

Como identificar um reinício do CallManager da Cisco como um impacto ou prestar serviços de manutenção à parada programada

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[Diferença entre impactos do CallManager da Cisco e paradas programadas](#)

[Travamentos](#)

[Paradas programadas](#)

[Como relatar impactos do CallManager da Cisco ao Suporte técnico de Cisco](#)

[Informações Relacionadas](#)

[Introdução](#)

Este documento descreve a diferença entre um impacto do CallManager da Cisco e uma parada programada do serviço. O documento igualmente fornece o procedimento para relatar um impacto do CallManager da Cisco e para permitir o [Suporte técnico de Cisco](#) de pesquisar defeitos a edição.

[Pré-requisitos](#)

[Requisitos](#)

Não existem requisitos específicos para este documento.

[Componentes Utilizados](#)

As informações neste documento são baseadas nestas versões de software:

- CallManager da Cisco 3.x e 4.0

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

[Convenções](#)

Consulte as [Convenções de Dicas Técnicas da Cisco](#) para obter mais informações sobre convenções de documentos.

Diferença entre impactos do CallManager da Cisco e paradas programadas

Travamentos

Um erro no código do CallManager da Cisco causa impactos do CallManager. Há três tipos principais de impactos:

- Violações de acesso
- Partilha por zero
- Exceções desconhecidas

Os impactos gerenciam o Dr. entradas Watson, que são adicionados à extremidade do arquivo existente do Dr. Watson. Os impactos igualmente gerenciam arquivos do user.dmp. O lugar destes arquivos é usuários de C:\Documents and Settings\All \ documentos \ DrWatson.

O nome do arquivo do Dr. Watson, que é um arquivo de texto, é drwtsn32.log.

Escolha o **drwtsn32** do indicador da corrida configurar os ajustes.

Como ler um Dr. Watson Arquivo

Termine estas etapas para ler o arquivo do Dr. Watson:

1. Procure pela palavra “quando”, no lowercase, e encontre a data e hora em que o problema ocorreu. O arquivo do Dr. Watson grava todos os impactos do aplicativo. Alguns registros de impacto não podem ser impactos do CallManager da Cisco. Os exemplos dos registros de impacto que não são impactos do CallManager da Cisco incluem RisDC.exe e aupair.exe.
2. Depois que você encontra a data e hora do problema, encontre o número do identificador do processo (PID), e procure a lista de tarefas para determinar que aplicativo causou um crash. A lista de tarefas aparece no exemplo nesta etapa. Neste exemplo, o aplicativo que causou um crash tem um PID de 752, e o nome do aplicativo é 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
regsvc.exe   592 MSTask.exe  924 Explorer.exe  992 cmd.exe    972 msixexec.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. Se o impacto é um impacto do CallManager da Cisco, note o número da exceção para determinar o tipo de travamento. **Nota:** Distribua ao equipe de desenvolvimento apropriado um impacto do aplicativo que não seja um impacto do CallManager da Cisco, caso

necessário.Application exception occurred:

App: (pid=752)

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

Exception number: c0000005 (access violation) Neste exemplo, o número da exceção é c0000005, que é uma violação de acesso. Esta violação de acesso significa que o aplicativo tentou à memória de acesso fora do limite de memória do aplicativo que o grupo do sistema operacional. Um outro tipo de travamento possível é partilha por zero. Porque este exemplo mostra, o número da exceção para a partilha por zero é c0000094:Application exception occurred:

App: (pid=1564)

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

Exception number: c0000094 (divide by zero) O tipo de travamento pode igualmente ser exceção desconhecida. Porque este exemplo mostra, o número da exceção para exceção desconhecida é e06d7363:Application exception occurred:

App: (pid=2784)

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

Exception number: e06d7363 Quando você determina se um impacto é uma violação de acesso, partilha por zero, ou exceção desconhecida, você pode combinar um impacto com um Bug da Cisco existente. Se você não encontra nenhum fósforo, a engenharia de desenvolvimento tem um bom começo para determinar o que ocorreu.

4. Procure sob quando seção do arquivo pela palavra FALHA para começar a definir a “assinatura” do impacto. **Nota:** A FALHA aparece em maiúsculo. Esta seção da FALHA do arquivo contém seis partes de informação importante, que são: O número da linha que experimentou o problema Os índices dos registros para esta linha na altura do impacto A função que executou na altura do impacto A instrução de código de montagem que conduziu ao impacto A pilha para trás segue que mostra os endereços das funções que executada, em ordem, imediatamente antes do impacto A descarga crua da pilha, que fornece mais informação sobre o que estava na pilha de tempo de execução na altura do impacto Este código fornece um exemplo de um impacto do CallManager da Cisco que seja um impacto da violação de acesso. O texto em negrito destaca os seis elementos críticos, assim como a palavra FALHA, que marca esta seção do arquivo:

```
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
e8646cfbff    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
(77fcae4)  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 µm 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=;.
gZ....6 05cef47c 98 f6 e6 36 00 00 00 00 - 00 00 00 00 00 00 00 00 ...6..... Estes

```

seis bit da informação constituem parte de, mas não tudo de, a assinatura do impacto. O resto da informação que define a assinatura é:O tipo de travamento (violação de acesso, partilha por zero, ou exceção desconhecida)As últimas indicações de traço do Signal Distribution Layer (SDL) que executaram antes do impactoNota: O último arquivo SDL que teve o uso antes do impacto, assim como o arquivo do Dr. Watson, diplomatas a alguns causam um crash o erro.Os diplomatas desta informação de assinatura (o último arquivo SDL, o último arquivo do CallManager da Cisco, e o arquivo do Dr. Watson) ao Distributed Defect Tracking System (DDTS) gravam quando você cria um impacto novo DDTs.Se você combina o impacto novo com um DDTs que já exista e tenha a mesma causa de raiz, esta informação é a mesma:O tipo de exceçãoO nome da função que executou na altura do impactoOs nomes das funções na pilha para trás seguemNota: Estes nomes não aparecem sempre em um arquivo do Dr. Watson.A instrução de código de montagem que aparece ao lado do marcador da FALHAAs últimas linhas de rastreamento SDL, que devem ser muito similaresOs índices dos registros, dos endereços de memória, e da outra informação podem diferir da informação em um outro DDTs que exista, mesmo se o impacto tem a mesma causa de raiz. Os endereços variam se você executa uma versão diferente do CallManager da Cisco. Se você executa a mesma versão do CallManager da Cisco, os endereços na seção do código do conjunto e na seção do rastreamento de pilha são os mesmos.

5. Recolha estes arquivos para debugar o impacto:drwtsn32.loguser.dmpOs últimos arquivos SDL e de Trace do CallManager da Cisco, de aproximadamente 5 minutos antes do impacto

e de 5 minutos após o reinício. arquivos do proglog **Nota:** Recolha estes arquivos somente nas versões do CallManager da Cisco 3.2 e mais atrasado. Log de eventos, sistema e aplicativo, se disponível. Logs do monitoramento de desempenho (perfmon), se disponível.

Erro de DBLException

Você vê esta Mensagem de Erro no log do aplicativo do editor do CallManager da Cisco e do subscritor:

```
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..
```

Ou:

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

Details:

```
Error No. -2147219962 (0x80040606):
```

```
CDBLException Dump: [COM Error] COM Error Description = []
```

Este tipo de erro ocorre quando um telefone IP é rejeitado do registro ou devido a uma assinatura quebrada entre os bases de dados da publisher e subscriber. Esta edição pode ser resolvida usando a ferramenta do DBLHelper. Para obter mais informações sobre do DBLHelper, refira a [utilização do DBLHelper para restabelecer uma assinatura quebrada do SQL do Cluster do CallManager da Cisco](#).

Este erro pode igualmente ocorrer devido ao impacto do serviço do monitor da camada do base de dados de Cisco. Execute estas etapas para resolver a edição:

1. Vá ao **iniciar > programas > ferramentas administrativas > aos serviços de componente**.
2. Expanda **serviços de componente > computadores > aplicativos do Meu Computador > COM+**.
3. Comece o serviço **MSDTC** (coordenador distribuído da transação), se mostra parado.

[Paradas programadas](#)

O outro tipo de reinício do CallManager da Cisco é uma parada programada. Uma parada programada é quando o CallManager da Cisco é incapaz de se operar eficazmente e se fecha para baixo. As paradas programadas caem em duas categorias:

- [Timeouts de inicialização](#)
- [Temporizador SDL e encerramentos de encadeamento do roteador SDL](#)

Se o CallManager da Cisco se fecha para baixo, você encontra um código de motivo da parada programada nas últimas linhas de rastreamento do rastreamento do CallManager. Aqui está um exemplo:

```
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>

Neste exemplo, o código de motivo é 4. Esta lista fornece códigos de motivo da parada programada do código do CallManager da Cisco:

```
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
    };
};
```

A razão 1 e a razão 2 são casos raros das paradas internas, quando as outras razões forem mais comuns. A razão 3 indica que a linha do roteador SDL parou a resposta. A razão 4 indica que a linha do temporizador SDL parou a resposta. Razões 5 – 13 relacionam-se ao fogo do temporizador da iniciação.

[Timeouts de inicialização](#)

Quando o serviço do CallManager da Cisco começa primeiramente, a linha do monitor do processo do CallManager (CMPProcMon) começa. Então, os começos da linha do MmmanInit, que desova todos os processos restantes. Em seguida, os começos da linha do roteador SDL. Esta linha segura os sinais que enviam de um processo a outro. Todos os três das linhas começam ao mesmo tempo. A linha do CMPProcMon e a linha do roteador SDL são ascendentes e são executado quando a linha do MmmanInit começar acima outros processos.

CMPProcmon e o SDL precisam de ser em serviço quando o MmmanInit começar acima vários processos. A linha do MmmanInit começa estes processos, nesta ordem:

1. Base de dados (ProcessDb)**Nota:** ProcessDb é uma relação do CallManager da Cisco ao código da camada do base de dados (DBL).Ao mesmo tempo, o código do MmmanInit igualmente começa um número outro de CallManager da Cisco interno, processos independentes. Estes processos incluem H225Handler, MGCPBhHandler, e jefe.
2. Regiões
3. AARNeighborhood
4. Lugar
5. Plano de rota
6. Análise de dígitos
7. Controle de chamadas
8. Serviços suplementaresAs características incluem o parque de chamadas, dianteiro, a conferência, e a transferência.
9. Dispositivo
10. Diretório

11. Gerente do Calling Search Space (CSSManager)

12. Gerente do Time Of Day (TODManager)

A realização destas tarefas ocorre em uma série. Cada uma das doze tarefas tem um temporizador que associe com a tarefa. Este temporizador começa quando a tarefa começa. Se os fogos do temporizador antes que a tarefa terminar, CallManager da Cisco param e imprimem um traço SDL que leia:

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

Esta lista mostra cada um dos temporizadores, assim como o sinal SDL que começa o temporizador e o sinal SDL que para o temporizador. Os sinais de "InitDone" aparecem no traço SDL se você ajustou níveis de rastreamento apropriadamente. (Você ajustou SdlTraceTypeFlags em 0x8000CB15.)

Estes temporizadores padrão são baseados na versão do CallManager da Cisco 4.1(2). Se você executa a versão diferente, pôde ser levemente diferente.

1. O tempo da inicialização de base de dados (padrões a 900 segundos) - o sinal de início por este tempo é o sinal do "começo" enviado ao processo do MmmanInit. Você vê este no traço SDL.
2. Tempo de inicialização das regiões (padrões a 120 segundos).
3. Tempo de inicialização de AARNeighborhoods (padrões a 90 segundos).
4. Tempo de inicialização dos lugar (padrões a 90 segundos).
5. Tempo de inicialização do plano de rota (padrões a 600 segundos).
6. Tempo de inicialização da análise de dígitos (padrões a 900 segundos).
7. Tempo de inicialização do Controle de chamadas (padrões a 90 segundos).
8. O tempo de inicialização dos serviços suplementares (padrões a 900 segundos) - o sinal de início é CclnitDone e o sinal da extremidade é SsInitDone.
9. Tempo de inicialização do dispositivo (padrões a 360 segundos).
10. Tempo de inicialização do diretório (padrões a 90 segundos)
11. Tempo de inicialização de CSSManager (padrões a 900 segundos).
12. Tempo de inicialização de TODManager – (padrões a 900 segundos).

Afinal as tarefas estão completas, CallManager da Cisco abrem os links SDL aos serviços do CallManager que são executado em outros Nós na rede. O CallManager da Cisco igualmente abre os links SDL aos serviços do gerente da integração de telefonia e computador (CTI) que são executado no mesmos nó ou Nós diferentes na rede.

Então, o MmmanInit envia um sinal de CMInitComplete de volta à linha do CMProcMon. Quando o CMProcMon começa primeiramente, começa um temporizador duro-codificado 60-minute para a iniciação do CallManager da Cisco. O temporizador tem o nome CMInitComplete_WaitTime. (Este temporizador não é um parâmetro de serviço; o temporizador não é configurável.) Se a linha do CMProcMon não recebe o sinal de CMInitComplete dentro de 60 minutos, o CallManager da Cisco para e emite uma indicação de traço que leia:

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

Se qualquer das doze tarefas de inicialização falha, ou se o tempo total para estas tarefas excede 60 minutos, o CallManager da Cisco para.

Nota: O temporizador de CMInitComplete_WaitTime era uma vez duro codificado aos minutos 10. Este código duro mudou a 60 minutos como parte da identificação de bug Cisco [CSCdx31622](#) ([clientes registrados somente](#)). A mudança incorporou o 3.1(3) projetando o trem (ES) especial, com ES 38 como o começo. A mudança está igualmente 3.2(2) no trem ES, com ES 11 como o

começo, e no CallManager da Cisco 3.3.

Se você experimenta problemas com um fogo do temporizador da iniciação, você pôde somente precisar de aumentar a configuração de temporizador para resolver a partida. Se esta mudança não resolve o problema, o problema pode ser um tempo de resposta lento do base de dados que faça com que a operação cronometre para fora. Collect detalhou traços DBL, assim como traços SDL e de CallManager da Cisco, caso necessário.

Recolha estes arquivos para debugar um problema de inicialização:

- Trace do CallManager da Cisco detalhado
- Traço SDL **Nota:** Ajuste o SdlTraceTypeFlags a 0x8000CB15.
- Traço detalhado DBL

Temporizador SDL e encerramentos de encadeamento do roteador SDL

A linha do roteador SDL é a linha a mais importante da execução dentro do aplicativo do CallManager da Cisco. Controla a emissão de sinais de Processamento de chamadas. A linha do CProcMon verifica a saúde da linha do roteador SDL uma vez cada dois segundos. Os traços do CallManager da Cisco mostram este exame médico completo, como você pode ver nestas indicações:

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

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

Se a linha do CProcMon incorpora e retira a verificação do roteador, a linha do roteador SDL respondeu ao exame médico completo e é muito bem.

Contudo, se a linha do roteador SDL não responde, você vê quando laço em Trace do CallManager da Cisco, como o este mostras:

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

Nesta situação de emergência, a linha do roteador SDL recebe verificações uma vez cada segundo por um período de 20 segundos. Se a linha responde a qualquer hora durante o período 20-second, a operação normal recomeça, e a saúde da linha do roteador SDL recebe mais uma vez a verificação cada 2 segundos. Se, contudo, a linha do roteador SDL não responde às verificações sobre os 20 segundos, o aplicativo do CallManager da Cisco fechou. Esta indicação aparece no traço 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,
```

Observe o código de motivo 3 no texto desta indicação de traço. O código significa que a linha do roteador SDL parou a resposta, assim que o CallManager da Cisco fechou.

A causa mais provável de uma parada de thread de roteador SDL é uma falta dos recursos de

sistema. Um outro aplicativo usou a maioria ou todo o CPU durante um longo período do tempo, pelo menos 20 segundos. Esta atividade é porque os monitoramentos de desempenho são vitais debugar este tipo de parada programada.

O outro tipo de parada programada a explorar é o fechamento de encadeamento de temporizador de SDL. Um fechamento de encadeamento de temporizador de SDL ocorre quando o diferencial entre o pulso de disparo interno do CallManager da Cisco e o pulso de disparo externo do sistema operacional excede 16 segundos. Quando o fechamento de encadeamento de temporizador de SDL acontece, este traço aparece em Trace do CallManager da Cisco:

```
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>
```

O CallManager da Cisco verifica geralmente as linhas do temporizador uma vez cada segundo. O CallManager da Cisco adicionam 1 segundo ao tempo atual do sistema operacional e as lojas que avaliam afastado como o “tempo esperado.” Então, o CallManager da Cisco dorme para 1 segundo. Depois que o CallManager da Cisco acorda, verifica o tempo novo do sistema operacional e subtrai o tempo esperado. Se a diferença entre estas duas épocas é mais de 1 segundo, esta indicação de advertência aparece em Trace do CallManager da Cisco:

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

A Latência real nesta indicação mostra que a linha interna do temporizador SDL do CallManager da Cisco executa lento. Aqui, a diferença entre o tempo esperado do CallManager da Cisco e o tempo real do sistema operacional é 1.63 segundos.

Se esta diferença excede 16 segundos, o CallManager da Cisco fecha e fornece o código de motivo da parada programada de 4.

A causa mais provável de um fechamento de encadeamento de temporizador de SDL é uma falta do processador central - hora para o CallManager da Cisco. Um outro aplicativo, tal como VirusScan ou backup de Sti, usou a maioria dos recursos do CPU no mínimo 16 segundos. Os logs do perfmon são vitais determinar a causa de raiz deste tipo de parada programada.

Se o backup de aplicativos do Cisco IP Telephony é executado durante um longo período do tempo na utilização elevada da CPU, um travamento de sistema pode ocorrer. Para obter informações sobre de como evitar este travamento de sistema, refira:

- [Verifique ajustes no utilitário de backup para evitar a](#) seção da [alta utilização da CPU do travamento de serviço do CallManager da Cisco do](#) documento

Recolha estes arquivos no caso de uma linha ou de um fechamento de encadeamento de temporizador de SDL do roteador SDL:

- Trace do CallManager da Cisco detalhado
- Traço SDL **Nota:** Ajuste o SdlTraceTypeFlags a 0x8000CB15.
- O perfmon segue, se disponível, que mostra o percentual de utilização de CPU de todos os processos que são executado na caixa **Nota:** Você pode capturar estes traços remotamente para reduzir o impacto no desempenho no CPU do servidor do CallManager da Cisco.

[Como relatar impactos do CallManager da Cisco ao Suporte](#)

[técnico de Cisco](#)

O diagnóstico da causa real de um impacto do CallManager da Cisco é difícil. A fim de determinar a causa e expedir uma solução, o [Suporte técnico de Cisco](#) exige-o recolher traços e logs do Dr. Watson e transferir arquivos pela rede a informação às notas de [caso de suporte técnico de Cisco](#). Você envia as notas de caso a attach@cisco.com e fornece o número de caso na linha de assunto do email. O procedimento é:

1. Recolha arquivos de Trace do CallManager da Cisco de 30 minutos antes e de 15 minutos após o impacto. O lugar dos traços é C:\Program Files\Cisco\Trace\CCM.
2. Recolha arquivos de rastreamento SDL de 30 minutos antes e de 15 minutos após o impacto. O lugar dos traços é C:\Program Files\Cisco\Trace\SDL\CCM.
3. Recolha o user.dmp e os arquivos de drwtsn32.log. O lugar dos arquivos é usuários de C:\Documents and Settings\All \ documentos \ DrWatson.
4. Selecione o **iniciar > programas > ferramentas administrativas > visualizador de evento** para recolher arquivos do sistema e de log de eventos do aplicativo do visualizador de eventos. Você pode saltar esta etapa, se os dados de log de eventos não são necessários. Mas, despeje o sistema e os eventos de aplicativo e filtre para fora somente os eventos de aproximadamente 30 minutos antes do impacto. Investigue estes eventos antes que você os envie ao [Suporte técnico de Cisco](#). Você pode encontrar um evento que justifique mais atenção. **Cuidado:** Seja cuidadoso se você usa o visualizador de eventos, um utilitário Microsoft incorporado, despejar estes eventos a um arquivo de texto. Em um sistema que tenha a utilização elevada da CPU, este uso do visualizador de eventos pode facilmente morrer de fome todos processos restantes do CPU. Estes processos incluem o processo de keepalive do CallManager da Cisco que mantém registros do telefone. Você pode usar um utilitário de shareware com o nome elogdmp.exe para despejar todas as entradas nos logs individuais a um arquivo de texto. As implicações de CPU são insignificantes quando você usa a ferramenta elogdmp.exe. Emita este comando de um prompt do DOS:

```
elogdmp COMPUTER_NAME Application > AppEvents.txt elogdmp COMPUTER_NAME System > SysEvents.txt
```
5. Comprima todos os arquivos como arquivos zip na ordem que esta etapa mostre antes que você email e copie os arquivos. Use a versão do WinZip 8 para comprimir os arquivos. (Cisco tem uma licença de site para esta utilidade.) Geralmente, cópia de arquivos a uma máquina local para a avaliação mais rápida. Os arquivos que você comprime o uso menos espaço, e você podem mover estes arquivos muito mais rapidamente do que formatos do arquivo crus. Comprima o user.dmp e os arquivos de drwtsn32.log junto. Imediatamente envie e copie este arquivo zip. Forneça uma definição descritiva de sintoma e inclua a versão do CallManager da Cisco exata, as cargas do dispositivo apropriadas, e as versões do Cisco IOS ® Software. Se alguma correção de programa especial está no uso, assegure-se de que você faça este fato claro. Comprima o CallManager da Cisco e arquivos de rastreamento SDL junto. Envie e copie este arquivo zip quando você esperar o contato. Comprima os logs do perfmon junto. Envie e copie este arquivo zip quando você esperar o contato. Comprima as entradas de Log de evento junto. Envie e copie este arquivo zip quando você esperar o contato.
6. Depois que você recolheu todos os traços e logs, comprima os arquivos e envie o arquivo zip a attach@cisco.com. Forneça o número de caso na linha de assunto do email.

[Informações Relacionadas](#)

- [Falha no serviço Cisco CallManager](#)
- [Pesquisando defeitos impactos do CallManager da Cisco](#)
- [Suporte à Tecnologia de Voz](#)
- [Suporte ao Produto de Voz e Comunicações Unificadas](#)
- [Troubleshooting da Telefonia IP Cisco](#)
- [Suporte Técnico e Documentação - Cisco Systems](#)