

# Определение перезапуска Cisco CallManager как аварийное завершение или завершение работы службы

## Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Условные обозначения](#)

[Различие между сбоями Cisco CallManager и завершениями](#)

[Сбои](#)

[Завершения](#)

[Как сообщить о сбоях Cisco CallManager технической поддержке Cisco](#)

[Дополнительные сведения](#)

## Введение

Этот документ описывает различие между катастрофическим отказом Cisco CallManager и сервисным завершением. Документ также предоставляет процедуру, чтобы сообщить, что Cisco CallManager завершается катастрофическим отказом и позволяет [технической поддержке Cisco](#) решить проблему.

## Предварительные условия

### Требования

Для этого документа отсутствуют особые требования.

### Используемые компоненты

Сведения, содержащиеся в этом документе, касаются следующих версий программного обеспечения:

- Cisco CallManager 3.x и 4.0

Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

## Условные обозначения

[Более подробную информацию о применяемых в документе обозначениях см. в описании условных обозначений, используемых в технической документации Cisco.](#)

## Различие между сбоями Cisco CallManager и завершениями

### Сбои

Дефект в Cisco CallManager кодирует сбой CallManager причин. Существует три основных типа сбоев:

- Нарушения доступа
- Разделитесь на нуль
- Неизвестные исключения

Сбои генерируют записи Dr.Watson, которые добавлены до конца существующего файла Dr.Watson. Сбои также генерируют файлы user.dmp. Местоположение этих файлов является C:\Documents and Settings\All Users\Documents\DrWatson.

Название файла Dr.Watson, который является текстовым файлом, является drwtsn32 log.

Выберите **drwtsn32** из окна Run для настройки параметров настройки.

### **Как считать файл Dr.Watson**

Выполните эти шаги для чтения файла Dr.Watson:

1. Ищите слово, “когда”, в нижнем регистре, и находят дату и время, в которую произошла проблема. Файл Dr.Watson делает запись всех сбоев приложения. Некоторые записи катастрофического отказа могут не быть сбоями Cisco CallManager. Примеры записей катастрофического отказа, которые не являются сбоями Cisco CallManager, включают RisDC.exe и aupair.exe.
2. После того, как вы определяете местоположение даты и времени проблемы, определяете местоположение идентификатора процесса (PID) номер и ищите лист задач для определения, какое приложение завершилось катастрофическим отказом. Лист задач появляется в примере в этом шаге. В данном примере приложение, которое завершилось катастрофическим отказом, имеет PID 752, и название приложения является 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 msixec.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. Если катастрофический отказ является катастрофическим отказом Cisco CallManager, обратите внимание на номер исключения для определения типа аварии. **Примечание:** Маршрут к соответствующей команде разработки сбоя приложения, который не является катастрофическим отказом Cisco CallManager, при необходимости. Application exception occurred:

App: (pid=752)

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

Exception number: c0000005 (access violation) В данном примере номером исключения является c0000005, который является access violation. Этот access violation означает, что приложение попыталось получить доступ к памяти за пределами ограничения памяти приложения, которое установила операционная система. Другой возможный тип аварийного отказа является делением нулем. Как показано в примере номером исключения для деления нулем является c0000094: Application exception occurred:

App: (pid=1564)

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

Exception number: c0000094 (divide by zero) Тип аварийного отказа может также быть неизвестным исключением. Как показано в примере номером исключения для неизвестного исключения является e06d7363: Application exception occurred:

App: (pid=2784)

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

Exception number: e06d7363 Когда вы определяете, является ли катастрофический отказ нарушением доступа, разделитесь на нуль или неизвестное исключение, можно совпасть с катастрофическим отказом с существующей ошибкой Cisco. Если вы не находите соответствий, у инженера - разработчика есть хорошее начало для определения то, что произошло.

4. Поиск под разделом when файла для слова FAULT, чтобы начать определять "подпись" катастрофического отказа. **Примечание:** появляется в прописных буквах. Этот раздел FAULT файла содержит шесть частей важной информации, которые являются: Количество потока, который испытал проблему Содержание регистров для этого потока во время катастрофического отказа Функция, которая выполнялась во время катастрофического отказа Оператор кода компоновки, который привел к катастрофическому отказу Стек назад отслеживает, который показывает адреса функций, которые выполнялись, в заказе, непосредственно перед катастрофическим отказом Необработанный дамп стека, который предоставляет дополнительные сведения о том, что было на стеке этапа выполнения во время катастрофического отказа Этот код предоставляет пример катастрофического отказа Cisco CallManager, который является катастрофическим отказом нарушения доступа. Полужирный текст выделяет эти шесть критических элементов, а также слово FAULT, который отмечает этот раздел файла: 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 µ 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 lb..x=;.
gZ....6 05cef47c 98 f6 e6 36 00 00 00 00 - 00 00 00 00 00 00 00 00 ...6..... Эти

```

шесть битов информации составляют часть, но не весь из, подпись катастрофического отказа. Остаток информации, которая определяет подпись: Тип аварии (нарушение доступа, разделиться на нуль или неизвестное исключение), Последний Signal Distribution Layer (SDL) отслеживает операторы, которые выполнились перед катастрофическим отказом **Примечание:** Последний файл SDL, который имел использование перед катастрофическим отказом, а также файлом Dr.Watson, атташе в любом дефекте катастрофического отказа. Эти данные о подписи (последний файл SDL, последний файл Cisco CallManager и файл Dr.Watson) подключают к записи Distributed Defect Tracking System (DDTS) при создании нового DDTS катастрофического отказа. Если вы совпадаете с новым катастрофическим отказом с

DDTS, который уже существует и имеет ту же основную причину, этой информацией является то же: Тип исключения Название функции, которая выполнялась во время катастрофического отказа Названия функций в стеке назад отслеживаются **Примечание:** Эти названия не всегда появляются в файле Dr.Watson. Оператор кода компоновки, который появляется рядом с маркером `FAULT` Последние линии трассировки SDL, которые должны быть подобными Содержание регистров, адресов памяти и другой информации может отличаться от информации в другом DDTs, который существует, даже если катастрофический отказ имеет ту же основную причину. Адреса варьируются при выполнении другой версии Cisco CallManager. При выполнении той же версии Cisco CallManager адреса в разделе ассемблерного кода и в разделе трассировки стека являются тем же.

5. Соберите эти файлы для отладки катастрофического отказа: `drwtsn32 loguser.dmp` Последний SDL и файлы Трассировки Cisco CallManager, приблизительно с 5 минут перед катастрофическим отказом и спустя 5 минут после перезапуска. файлы `proglog` **Примечание:** Соберите эти файлы только в Версиях Cisco CallManager 3.2 и позже. Журналы событий, и Система и Приложение, при наличии. Монитор производительности (`perfmon`) журналы, при наличии.

## Ошибка DBLException

Вы видите это сообщение об ошибках в журнале приложения и Издателя Cisco CallManager и Абонента:

```
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..
```

Или:

```
A COM error occurred during processing. (6)
```

Details:

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

Когда IP-телефон отклонен от регистрации или из-за сломанной подписки между базами данных издателя и подписчика, этот тип ошибки происходит. Эта проблема может быть решена при помощи программного средства DBLHelper. Для получения дополнительной информации о DBLHelper обратитесь к [Использованию DBLHelper Восстановить Вышедший из строя кластер Cisco CallManager в процессе подписки SQL](#).

Эта ошибка может также произойти из-за сервисного катастрофического отказа Монитора уровня базы данных Cisco. Выполните эти шаги для решения вопроса:

1. Перейдите к Пуску> Программы> Средства администрирования> Сервисы компонента.
2. Разверните Сервисы компонента> Компьютеры> Приложения COM + Моего компьютера.
3. Запустите MSDTC (Distributed Transaction Coordinator) сервис, если это показывает,

остановился.

## Завершения

Другой тип перезапуска Cisco CallManager является завершением. Завершение состоит в том, когда Cisco CallManager неспособен работать эффективно и завершает работу себя. Завершения попадают в две категории:

- [Таймауты инициализации](#)
- [Таймер SDL и окончания потока маршрутизатора SDL](#)

Если Cisco CallManager завершает работу себя, вы находите завершение Reason code в последних нескольких линиях трассировки Трассировки CallManager . Например:

```
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>
```

В данном примере кодом причины является 4. Этот список предоставляет коды причины завершения от кода 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
    };
};
```

Обоснуйте 1, и Причина 2 редкие случаи внутренних выключений, в то время как другие причины более распространены. Причина 3 указывает, что поток маршрутизатора SDL остановил ответ. Причина 4 указывает, что поток таймера SDL остановил ответ. Причины 5–13 касаются огня таймера инициализации.

## Таймауты инициализации

Когда Сервис Cisco CallManager сначала запускается, Монитор Процесса CallManager (CMPProcMon), поток запускается. Затем поток MmmanInit запускается, который порождает все другие процессы. Затем, поток маршрутизатора SDL запускается. Этот поток обрабатывает сигналы, которые передают от одного процесса до другого. Все три из потоков запускаются в то же время. В то время как поток MmmanInit запускает другие процессы, поток CMPProcMon и поток маршрутизатора SDL подключены и работают.

В то время как MmmanInit запускает различные процессы, CMPProcMon и SDL должны быть в порядке. Поток MmmanInit запускает эти процессы в этом заказе:

1. База данных (ProcessDb)**Примечание: ProcessDb** является интерфейсом Cisco CallManager к Уровню базы данных (DBL) код.В то же время код MmmanInit также запускает много других Cisco CallManager внутренние, независимые процессы. Эти процессы включают H225Handler, MGCPBhHandler и LineManager.
2. Области
3. AARNeighborhood
4. Местоположения
5. Таблица маршрутизации
6. Анализ цифровой информации
7. Контроль вызовов
8. Дополнительные сервисыФункции включают парк вызовов, вперед, конференцию и передачу.
9. Устройство
10. Каталог
11. Менеджер пространства поиска вызова (CSSManager)
12. Менеджер времени дня (TODManager)

Выполнение этих задач происходит в серии. Каждая из этих двенадцати задач имеет таймер, который связывается с задачей. Когда задача начинается, этот таймер запускается. Если таймер срабатывает, прежде чем задача завершает, Cisco CallManager останавливает и распечатывает трассировку SDL, которая читает:

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

Этот список показывает каждый из таймеров, а также сигнал SDL, который запускает таймер и сигнал SDL, который останавливает таймер. Если у вас есть уровни set trace соответственно, сигналы "InitDone" появляются в трассировке SDL. (Вы устанавливаете SdlTraceTypeFlags в 0x8000CB15.)

Эти таймеры по умолчанию основываются на Версии Cisco CallManager 4.1 (2). Если вы выполняете другую версию, сила немного отличаются.

1. Время Инициализации базы данных (настройки по умолчанию к 900 секундам) - стартовый сигнал для на этот раз является сигналом "запуска", передаваемым процессу MmmanInit. Вы видите это в трассировке SDL.
2. Время инициализации областей (настройки по умолчанию к 120 секундам).
3. Время инициализации AARNeighborhoods (настройки по умолчанию к 90 секундам).
4. Время инициализации местоположений (настройки по умолчанию к 90 секундам).
5. Время инициализации Таблицы маршрутизации (настройки по умолчанию к 600 секундам).
6. Время инициализации Анализа цифровой информации (настройки по умолчанию к 900 секундам).
7. Время инициализации Управления вызовами (настройки по умолчанию к 90 секундам).
8. Время инициализации Дополнительных сервисов (настройки по умолчанию к 900 секундам) - стартовым сигналом является CcInitDone, и конечным сигналом является SsInitDone.
9. Время Инициализации устройства (настройки по умолчанию к 360 секундам).
10. Время инициализации каталога (настройки по умолчанию к 90 секундам)
11. Время инициализации CSSManager (настройки по умолчанию к 900 секундам).
12. Время инициализации TODManager – (настройки по умолчанию к 900 секундам).

После того, как все задачи завершены, Cisco CallManager открывает ссылки SDL на

Сервисы CallManager, которые работают на других узлах в сети. Cisco CallManager также открывает ссылки SDL на Менеджера Интеграции компьютерной телефонии (CTI) сервисы, которые работают на том же узле или других узлах в сети.

Затем MmmanInit передает сигнал CMInitComplete обратно в поток CMProcMon. Когда CMProcMon сначала запускается, он запускает 60-минутный жестко закодированный таймер для инициализации Cisco CallManager. Таймер имеет название CMInitComplete\_WaitTime. (Этот таймер не является параметром сервиса; таймер не конфигурируем.), Если поток CMProcMon не получает сигнал CMInitComplete в течение 60 минут, Cisco CallManager останавливает и делает заявление трассировки, которое читает:

```
Timed out waiting for CMInitComplete signal
```

. если кто-либо из этих двенадцати сбоев задач инициализации, или если общее время для этих задач превышает 60 минут, Cisco CallManager останавливается,

**Примечание:** Таймер CMInitComplete\_WaitTime был однажды трудно закодирован к 10 минутам. Этот твердый код изменился на 60 минут как часть идентификатора ошибки Cisco [CSCdx31622 \(только зарегистрированные клиенты\)](#). Изменение ввело 3.1 (3) серия Engineering Special (ES) с ES 38 как запуск. Изменение находится также в 3.2 (2) серия ES с ES 11 как запуск, и в Cisco CallManager 3.3.

При испытании проблем с огнем таймера инициализации вы, возможно, только должны были бы увеличить настройку таймера для решения запуска. Если это изменение не решает проблему, проблемой может быть медленное время отклика базы данных, которое заставляет операцию испытывать таймаут. Соберите подробные DBL трассировки, а также SDL и Трассировки Cisco CallManager, при необходимости.

Соберите эти файлы для отладки проблемы инициализации:

- Подробная Трассировка Cisco CallManager
- SDI-запись **Примечание:** Установите SdiTraceTypeFlags в 0x8000CB15.
- Подробная DBL трассировка

## [Таймер SDL и окончания потока маршрутизатора SDL](#)

Поток маршрутизатора SDL является самым важным потоком выполнения в приложении Cisco CallManager. Это управляет передачей сигналов обработки вызова. Поток CMProcMon проверяет состояние потока маршрутизатора SDL один раз в две секунды. Трассировки Cisco CallManager показывают этот медицинский осмотр, как вы видите в этих операторах:

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

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

Если поток CMProcMon вводит и выходит из проверки маршрутизатора, поток маршрутизатора SDL ответил на медицинский осмотр и прекрасен.

Однако, если поток маршрутизатора SDL не отвечает, вы видите `while loop` в Трассировке Cisco CallManager, поскольку это показывает:

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



[4th number"

В этой аварийной ситуации поток маршрутизатора SDL получает чеки один раз во второй срок на 20 секунд. Если поток отвечает когда-либо в течение 20-секундного периода, резюме нормальной работы, и состояние потока маршрутизатора SDL еще раз получает проверку каждые 2 секунды. Если, однако, поток маршрутизатора SDL не отвечает на проверки за эти 20 секунд, приложение Cisco CallManager завершает работу. Этот оператор появляется в трассировке 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,
```

Заметьте код причины 3 в тексте этого оператора трассировки. Код означает, что поток маршрутизатора SDL остановил ответ, таким образом, Cisco CallManager завершил работу.

Наиболее вероятная причина отключения потока маршрутизатора SDL является отсутствием ресурсов системы. Другое приложение использовало больше всего или весь ЦП в течение длительного времени времени, по крайней мере 20 секунд. Это действие состоит в том, почему мониторы производительности жизненно важны для отладки этого типа завершения.

Другой тип завершения для исследования является отключением процесса таймера SDL. Когда дифференциал между внутренними часами Cisco CallManager и внешними часами операционной системы превышает 16 секунд, отключение процесса таймера SDL происходит. Когда отключение процесса таймера SDL происходит, эта трассировка появляется в Трассировке Cisco 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>
```

Cisco CallManager обычно проверяет потоки таймера один раз во второй. Cisco CallManager добавляет 1 секунду к текущему времени операционной системы и хранит то значение как "ожидаемое время". Затем Cisco CallManager спит в течение 1 секунды. После того, как Cisco CallManager просыпается, он проверяет новое время операционной системы и вычитает ожидаемое время. Если различием между этими двумя разами составляет больше чем 1 секунду, этот оператор предупреждения появляется в Трассировке Cisco CallManager:

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

Actual latency в этом операторе показывает, что внутренний таймер SDL Cisco CallManager распараллеливает медленные выполнения. Здесь, различием между ожидаемым временем Cisco CallManager и фактическим временем операционной системы составляют 1.63 секунды.

Если это различие превышает 16 секунд, Cisco CallManager завершает работу и предоставляет код причины завершения 4.

Наиболее вероятная причина отключения процесса таймера SDL является отсутствием времени процессора для Cisco CallManager. Другое приложение, такое как VirusScan или

резервное копирование STI, использовало большинство ресурсов ЦПУ в течение по крайней мере 16 секунд. Журналы perfmon жизненно важны для определения основной причины этого типа завершения.

Если выполнения Резервной копии Cisco IP Telephony Applications в течение длительного времени времени при высокой загрузке ЦП, может произойти сбой системы. Для получения информации о том, как избежать этого сбоя системы, обратитесь к:

- [Проверьте Параметры настройки на Программе резервирования для Предотвращения раздела Высокой загрузки CPU](#) документа [Отказ сервиса Cisco CallManager](#)

Соберите эти файлы в случае потока маршрутизатора SDL или отключения процесса таймера SDL:

- Подробная Трассировка Cisco CallManager
- SDI-запись **Примечание:** Установите SdiTraceTypeFlags в 0x8000CB15.
- Трассировки perfmon, при наличии, которые показывают процент использования CPU всех процессов, которые работают на коробке **Примечание:** Можно перехватить эти трассировки удаленно для сокращения влияния на производительность на ЦП Cisco CallManager server.

## [Как сообщить о сбоях Cisco CallManager технической поддержке Cisco](#)

Диагноз фактической причины катастрофического отказа Cisco CallManager является трудным. Чтобы определить причину и ускорить решение, [техническая поддержка Cisco](#) требует, чтобы вы собрали трассировки и журналы Dr.Watson и загрузили информацию к отметкам о случае [технической поддержки Cisco](#). Вы передаете отметки о случае к [attach@cisco.com](mailto:attach@cisco.com) и предоставляете номер заявки в почтовой строке темы. Процедура:

1. Соберите файлы Трассировки Cisco CallManager с 30 минут прежде и спустя 15 минут после катастрофического отказа. Местоположение трассировок является C:\Program Files\Cisco\Trace\CCM.
2. Соберите файлы трассировки SDL с 30 минут прежде и спустя 15 минут после катастрофического отказа. Местоположение трассировок является C:\Program Files\Cisco\Trace\SDL\CCM.
3. Соберите user.dmp и drwtsn32. файлы журналов. Местоположение файлов является C:\Documents and Settings\All Users\Documents\DrWatson.
4. Выберите **Start> Programs> Administrative Tools> Event Viewer** для собирания Системных файлов и Файлов регистрации событий приложений от Просмотра событий. Если данные журнала событий не необходимы, можно пропустить этот шаг. Но, формируйте дампы Системы и Событий приложения и отфильтруйте только события приблизительно с 30 минут перед катастрофическим отказом. Исследуйте эти события, прежде чем вы передадите им к [технической поддержке Cisco](#). Можно найти событие, которое гарантирует большее внимание. **Внимание.** : Будьте осторожны при использовании Просмотра событий, встроенной утилиты Microsoft, для формирования дампа этих событий к текстовому файлу. В системе, которая имеет высокую загрузку ЦП, это использование Просмотра событий может легко исчерпать ресурсы все другие процессы от ЦП. Эти процессы включают Процесс поддержки активности Cisco

CallManager, который поддерживает регистрации телефона. Можно использовать условно бесплатное служебное программа с названием elogdmp.exe для формирования дампа всех записей в отдельных журналах к текстовому файлу. Вовлечение ЦПУ незначительно при использовании elogdmp.exe программного средства. Выполните эту команду от командной строки DOS:

```
elogdmp COMPUTER_NAME Application > AppEvents.txt  
elogdmp COMPUTER_NAME System > SysEvents.txt
```

5. Сожмите все файлы как файлы архива zip в заказе, который показывает этот шаг, прежде чем вы пошлете по электронной почте и скопируете файлы. Используйте версию WinZip 8 для сжатия файлов. (Cisco имеет корпоративную лицензию для этой утилиты.) В целом файлы копируют к локальному компьютеру для более быстрой оценки. Файлы, которые вы сжимаете, используют меньше пространства, и можно переместить эти файлы намного более быстро, чем необработанные форматы файлов. Сожмите user.dmp и файлы drwtsn32 log вместе. Сразу передайте и скопируйте этот файл архива zip. Предоставьте описательное определение симптома и включайте точную Версию Cisco CallManager, соответствующие загрузки устройства и версии программного обеспечения Cisco IOS. Если какие-либо специальные исправления используются, гарантируют, что вы ясно даете понять этот факт. Сожмите Cisco CallManager и файлы трассировки SDL вместе. Передайте и скопируйте этот файл архива zip, в то время как вы ждете контакта. Сожмите журналы perfmon вместе. Передайте и скопируйте этот файл архива zip, в то время как вы ждете контакта. Сожмите записи журнала событий вместе. Передайте и скопируйте этот файл архива zip, в то время как вы ждете контакта.
6. После того, как вы собрали все трассировки и журналы, сжимаете файлы и передаете файл архива zip к [attach@cisco.com](mailto:attach@cisco.com). Предоставьте номер заявки в почтовой строке темы.

## [Дополнительные сведения](#)

- [Аварийный отказ службы Cisco CallManager](#)
- [Устранение причин сбоев Cisco CallManager](#)
- [Поддержка голосовых технологий](#)
- [Поддержка продуктов Голосовой и Унифицированной связи](#)
- [Устранение неполадок в системах IP-телефонии Cisco](#)
- [Cisco Systems – техническая поддержка и документация](#)