

# Remote Process Monitor Console ( Procmon ) の使用

## 目次

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[使用方法](#)

[Procmon トレース](#)

[先の Procmon ICM 4.0 より](#)

[ICM 4.0 および それ以降の Procmon](#)

[関連情報](#)

## [はじめに](#)

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をリストして下さい
- ICM プロセスのトレースを高めて下さい
- ACD 固有の情報のためのクエリ

## [前提条件](#)

### [要件](#)

次の項目に関する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およびサポート

### [使用するコンポーネント](#)

この資料に記載されている情報は ICM バージョン 4.6.2 および それ 以降に基づいています。

本書の情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。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。稼働中のネットワークで作業を行う場合、コマンドの影響について十分に理解したうえで作業してくだ

さい。

## 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。

## 使用方法

Telnetセッションか DOS コマンド プロンプトから **procmon** を実行できます。procmon プロセスはローカルおよび remote コマンドを両方サポートします。ローカル コマンドは procmon の内で監視されたプロセスに remote コマンドをプログラムする必要があるが、定義されます。このセクションはプロセスを使用に必要な procmon コマンドおよびプロセス固有のコマンドのリストに与えます ( 以下を参照 ) :

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

次に例を示します。

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

注: この出力最終的な行はスペース制限による 2 つの行に表示します。

コマンドの基本的なリストを表示するために、ヘルプを発行して下さい。このようなリストは下記のものを表示します:

コマンド	定義
echo	コマンド・ラインの制御エコー
emsmon	制御リモート EMS <sup>1</sup> モニタ プロセス ( 開始する、停止、一時停止、レジューム )
error_stop	Error フラグの停止の設定を制御します
ヘルプ、か 。	ヘルプを表示します
monitor_he lp、mhelp	監視サーバー ヘルプを表示します
monitor_sl eep、 msleep	指定時間 ( 秒/ミリ秒 ) のスリープ
やめられる 、q	プログラムを終了します
read_file、 読んで下さ い	別の入力ファイルにコマンド 入力を指示します

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

この表は解決するとき使用できるコマンドのリストを提供したものです:

コマンド	定義
pim_list_agents、la	PIM によって現在設定されるエージェントをリストします
pim_list_services、ls	PIM によって現在設定されるサービスをリストします
pim_list_skill_groups、lsg	PIM によって現在設定されるスキルグループをリストします
acd_debug、デバッグ	オン/オフ debug trace を回します
pim_list_trace、ltrace	現在の PIM トレース ビット設定をリストします
pim_trace、トレース	セットかリセット PIM トレース ビット
pim_dump_periph、acdperiph	周辺装置オブジェクトの内容をダンプします
quit	procmon を終了します

各コマンドに別途の構文があります。構文を判別するために、コマンドを発行し、それにと/続いて下さいか。

注: 各々の周辺装置型はコマンドが異なるセットが含まれています。各周辺装置と関連付けられるコマンドのリストに関しては、mhelp を発行して下さい。

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 周辺装置と関連付けられる各サービスの ServiceLevelType フィールドのデフォルト値を示します。この値は ICM がサービスレベルをどのように計算するかを示します。個々のサービスのためのデフォルトを無効にすることができます。

PSLType 周辺装置と関連付けられる各サービスの PeripheralServiceLevelType のデフォルト値を示します。個々のサービスのためのデフォルトを無効にすることができます。

```
>>>>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
3	3	Y	5016	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=0xE848200003FFFFFF800E00000000000000000000000000000040

Time stamp: 09/13/02

**注: デバッグ機能はプロセスが間、だけアクティブのままになります。プロセス終了が、デバッグユーティリティもはや機能しない時。トレースパーマネントを作るために、レジストリのEMSトレースに UserTraceLevel 行で見つける hexadecimal 数を追加して下さい。**

>>>>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 に関する詳細については、[Procmon を使用してトラブルシューティング Avaya Definity G3](#) を参照して下さい。**

## [Procmon トレース](#)

## 先の Procmon ICM 4.0 より

- PIM、MIS および CTISVR プロセスでトレースするターンアップに **procmon** を使用できます。構文— **procmon custid nodeid processname**。使用例は **procmon bt pg1a pim1** です。
- の型 **mhelp >> Procmon** のためのヘルプにアクセスするプロンプト。たとえば、**>> mhelp**。
- Tracing の追加— **tracing** を追加するために、**sxtrace**、**scrtrace** および **satrace** コマンドを使用して下さい; **/all** と使用して下さい。 **sxtrace** のための使用例は **>>sxtrace /all** です。また **svxtrace**、**svcrtrace** および **svatrace** コマンドの発行によってトレースを保存して下さい。スペクトルに関する問題を解決するとき付け加える推奨し、3つのトレースレベルをすべて保存しますことを。
- 取除いて下さいトレース—トレースを取除くために、**cxtrace**、**ccrtrace** および **catrace** コマンドを使用して下さい; **/all** と使用して下さい。例 **cxtrace** 使用は **>>cxtrace /all** です。トラブルシューティングが終了次第トレースを取除くことがより適切常にです。
- Ems は記録します—すべてのトレースによって、**regedt32** の **EmsLogFileMax** および **EmsAllLogFilesMax** 設定を高める必要があります。これらの値へのパスは次のとおりです:

```
>>>>acdperiph
BuildNum: 08799 (Rel 4.6.2) Time: 06/11/02 16:27:40
SwitchTime=08/26/02 13:56:22, DefRoute=CTIVarMap-NNNNNNNNNNN (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]=taclab1g3 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 MDSPICHandle=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: スペースの制約上、上記の値は 2 行で表示されています。

## ICM 4.0 および それ以降の Procmon

- PIM、MIS および CTISVR プロセスでトレースするターンアップに **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** のためのヘルプに、型 **mhelp** でアクセスするため **>>** プロンプト; たとえば、**>> mhelp**。
- **トレース— ltrace** コマンドはすべての利用可能なトレース オプションを表示します。で **トレース xact\*** の入力によってランザクションリンクトレース (**sxtrace**) を **>>** プロンプト加えて下さい。 **トレース spectrum\*** コマンドでエージェントをトレースして下さい。

## 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 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 を使用するトラブルシューティング Avaya Definity G3](#)
- [Cisco ICMソフトウェア リリース 4.6.2 に関するリリース ノート](#)
- [テクニカル サポートとドキュメント – Cisco Systems](#)