora de la caída La función que ejecutó a la hora de la caída El enunciado de código ensamblador que dio lugar a la caída El stack detrás localiza que muestra los direccionamientos de las funciones que ejecutado, en la orden, momentos antes de la caída El volcado sin procesar del stack, que proporciona más información sobre cuál estaba en el stack de tiempo de ejecución a la hora de la caída Este código proporciona un ejemplo de una caída del Cisco CallManager que sea una caída de la violación de acceso. El texto en **negrita** resalta los seis elementos críticos, así como la palabra **INCIDENTE**, que marca esta sección del archivo:

```
State Dump for Thread Id
0x998 !--- This number is the number of the thread that experienced the problem.
eax=00cae82c ebx=02070000 ecx=00e95da0 edx=346984d8 esi=34698970 edi=346984d8 eip=77fcb9b3
esp=05cef34c ebp=05cef358 iopl=0         nv up ei ng nz na pe cy cs=001b  ss=0023  ds=0023
es=0023  fs=003b  gs=0000             efl=00000283 !--- This provides the contents of the
registers at the time of the crash. function: RtlSizeHeap !--- This function executed at
the time of the crash.
77fcb992 0f87aafeffff      jnbe   RtlFreeHeap+0x20f
(77fcb842)          77fcb998 807d1400      cmp    byte ptr [ebp+0x14],0x0      ss:
0650c92a=??          77fcb99c 0f8586300000  jne   RtlZeroHeap+0x327 (77fcea28)
77fcb9a2 57                push   edi          77fcb9a3 53                push   ebx
77fcb9a4 e8646cfbfff      call  RtlConsoleMultiByteToUnicodeN+0x348 (77f8260d)
77fcb9a9 8b4f0c           mov    ecx,[edi+0xc]      ds: 34eb5aaa=00003781
77fcb9ac 8b4708           mov    eax,[edi+0x8]      ds: 34eb5aaa=00003781
77fcb9af 3bc1            cmp    eax,ecx          77fcb9b1 8901            mov
[ecx],eax          ds: 00e95da0=00cae82c FAULT ->77fcb9b3 894804      mov
[eax+0x4],ecx      ds: 014cbdfc=ec810000 !--- This is the assembly code statement that
resulted in the crash.
77fcb9b6 744d            jz    77fd4405          77fcb9b8
8a4705            mov   al,[edi+0x5]      ds: 34eb5aaa=81          77fcb9bb
a804             test  al,0x4           77fcb9bd 0f8521310000  jne   RtlZeroHeap+0x3e3
(77fcea4)          77fcb9c3 8a4605            mov   al,[esi+0x5]      ds:
```

```

34eb5f42=d5          77fcb9c6 2410          and    al,0x10          77fcb9c8
a810                test    al,0x10        77fcb9ca 884705          mov
[edi+0x5],al        ds: 34eb5aaa=81        77fcb9cd 0f8555030000      jne
RtlSizeHeap+0x3ef (77fcbd28) 77fcb9d3 0fb70f          movzx  ecx,word ptr
[edi]               ds: 346984d8=0093      77fcb9d6 8b4510          mov
eax,[ebp+0x10]      ss: 0650c92a=???????? *----> Stack Back Trace <----* !--- This
shows, in order, the addresses of the functions that executed !--- just before the crash.
FramePtr ReturnAd Param#1 Param#2 Param#3 Param#4 Function Name 05CEF358 77FCB733
02070000 34698970 05CEF3D0 00000000 ntdll!RtlSizeHeap 05CEF400 7800115C 02070000 00000000
34698978 05CEF454 ntdll!RtlFreeHeap 05CEF448 00C0304F 34698978 00545EC2 34698978 34698978
!free 05CEF460 00B66F85 00000001 00B6626C 033B3D58 025A6720 !<nosymbols> 05CEFF34 018E736B
025A6720 77E964CB 033C6B20 033C6B20 !<nosymbols> 05CEFF80 780060CE 033B3D58 77E964CB
00000018 033C6B20 !ACE_OS_Thread_Adapter:: invoke µµ 05CEFFEC 00000000 00000000 00000000
00000000 00000000 kernel32!TlsSetValue *----> Raw Stack Dump <----* !--- This provides more
information about what was on the run-time stack !--- at the time of the crash. 05cef34c
00 00 07 02 70 89 69 34 - 00 00 00 00 00 00 f4 ce 05 ....p.i4..... 05cef35c 33 b7 fc 77
00 00 07 02 - 70 89 69 34 d0 f3 ce 05 3..w....p.i4.... 05cef36c 00 00 00 00 54 f4 ce 05 -
78 89 69 34 20 67 5a 02 ....T...x.i4 gZ. 05cef37c 44 5b e3 09 94 f3 ce 05 - 30 e6 b5 00
fc f3 ce 05 D[.....0..... 05cef38c 38 29 6a 09 40 5b e3 09 - a8 f3 ce 05 65 e5 b5 00
8)j.@[.....e... 05cef39c fc f3 ce 05 38 29 6a 09 - 40 5b e3 09 c4 f3 ce 05
....8)j.@[..... 05cef3ac 39 e2 b5 00 57 92 89 01 - 30 db 55 02 f5 50 5b 00
9...W...0.U..P[. 05cef3bc e0 f3 ce 05 cc f3 ce 05 - 0f 4f 5b 00 e0 f3 ce 05
.....O[..... 05cef3cc 00 00 07 02 19 00 00 00 - 01 f4 ce 05 f8 2b cf 21
.....+! 05cef3dc f8 2b cf 21 01 f1 ce 05 - 28 ff ce 05 70 f3 ce 05
.+!....(...p... 05cef3ec 98 ef ce 05 38 f4 ce 05 - a7 9d fb 77 90 26 f8 77
....8.....w.&.w 05cef3fc 01 00 00 00 48 f4 ce 05 - 5c 11 00 78 00 00 07 02
....H...\..x.... 05cef40c 00 00 00 00 78 89 69 34 - 54 f4 ce 05 04 fa ce 05
....x.i4T..... 05cef41c 20 67 5a 02 02 00 00 00 - 64 00 00 00 5c 00 00 00
gZ.....d...\... 05cef42c fe 08 00 00 00 00 00 00 - 98 ef ce 05 28 ff ce 05
.....(... 05cef43c b8 ff 00 78 50 32 03 78 - ff ff ff ff 60 f4 ce 05
...xP2.x....`... 05cef44c 4f 30 c0 00 78 89 69 34 - c2 5e 54 00 78 89 69 34
00..x.i4.^T.x.i4 05cef45c 78 89 69 34 34 ff ce 05 - 85 6f b6 00 01 00 00 00
x.i44....o..... 05cef46c 6c 62 b6 00 58 3d 3b 03 - 20 67 5a 02 98 f6 e6 36 1b..X=i.
gZ....6 05cef47c 98 f6 e6 36 00 00 00 00 - 00 00 00 00 00 00 00 00 ...6..... Estos

```

seis bits de la información constituyen la parte de, pero no a toda de, la firma de la caída. El resto de la información que define la firma es:El tipo de caída (violación de acceso, divisoria por cero, o excepción desconocida)Las instrucciones de seguimiento más recientes del Signal Distribution Layer (SDL) que ejecutaron antes de la caídaNota: El archivo del último SDL que tenía uso antes de la caída, así como el archivo del Dr.Watson, los attaches a cualquier caída introducen errores de funcionamiento.Los attaches de esta información de firma (el archivo del último SDL, el archivo más reciente del Cisco CallManager, y el archivo del Dr.Watson) al Distributed Defect Tracking System (DDTS) registran cuando usted crea una nueva caída DDTs.Si usted hace juego la nueva caída con un DDTs que exista y tenga ya la misma causa raíz, esta información es lo mismo:El tipo de excepciónEl nombre de la función que ejecutó a la hora de la caídaLos nombres de las funciones en el stack detrás localizanNota: Estos nombres no aparecen siempre en un archivo del Dr.Watson.El enunciado de código ensamblador que aparece al lado de la etiqueta de plástico del INCIDENTE Las líneas de seguimiento del último SDL, que deben ser muy similaresEl contenido de los registros, de las direcciones de memoria, y de la otra información puede diferenciar de la información en otro DDTs que exista, incluso si la caída tiene la misma causa raíz. Los direccionamientos varían si usted funciona con una diversa versión del Cisco CallManager. Si usted funciona con la misma versión del Cisco CallManager, los direccionamientos en la sección del código del ensamblaje y en la sección del seguimiento de pila son lo mismo.

5. Recoja estos archivos para hacer el debug de la caída:drwtsn32.loguser.dmpEl último SDL y seguimiento del CallManager archivos, a partir de aproximadamente 5 minutos antes de la

caída y de 5 minutos después del reinicio. archivos del proglog **Nota:** Recoja estos archivos solamente en las versiones 3.2 del Cisco CallManager y posterior. Registros de acontecimientos, sistema y aplicación, si está disponible. Registros del monitor de rendimiento (perfmon), si está disponible.

## Error de DBLException

Usted ve este mensaje de error en el log de aplicaciones del editor del CallManager de Cisco y del suscriptor:

```
Error: DBLException - DBL Exception.  
  ErrorCode: 8  
  ExceptionString: Invalid parameter  
  UNKNOWN_PARAMNAME:Text: addDevice  
  App ID: Cisco CallManager  
  Cluster ID: XXXX-Cluster  
  Node ID: 192.168.0.2  
Explanation: Severe database layer interface error occurred.  
Recommended Action: Contact TAC..
```

O:

A COM error occurred during processing. (6)

Details:

```
Error No. -2147219962 (0x80040606):  
CDBLException Dump: [COM Error] COM Error Description = []
```

Este tipo de error ocurre cuando un teléfono del IP se rechaza del registro o debido a una suscripción quebrada entre las bases de datos del editor y suscriptor. Este problema se puede solucionar usando la herramienta del DBLHelper. Para más información sobre el DBLHelper, refiérase [con el DBLHelper para restablecer una suscripción a SQL quebrada del clúster del Cisco CallManager](#).

Este error puede también ocurrir debido a la caída del servicio del monitor de la capa de la base de datos de Cisco. Realice estos pasos para resolver el problema:

1. Vaya al **Start (Inicio) > Programs (Programas) > Administrative Tools (Herramientas administrativas)** > a los **servicios del componente**.
2. Expanda **Servicios de componente > Equipos > Mi PC > Aplicaciones COM+**.
3. Comience **MSDTC** el servicio (del Coordinador de transacciones distribuidas), si muestra parado.

## Apaga

El otro tipo de reinicio del Cisco CallManager es un apagar. Un apagar es cuando el Cisco CallManager no puede actuar eficazmente y se cierra. Apaga la caída en dos categorías:

- [Tiempos de espera de inicialización](#)
- [Temporizador SDL y muertes de secuencia del router SDL](#)

Si el Cisco CallManager se cierra, usted encuentra un código de motivo del apagar en las líneas de seguimiento últimas del seguimiento de CallManager. Aquí tiene un ejemplo:

```
03/22/2003 14:32:16.562 Cisco CallManager|CallManagerFailure - Indicates  
some failure in the Cisco CallManager system. Host name of hosting
```

node.:NEROCM1 IP address of hosting node.:172.27.27.224 Reason code.:4 Additional Text  
[Optional]: App ID:Cisco CallManager Cluster ID:NEROCM1-Cluster Node  
ID:172.27.27.224|<CLID::NEROCM1-Cluster><NID::172.27.27.224><CT::Alarm>

En este ejemplo, el código de motivo es 4. Esta lista proporciona apaga los códigos de motivo del código del Cisco CallManager:

```
class CallManagerFailureAlarm : public CallManagerAlarmCatalog {
public:
    enum Reason {
        Unknown = 1,
        HeartBeatStopped = 2,
        RouterThreadDied = 3,
        TimerThreadDied = 4,
        CriticalThreadDied = 5,
        DeviceMgrInitFailed = 6,
        DigitAnalysisInitFailed = 7,
        CallControlInitFailed = 8,
        LinkMgrInitFailed = 9,
        DbMgrInitFailed = 10,
        MsgTranslationInitFailed = 11,
        SupServiceInitFailed = 12,
        DirectoryInitFailed = 13
    };
};
```

La razón 1 y la razón 2 son casos pocos probables de apagados internos, mientras que las otras razones son mas comunes. La razón 3 indica que el hilo del router SDL ha parado la respuesta. La razón 4 indica que el hilo del temporizador SDL ha parado la respuesta. Las razones 5 – 13 se relacionan con el fuego del temporizador de la inicialización.

### Tiempos de espera de inicialización

Cuando el servicio CallManager de Cisco primero comienza, el hilo del monitor del proceso del CallManager (CMProcMon) comienza. Entonces, el comienzo del hilo del MmmanInit, que spawn/genera el resto de procesos. Después, el comienzo del hilo del router SDL. Este hilo maneja las señales que envían a partir de un proceso a otro. Los tres de los hilos comienzan al mismo tiempo. El hilo del CMProcMon y el hilo del router SDL son ascendentes y se ejecutan mientras que el hilo del MmmanInit comienza para arriba otros procesos.

CMProcmon y el SDL necesitan ser en servicio mientras que el MmmanInit comienza para arriba los diversos procesos. El hilo del MmmanInit comienza estos procesos, en esta orden:

1. Base de datos (ProcessDb)**Nota:** ProcessDb es una interfaz del Cisco CallManager al código de la capa de la base de datos (DBL).Al mismo tiempo, el código del MmmanInit también comienza vario otro Cisco CallManager interno, los procesos independientes. Estos procesos incluyen H225Handler, MGCPBhHandler, y al jefe jerárquico.
2. Regiones
3. AARNeighborhood
4. Ubicaciones
5. Ruta Plan
6. Análisis de dígitos
7. Control de llamadas
8. Servicios suplementariosLas características incluyen Call park (Detención de llamadas), adelante, la conferencia, y la transferencia.
9. Dispositivo
10. Directorio



11. Administrador del Calling Search Space (CSSManager)

12. Administrador del Time Of Day (TODManager)

La realización de estas tareas ocurre en una serie. Cada una de las doce tareas tiene un temporizador que se asocia a la tarea. Este temporizador comienza cuando la tarea comienza. Si los fuegos del temporizador antes de que la tarea complete, Cisco CallManager para e imprimen una traza SDL que lea:

*Critical thread death: name of the timer which fired*

Esta lista muestra cada uno de los temporizadores, así como la señal SDL que comienza el temporizador y la señal SDL que para el temporizador. Las señales de "InitDone" aparecen en la traza SDL si usted ha fijado los niveles de traza apropiadamente. (Usted fijó SdlTraceTypeFlags en 0x8000CB15.)

Estos temporizadores predeterminados se basan en la versión del CallManager de Cisco 4.1(2). Si usted funciona con diversa versión, pudo ser levemente diferente.

1. El tiempo de la inicialización de la base de datos (valores por defecto a 900 segundos) - la Señal de inicio por este tiempo es la señal del "comienzo" enviada al proceso del MmmanInit. Usted ve esto en la traza SDL.
2. Hora de inicialización de las regiones (valores por defecto a 120 segundos).
3. Hora de inicialización de AARNeighborhoods (valores por defecto a 90 segundos).
4. Hora de inicialización de las ubicaciones (valores por defecto a 90 segundos).
5. Hora de inicialización de la ruta Plan (valores por defecto a 600 segundos).
6. Hora de inicialización de la análisis de dígitos (valores por defecto a 900 segundos).
7. Hora de inicialización del Control de Llamadas (valores por defecto a 90 segundos).
8. La hora de inicialización de los servicios suplementarios (valores por defecto a 900 segundos) - la Señal de inicio es CcInItDone y la señal del extremo es SsInItDone.
9. Hora de inicialización del dispositivo (valores por defecto a 360 segundos).
10. Hora de inicialización del directorio (valores por defecto a 90 segundos)
11. Hora de inicialización de CSSManager (valores por defecto a 900 segundos).
12. Hora de inicialización de TODManager – (valores por defecto a 900 segundos).

Después de todo las tareas son completas, Cisco CallManager abren los links SDL en los servicios de CallManager que se ejecutan en otros Nodos en la red. El Cisco CallManager también abre los links SDL en los servicios del administrador de Integración de telefonía de computadora (CTI) que se ejecutan en el mismo nodo o diversos Nodos en la red.

Entonces, el MmmanInit envía una señal de CMInitComplete de nuevo al hilo del CMProcMon. Cuando el CMProcMon primero comienza, comienza un temporizador puesto en hard-code 60-minute para la inicialización del Cisco CallManager. El temporizador tiene el nombre CMInitComplete\_WaitTime. (Este temporizador no es un parámetro de servicio; el temporizador no es configurable.) Si el hilo del CMProcMon no recibe la señal de CMInitComplete en el plazo de 60 minutos, el Cisco CallManager para y publica una instrucción de seguimiento que lea:

*Timed out waiting for CMInitComplete signal*

Si de las doce tareas de inicialización falla, o si el tiempo total para estas tareas excede 60 minutos, el Cisco CallManager para.

**Nota:** El temporizador de CMInitComplete\_WaitTime era una vez duro cifrado a 10 minutos. Este código duro ha cambiado a 60 minutos como parte del Id. de bug Cisco [CSCdx31622](#) ([clientes registrados solamente](#)). El cambio ingresó el 3.1(3) dirigiendo el tren especial (ES), con ES 38 como el comienzo. El cambio está también en 3.2(2) el tren ES, con ES 11 como el comienzo, y

en el Cisco CallManager 3.3.

Si usted experimenta los problemas con un fuego del temporizador de la inicialización, usted puede ser que necesite solamente aumentar la configuración del temporizador para resolver el lanzamiento. Si este cambio no resuelve el problema, el problema puede ser un tiempo de respuesta lento de la base de datos que hace la operación medir el tiempo hacia fuera. Collect detalló las trazas DBL, así como las trazas SDL y del Cisco CallManager, en caso necesario.

Recoja estos archivos para hacer el debug de un problema de inicialización:

- Detallado seguimiento del CallManager
- Traza SDL **Nota:** Fije el SdlTraceTypeFlags a 0x8000CB15.
- Traza detallada DBL

### Temporizador SDL y muertes de secuencia del router SDL

El hilo del router SDL es el hilo más importante de la ejecución dentro de la aplicación del Cisco CallManager. Controla el envío de las señales de Procesamiento de Llamadas. El hilo del CMPProcMon marca la salud del hilo del router SDL una vez cada dos segundos. Las trazas del Cisco CallManager muestran esta revisión médica, como usted puede ver en estas declaraciones:

```
02/05/2003 00:30:32.790 Cisco CallManager|CMPProcMon - -----Entered Router
Verification|<CLID::USNYTSVOIPB01-Cluster><NID::10.2.40.11>
```

```
02/05/2003 00:30:32.790 Cisco CallManager|CMPProcMon - ----Exited Router
Verification|<CLID::USNYTSVOIPB01-Cluster><NID::10.2.40.11>
```

Si el hilo del CMPProcMon ingresa y sale la verificación del router, el hilo del router SDL respondió al control de salud y está muy bien.

Sin embargo, si no responde el hilo del router SDL, usted ve `mientras que loop` en seguimiento del CallManager, como este las demostraciones:

```
CMPProcMon - ----Entered While loop ++++ TimeAtWhileEntry: [some number here],
TimeBeforeSleep: [another number], TimeAfterSleep: [a third number], sleepTimeWas :
[4th number"
```

En esta situación de emergencia, el hilo del router SDL recibe los controles una vez cada segundo por un período de 20 segundos. Si el hilo responde en cualquier momento durante el período 20-second, el funcionamiento normal reanuda, y la salud del hilo del router SDL recibe de nuevo la verificación cada 2 segundos. Si, sin embargo, el hilo del router SDL no responde a los controles durante los 20 segundos, la aplicación del Cisco CallManager apaga. Esta declaración aparece en la traza SDL:

```
000177508| 01/12/31 07:28:40.389| 001| AlarmErr |
| | | | |
| AlarmClass: CallManager, AlarmName: CallManagerFailure, AlarmSeverity: Error
AlarmMessage: , AlarmDescription: Indicates some failure in the Cisco CallManager
system.,
AlarmParameters: HostName:CCM-PUB, IPAddress:10.5.162.180, Reason:3, Text:, AppID:Cisco
CallManager, ClusterID:CCM-PUB-Cluster, NodeID:10.5.162.180,
```

Note el código de motivo 3 en el texto de esta instrucción de seguimiento. El código significa que el hilo del router SDL ha parado la respuesta, así que Cisco CallManager ha apagado.

La causa más probable de un cierre del router SDL es una falta de recursos del sistema. Otra aplicación ha utilizado la mayoría o todo el CPU durante un largo periodo del tiempo, por lo



menos 20 segundos. Esta actividad es de porqué los monitores de rendimiento son vitales hacer el debug de este tipo apagan.

El otro tipo de apaga para explorar es el cierre de secuencia del temporizador SDL. Un cierre de secuencia del temporizador SDL ocurre cuando el diferencial entre el reloj interno del Cisco CallManager y el reloj externo del sistema operativo excede 16 segundos. Cuando sucede el cierre de secuencia del temporizador SDL, esta traza aparece en seguimiento del CallManager:

```
03/22/2003 14:32:16.562 Cisco CallManager|CallManagerFailure - Indicates
some failure in the Cisco CallManager system. Host name of hosting
node.:NEROCM1 IP address of hosting node.:172.27.27.224 Reason code.:4 Additional Text
[Optional]: App ID:Cisco CallManager Cluster ID:NEROCM1-Cluster Node
ID:172.27.27.224|<CLID::NEROCM1-Cluster><NID::172.27.27.224><CT::Alarm>
```

El Cisco CallManager marca generalmente los hilos del temporizador una vez cada segundo. El Cisco CallManager agrega 1 segundo al tiempo actual del sistema operativo y los almacenes que valoran lejos como “tiempo esperado.” Entonces, el Cisco CallManager duerme para 1 segundo. Después de que el Cisco CallManager se despierte, marca el nuevo tiempo del sistema operativo y resta el tiempo esperado. Si la diferencia entre estas dos épocas es más de 1 segundo, esta declaración amonestadora aparece en seguimiento del CallManager:

```
CMProcMon::star_sdlVerification - Test Timer exceeded minimum timer latency
threshold of 1000 milliseconds, Actual latency: 1630 milliseconds
```

La latencia real en esta declaración muestra que el hilo interno del temporizador SDL del Cisco CallManager ejecuta lento. Aquí, la diferencia entre el tiempo esperado del Cisco CallManager y el tiempo real del sistema operativo es 1.63 segundos.

Si esta diferencia excede 16 segundos, el Cisco CallManager apaga y proporciona el código de motivo del apagar de 4.

La causa más probable de un cierre de secuencia del temporizador SDL es una falta de hora de la CPU para el Cisco CallManager. Otra aplicación, tal como VirusScan o backup Sti, ha utilizado la mayor parte de a los recursos de la CPU por lo menos 16 segundos. Los registros del perfmon son vitales determinar la causa raíz de este tipo de apagan.

Si el backup de aplicaciones del Cisco IP Telephony se ejecuta durante un largo periodo del tiempo en CPU elevada la utilización, una caída del sistema puede ocurrir. Para la información sobre cómo evitar esta caída del sistema, refiérase:

- [Marque las configuraciones en la utilidad Backup \(Copias de resguardorespaldo\) para evitar CPU elevada](#) la sección de la [caída del servicio Cisco CallManager del](#) documento

Recoja estos archivos en el caso de un hilo o de un cierre de secuencia del temporizador SDL del router SDL:

- Detallado seguimiento del CallManager
- Traza SDL**Nota:** Fije el SdlTraceTypeFlags a 0x8000CB15.
- Trazas del perfmon, si está disponible, que muestran el por ciento de uso de la CPU de todos los procesos que se ejecuten en el cuadro**Nota:** Usted puede capturar estas trazas remotamente para reducir el impacto del rendimiento en el CPU del Cisco Callmanager server.

[\*\*Cómo señalar las caídas del Cisco CallManager al Soporte técnico de Cisco\*\*](#)

La diagnosis de la causa real de una caída del Cisco CallManager es difícil. Para determinar la causa y apresurar una solución, el [Soporte técnico de Cisco](#) le requiere recoger las trazas y los registros del Dr.Watson y cargar la información a las notas de [caso de soporte técnico de Cisco](#). Usted envía las notas de caso a [attach@cisco.com](mailto:attach@cisco.com) y proporciona el número de caso en el asunto del email. El procedimiento es:

1. Recoja seguimiento del CallManager los archivos a partir de 30 minutos antes y de 15 minutos después de la caída.La ubicación de las trazas es C:\Program Files\Cisco\Trace\CCM.
2. Recoja los archivos de traza SDL a partir de 30 minutos antes y de 15 minutos después de la caída.La ubicación de las trazas es C:\Program Files\Cisco\Trace\SDL\CCM.
3. Recoja el user.dmp y los archivos de drwtsn32.log.La ubicación de los archivos es usuarios de C:\Documents and Settings\All \ documentos \ DrWatson.
4. Seleccione el **Start (Inicio) > Programs (Programas) > Adminsitration Tools (Herramientas administrativas) > Event viewer (Visor de eventos)** para recopilar los archivos del sistema y del registro de eventos de aplicación del visor de eventos.Usted puede saltar este paso, si los datos de registro de acontecimientos no son necesarios. Pero, vacie el sistema y los eventos de aplicación y filtre hacia fuera solamente los eventos a partir de aproximadamente 30 minutos antes de la caída. Investigue estos eventos antes de que usted los envíe al [Soporte técnico de Cisco](#). Usted puede encontrar un evento que autorice más atención.**Precaución:** Tenga cuidado si usted utiliza el visor de eventos, una utilidad de Microsoft incorporada, de vaciar estos eventos a un archivo de texto. En un sistema que tenga CPU elevada utilización, este uso del visor de eventos puede morir de hambre fácilmente el resto de los procesos del CPU. Estos procesos incluyen el proceso de keepalive del Cisco CallManager que mantiene los registros de teléfono.Usted puede utilizar una utilidad shareware con el nombre elogdmp.exe para vaciar todas las entradas en los registros individuales a un archivo de texto. Las implicaciones de la CPU son insignificantes cuando usted utiliza la herramienta elogdmp.exe. Publique este comando de un prompt DOS:  

```
elogdmp COMPUTER_NAME Application > AppEvents.txt elogdmp COMPUTER_NAME System > SysEvents.txt
```
5. Comprima todos los archivos como archivo zip en la orden que este paso muestra antes de que usted email y copia los archivos.Utilice el WinZip versión 8 para comprimir los archivos. (Cisco tiene una licencia del sitio para esta utilidad.) Generalmente copia de archivos a una máquina local para una evaluación más rápida. Los archivos que usted comprime el uso menos espacio, y usted pueden mover estos archivos mucho más rápidamente que los formatos de archivo raw.Comprima el user.dmp y los archivos de drwtsn32.log juntos.Envíe y copie inmediatamente esto archivo zip. Proporcione una definición del síntoma descriptivo e incluya la versión del CallManager de Cisco exacta, las cargas del dispositivo apropiadas, y las versiones del Cisco IOS ® Software. Si algunas correcciones especiales son funcionando, asegúrese de que usted haga este hecho claro.Comprima el Cisco CallManager y los archivos de traza SDL juntos.Envíe y copie esto archivo zip mientras que usted espera el contacto.Comprima los registros del perfmon juntos.Envíe y copie esto archivo zip mientras que usted espera el contacto.Comprima las entradas de registro de evento juntas.Envíe y copie esto archivo zip mientras que usted espera el contacto.
6. Después de que usted haya recolectado todas las trazas y registros, comprima los archivos y envíe archivo zip a [attach@cisco.com](mailto:attach@cisco.com). Proporcione el número de caso en el asunto del email.

## Información Relacionada

- [Interrupción del servicio de Cisco CallManager](#)
- [Resolver problemas las caídas del Cisco CallManager](#)
- [Soporte de tecnología de voz](#)
- [Soporte de Productos de Voice and Unified Communications](#)
- [Troubleshooting de Cisco IP Telephony](#)
- [Soporte Técnico y Documentación - Cisco Systems](#)