

# Remote Process Monitor Console ( Procmon ) の使用

## 内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[用途](#)

[procmon のトレース](#)

[ICM 4.0 よりも前の Procmon](#)

[ICM 4.0 以降での Procmon](#)

[関連情報](#)

## 概要

Process Monitor Console ( procmon ) は、メイン インターフェイスのコンソール プロセスです。サイトの自動着信呼分配 ( ACD ) の情報を間接的に照会するときに procmon を使用できます。ペリフェラル ゲートウェイ ( PG ) に常駐する Cisco Intelligent Contact Management ( ICM ) プロセスにより、次の操作を実行できます。

- エージェント、スキル、サービス、およびコール データの一覧表示
- ICM プロセス上でのトレースの増加
- ACD 固有の情報に対するクエリー

## 前提条件

### 要件

次の項目に関する専門知識があることが推奨されます。

- ICM PG のトラブルシューティングとサポート

### 使用するコンポーネント

このドキュメントの情報は、ICM バージョン 4.6.2 以降に基づくものです。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期 ( デフォルト ) 設定の状態から起動しています。対象のネットワークが実稼働中である場合には、どのようなコマンドについても、その潜在的

な影響について確実に理解しておく必要があります。

## 表記法

ドキュメント表記の詳細は、『[シスコテクニカルティップスの表記法](#)』を参照してください。

## 用途

Telnet セッションまたは DOS コマンド プロンプトから **procmon** を実行できます。**procmon** プロセスでは、ローカルとリモートの両方のコマンドがサポートされます。ローカル コマンドは **procmon** 内部で定義されますが、リモート コマンドは監視対象のプロセス内にユーザがプログラムする必要があります。このセクションでは、次のようなプロセスで使用するための基本的な **procmon** コマンドとプロセス固有のコマンドの一覧を提供しています。

- Peripheral Interface Manager ( PIM )
- Computer Telephony Integration ( CTI; コンピュータ/テレフォニー インテグレーション ) Server ( CTISVR; コンピュータ/テレフォニー インテグレーション サーバ )
- Open Peripheral Controller ( OPC )

以下が一例です。

```
Syntax: c:\>procmon /?  
Version: Release 4.6.2, Build 08799  
Usage: PROCMon CustomerName NodeName ProcessName [SystemName] [/f InputFile]  
      [/wait] [/stop] [/help] [/?]
```

注：この出力の最後の行は、スペースの制限により2行にわたって表示されます。

コマンドの基本的な一覧を表示するには、**help** を発行します。次のような一覧が表示されます。

コマンド	定義
エコー	コマンドラインのエコーの制御
emsmon	リモート EMS <sup>1</sup> モニタ プロセスの制御 ( 開始、停止、一時停止、再開 )
error_stop	エラー フラグでの停止設定の制御
help、?	ヘルプの表示
monitor_help、mhelp	モニタ サーバのヘルプの表示
monitor_sleep、msleep	指定した秒またはミリ秒のスリープ
quit、q	プログラムの終了
read_file、read	別の入力ファイルへのコマンド入力のリダイレクト

<sup>1</sup> EMS = Event Management System

次の表では、トラブルシューティングの際に使用できるコマンドの一覧を提供しています。

コマンド	定義
------	----

pim_list_agents、la	PIM によって現在設定されているエージェントの一覧表示
pim_list_services、ls	PIM によって現在設定されているサービスの一覧表示
pim_list_skill_groups、lsg	PIM によって現在設定されているスキルグループの一覧表示
acd_debug、debug	debug トレースのオン/オフ
pim_list_trace、ltrace	現在の PIM トレースビット設定の一覧表示
pim_trace、trace	PIM トレースビットのセットまたはリセット
pim_dump_periph、acdperiph	ペリフェラル オブジェクトのコンテンツのダンプ
QUIT	procmon の終了

各コマンドには個別の構文が存在します。構文を確認するには、末尾に /? を続けたコマンドを発行します。

注：各ペリフェラルタイプには、異なるコマンドセットが含まれています。各ペリフェラルに関連付けられているコマンドの一覧の場合は、mhelp を発行します。

出力例を次に示します。

```
>>>>la
SkillTarget ID      Periph#    C          Ext#          Inst#          ActGroups
5000                6000      Y Yes      3000(3000)   <1/ 1> [BO] [265436],<2/ 2> [BO][866278]
5001                6001      Y Yes      3001(3001)   <1/ 1> [AV] [59704], <2/ 2> [AV] [59704]
5002                6002      Y No       -1(-1)
5003                6003      Y No       -1(-1)
5028                6030      Y No       -1(-1)
```

```
>>>>ls
SkillTarget ID      Periph#    C          SerMem        Pri          SerTH         SLType        PSLType        Ext#
5017                6500      Y          1             2            30            1             4
5018                6501      Y          2             1            30            1             4
5019                6502      Y          3             1            30            1             4
```

この出力では、SLType は、ペリフェラルと関連付けられている各サービスの ServiceLevelType フィールドのデフォルト値を示しています。この値は、ICM によるサービスレベルの算出方法を示しています。個々のサービスのデフォルトを上書きできます。

PSLType PeripheralServiceLevelType 個々のサービスのデフォルトを上書きできます。

```
>>>>lsg
Periph#    Pri    C    SkillTarget ID    Ext#
1          0     Y    5007              6900
1          1     Y    5008              6900
1          2     Y    5009              6900
2          0     Y    5010              6901
2          1     Y    5011              6901
2          2     Y    5012              6901
3          0     Y    5013              6902
3          1     Y    5014              6902
3          2     Y    5015              6902
```

>>>>debug /?

```
Usage: acd_debug [/noagent] [/agent] [/agent+] [/agent++] [/nobri] [/bri] [/bri+]
        [/nocall] [/call] [/call+] [/nocms] [/cms] [/cms+] [/csc]
        [/csc+] [/nocsc] [/noconfig] [/config] [/nocv] [/cv] [/noerror]
        [/error] [/nohb] [/hb] [/noopc] [/opc] [/nopost] [/post] [/nosim]
        [/sim] [/notg] [/tg] [/notimer] [/timer] [/notp] [/tp] [/tp+]
        [/trace] [/novq] [/vq] [/warning] [/nowarning] [/all] [/noall]
        [/set UserSetBit] [/help] [/?]
```

>>>>debug /call+ /post /agent

Trace: AGENT CALL+ POST

UserTraceLevel=0xE848200003FFFFFF800E000000000000000000000000000000040

Time stamp: 09/13/02

**注：デバッグ機能は、プロセスがアクティブの間のみアクティブです。プロセスが終了すると、debug ユーティリティは機能しなくなります。トレースを永続的にするには、UserTraceLevel 行に表示された 16 進数の数値をレジストリ内の EMS トレースに追加します。**

>>>>acdperiph

BuildNum: 08799 (Rel 4.6.2) Time: 06/11/02 16:27:40

SwitchTime=08/26/02 13:56:22, DefRoute=CTIVarMap-NNNNNNNNNN (y=PIM access)CTIString=

CVBridge=[G3MsgRecvCnt=169239 (0x29517) Min/AllBrisUp=1/1 NumMonitored=1

PhysBris=0x1 RtBris=0x0 BadBris=0x0]

Bri[0] State=ACTIVE GoIdle=0

[NtwrkCngstn[Forced=F Switch=F]

Window=10000 MsgDlyTime=500

BriCfgParams(Exp.) = [\*CvHost[0]=taclablg3 CvHost[1]= ]

Msgs [Sent=157095 (0x265a7) Recv=169239 (0x29517) ] [SA0id=314182 LastSA0IdRecvd=31482

Msgs [SendQ=0x0 SentQ=0x0 RecvQ=0x0 ]

Msgs [PriSendQ=0x0 RecvQ=0x0 ]

[ActiveAssoc[Avail=2033 Locked=11] OutstandingSent=0x0 Reg{MaxAllowed=4 ChkMtrs=1

ChkMsgRates=1

[Meters/Sec (Enabled: Min 0.00 Avg 0.17 Max 2.45 (Tot 28840.16 Samples 229013

SumAvg 0.13)]

[NotEnabled]

Timers=[3PMC=4 ACDSplt=61 AgntCls=30 AgntSt=240 BriHB=60

CfgRtry=900 StlBriMsg=10 SwtchTm=30 TG=60 StatMntr=28800 StatMntrInit=120]

SwitchTime=08/26/02 13:56:22

NumActiveCalls=0 NumAgentsSeen=2

ProcessName=pim1 ShutdownType=1 Duplex=1 Side=0

GeoTelBaseDir=C:\icr\lab1\PG1B RegistryBase=ICR\lab1\PG1B DMPSYSTEMID=1

MDSConnections=1 MDSPIMHandle=33 MDSOPCHandle=1 PIMHeartBeatTime=-1

CTIRestarts=0

RoutingClientState=SHUTDOWN

State=ACTIVE StateInitTime=08/27 10:06:55 (16.9 day)

Time stamp: 09/13/02 10:32:36

>>>>

**注：acdperiphの詳細は、『Procmonを使用したAvaya Definity G3のトラブルシューティング』を参照してください。**

## procmon のトレース

### ICM 4.0 よりも前の Procmon

- PIM、MIS、および CTISVR のプロセスでトレースを起動するには、procmon を使用できます。構文：**procmon custid nodeid processname**。使用例は **procmon bt pg1a pim1** です。
- >>プロンプロードでmhelpと入力して、Procmonのヘルプにアクセスします。たとえば、>>

mhhelpと入力します。

- トレースの追加：トレースを追加するには、**sxtrace**、**scrtrace**、および**satrace**コマンドを使用します。with/allを使用しません。**sxtrace**の使用例は、>>**sxtrace /all**です。**svxtrace**、**svcrtrace**、および**svatrace**のコマンドを発行して、トレースを保存する必要もあります。スペクトラムの問題にトラブルシューティングを行う場合は、3つのトレースレベルのすべてを追加し、保存することが推奨されます。
- トレースの削除：トレースを削除するには、**cxtrace**、**ccrtrace**、および**catrace**コマンドを使用します。with/allを使用しません。**cxtrace**の使用例は、>>**cxtrace /all**です。トラブルシューティングの完了時にはトレースを削除することが常に推奨されます。
- Ems logs – すべてのトレースで、regedt32のEmsLogFileMaxおよびEmsAllLogFilesMaxの設定を増やす必要があります。これらの値へのパスは次のとおりです。

```
HkeyLocalMachine\Software\Geotel\ICM\custid\PGxx\EMS\CurrentVersion\  
Library\Processes\processid
```

注：この値は、スペースの制限により2行にわたって表示されます。

## ICM 4.0 以降での Procmon

- PIM、MIS、および CTISVR のプロセスでトレースを起動するには、**procmon** を使用できます。構文：**procmon custid nodeid processname**。使用例は **procmon bt pg1a pim1** です。
- Procmonのヘルプにアクセスするには、>>プロンプトでmhhelpと入力します。たとえば、>>mhhelpと入力します。
- トレース：**ltrace**コマンドは、使用可能なすべてのトレースオプションを表示します。>>プロンプトで**trace xact\***と入力し、トランザクション・リンクトレース(**sxtrace**)を適用します。**trace spectrum\*** コマンドを使用して Agent トレースを適用します。

## 関連情報

- [トレース機能を強化する場合](#)
- [OPCTest のコマンドライン ユーティリティの使用](#)
- [トレース機能を強化する場合](#)
- [IPCC のトラブルシューティングガイド](#)
- [Dumplog ユーティリティの使用方法](#)
- [Procmon を使用した Avaya Definity G3 のトラブルシューティング](#)
- [Cisco ICM ソフトウェア リリース 4.6.2 のリリースノート](#)
- [テクニカル サポートとドキュメント – Cisco Systems](#)