



シスコのパフォーマンス管理レポートの概要

Cisco Prime Performance Manager は、拡張性が高く、使い勝手の良いパフォーマンス管理システムです。これを使用することにより、サービス プロバイダーは、サービス保証や容量計画などを含め、次世代ネットワークの予防的な管理が可能になります。

このガイドでは、Prime Performance Manager に付属する「購入してすぐに使用可能」な組み込みレポートを、独自のカスタマイズされたサービス レポートを作成することにより拡張する方法について説明します。

この章は、次の項で構成されています。

- 「XML ベースのサービス レポート」 (P.1-1)
- 「レポート記述インターフェイスの主要コンポーネント」 (P.1-2)
- 「機能のしくみ」 (P.1-3)
- 「オンライン レポート ヘルプ」 (P.1-4)
- 「指定可能な項目」 (P.1-10)
- 「レポート管理インターフェイス」 (P.1-12)

XML ベースのサービス レポート

Prime Performance Manager レポートは、XML でコーディングされます。購入してすぐの状態では、Prime Performance Manager にはサンプルとして使用可能な 4400 を超えるレポートが含まれています。サンプル レポートは、自分の作業ディレクトリにコピーし、XML を修正してから、独自のレポートとしてテストおよびデバッグできます。

組み込みレポートでは、次に関するデータ レポートが表示されます。

- アプリケーション トラフィック
- アプリケーション
- デバイス インターフェイス、MPLS トンネル、疑似配線、および SNMP デバイスの可用性
- ESXi、Hyper-V、KVM、UCS クラスタ、およびその他の計算技術
- データ ネットワーク サービス
- IP の Quality of Service (QoS)
- IP プロトコルのパフォーマンス
- IP サービス レベル契約 (SLA) 統計情報
- レイヤ 2 プロトコル

- Mobile IOS Statistics
- Mobile StarOS Statistics
- Mobile StarOS の CDMA KPI
- Mobile StarOS の CDMA 統計情報
- Mobile StarOS の KPI
- Mobile StarOS Statistics
- NetFlow
- NetFlow CGNAT
- ネットワークおよびネットワーク サービス
- Prime Performance Manager システム レポート
- リソース使用率 (CPU やメモリの使用率など)
- Security
- ストレージ (NetApp、EMC、およびファイバ チャネル)
- ATM 統計情報、イーサネット仮想回路 (EVC) 統計情報などの転送統計情報
- ビデオ ブロードキャスト

詳細については、www.cisco.com にある『Cisco Prime Performance Manager 1.4 Data Sheet』を参照してください。次の操作を実行する必要があります。

1. <http://www.cisco.com/go/performance> にアクセスします。
2. [Product Literature] > [Data Sheets] を選択します。

レポート記述インターフェイスの主要コンポーネント

レポートを作成するために必要な XML ファイル、MIB、およびコンフィギュレーション ファイルは、Prime Performance Manager ゲートウェイ サーバに配置されます。

このガイドでは、次のコンポーネントの使用方法について説明します。

- **サポートされる MIB** : Prime Performance Manager は、レポートの作成に使用可能な 600 個を超えるシスコおよび業界標準の MIB をサポートしています。
- **機能ファイル** : 機能ファイルにより、レポートに使用される MIB とポーリングされる MIB 変数を指定できます。機能ファイルには次の 2 つがあります。
 - SystemCapability.xml : 事前にコーディングされているシステム機能ファイル。このファイルは、Prime Performance Manager システムと一緒に提供されるレポートの機能を指定しています。このファイルを変更してはなりません。
 - UserCapability.xml : ユーザが作成したレポートのシステム機能ファイル。レポートの機能を追加または修正する必要がある場合は、このファイルで変更内容を指定します。
- **あらかじめパッケージに組み込まれている XML レポート** : Prime Performance Manager アプリケーションは、独自レポートのひな形として使用可能な 380 個を超える XML レポート ファイルを提供しています。
- **プロパティ ファイル** : 各 XML レポート ファイルには、それぞれ対応するプロパティ ファイルがあります。このプロパティ ファイルは、オンライン レポートと CSV レポートで使用される変数を定義します。

- **レポート マクロ** : XML レポート コードや UserCapability.xml ファイルから呼び出して処理タスクを実行可能な SNMP マクロの集合です。
たとえば、指定されたオブジェクトを IP アドレスに変換するために、IpAddress() マクロが提供されています。
- **BQL ファイル** : BQL ファイルは、Cisco Prime Network クライアント上でのレポートの相互起動を可能にします。
- **オンライン ヘルプ** : 独自レポートのシステム生成ヘルプとレポート ヘルプ ページが含まれます。

レポート ファイルおよび関連ファイルのディレクトリの場所

表 1-1 に、レポートおよび関連ファイルの場所を示します。

表 1-1 Prime Performance Manager のレポートおよびサポート ファイルのディレクトリの場所

ファイル	ロケーション
パフォーマンス レポート (5 分、15 分、毎時、毎日、毎週、毎月) は CSV にエクスポートされます。	/opt/CSCOppm-gw/reports
ゲートウェイ ログ	/opt/CSCOppm-gw/logs
単位のログ	/opt/CSCOppm-unit/logs
MIB	/opt/CSCOppm-gw/etc/mibs
機能ファイル	/opt/CSCOppm-gw/etc/ SystemCapability.xml /opt/CSCOppm-gw/etc/ UserCapability.xml
システムの XML レポート/プロパティ ファイル	/opt/CSCOppm-gw/etc/pollers/system
ユーザの XML レポート/プロパティ ファイル	/opt/CSCOppm-gw/etc/pollers/user
BQL ファイル	/opt/CSCOppm-gw/etc/bql/xl
レポート マクロ	Prime Performance Manager ゲートウェイ上でコンパイル済み。マクロのリファレンス情報については、 第 9 章「レポート マクロ リファレンス」 を参照してください。

機能のしくみ

Prime Performance Manager は、レポートを次のように処理します。

1. Prime Performance Manager は、次に基づいてネットワーク インベントリ内のデバイスをポーリングします。
 - Prime Performance Manager レポート内でポーリング用に選択された MIB。
 - SystemCapability.xml ファイルと UserCapability.xml ファイルで指定されたフィルタリング。

フィルタリング プロセスは、ポーリング対象デバイスに対し、MIB の使用が実際にサポートされているかどうかについて問い合わせます。また、MIB がサポートされるかどうか問い合わせます。それにより、ポーリングは、実際にテーブルデータが存在し、なおかつ他の指定された基準にも一致する MIB オブジェクトに制限されます。その結果、Prime Performance Manager は不要なポーリングを実行しなくなります。

2. システム XML レポートやユーザ定義レポートによってポーリングされる MIB 変数に基づいて、Prime Performance Manager は、ポーリングされたデータを格納する仮想データベース テーブルを作成します。
これにより、ポーリング データの処理が高速化され、レポートを迅速に表示できるようになります。
3. レポート XML 内の指定に基づいて、システムは、ポーリングによって返されたデータを処理しません。事前に定義されているレポート マクロを使用して、データを操作できます。
たとえば、値をパーセンテージに変換できます。
4. システム レポート内やユーザ定義レポート内のマクロ呼び出しに基づいて、Prime Performance Manager は、仮想テーブルを修正します。たとえば、2 つのテーブルが結合されたり、テーブル行に含めるためのデータが選択されたりします。
5. データの入ったレポートは、ユーザが Prime Performance Manager のレポート ツリーからそれらを選択すると表示されます。レポートの表示形式は、レポート XML でカスタマイズできます。
6. サーバに設定されたレポート期間が終了するたびに、システムは、仮想テーブル データを Prime Performance Manager データベースに保存します。

データ ポーリングおよびレポート表示をカスタマイズする機能により、ユーザに対してデータを柔軟かつ効果的な方法でレポートできます。このガイドのチュートリアル の章 (第 2 章「レポートの記述」) では、一般的なレポートである `cpu.xml` レポートのコーディングについて一通り説明した後、サンプル レポートを修正して独自のレポートを作成する方法について示します。

オンライン レポート ヘルプ

Prime Performance Manager は、レポートの作成に役立つ強力なヘルプ システム (独自レポート用の自動生成ヘルプなど) とレポート ヘルプ ページを提供しています。

ここでは、次の内容について説明します。

- 「[独自レポート用の自動生成ヘルプ](#)」 (P.1-4)
- 「[レポートのヘルプ](#)」 (P.1-10)

独自レポート用の自動生成ヘルプ

レポートを記述し、イネーブルにすると、Prime Performance Manager は、そのプロジェクト用のオンライン ヘルプを自動的に作成します。また、それ用のカスタマイズされたヘルプ ファイルを記述し、公開することもできます。Prime Performance Manager は、レポート ヘルプ ファイルを毎晩一回再構築します。レポート ヘルプは、ゲートウェイ CLI から `ppm docreps` コマンドを実行して手動で再生成することもできます。

自動生成ヘルプには、次のものが含まれます。

- **レポート定義ファイルへのリンク** : XML ファイル名をクリックして、XML 定義を表示します。PPM Viewer で定義を表示できます (単純な ASCII テキスト形式)。ブラウザのフレーム ソースビューアを使用することもできます。その場合、XML キーワードとコーディング要素が色分けにより強調表示されます。
- **カスタム ヘルプ** : [Custom Help] リンクをクリックして、レポートのカスタマイズされたヘルプを表示します。
- **レポートで使用される MIB へのリンク**。MIB ファイル名をクリックして、MIB を表示します。

例 1-1 に、`cpu.xml` レポートのオンライン ヘルプを示します。

例 1-1 cpu.xml レポートのシステム生成ヘルプ

```

=====
Definition File:
  cpu.xml   (PPM      Viewer)
  cpu.xml   (Browser Viewer - Use View Page Source Menu For Color Coded View)
=====

Online Help
Custom Help

MIBs Used:
  CISCO-PROCESS-MIB.my

MIBs Used:
  ENTITY-MIB.my

MIBs Used:
  HOST-RESOURCES-MIB.my

MIBs Used:
  UCD-SNMP-MIB.my

-----

Variables Collected:

  cpmCPUTotalTable = poll("cpmCPUTotalIndex,
                          cpmCPUTotalPhysicalIndex,
                          cpmCPUTotal5minRev,
                          cpmCPUTotal1minRev");

Variables Collected:

  cpmCPUThresholdTable = poll("cpmCPUTotalIndex,
                              cpmCPUThresholdClass,
                              cpmCPURisingThresholdValue,
                              cpmCPUFallingThresholdValue");

-----

Variables Collected:

  cpmCPUTotalTable = poll("cpmCPUTotalIndex,
                          cpmCPUTotalPhysicalIndex,
                          cpmCPUTotal5min,
                          cpmCPUTotal1min");

Variables Collected:

  cpmCPUThresholdTable = poll("cpmCPUTotalIndex,
                              cpmCPUThresholdClass,
                              cpmCPURisingThresholdValue,
                              cpmCPUFallingThresholdValue");

-----

Variables Collected:

  hrProcessorTable = poll("hrDeviceIndex,
                          hrProcessorLoad");

Variables Collected:

```

```
hrDeviceTable = poll("hrDeviceIndex,
                    hrDeviceDescr");
```

Variables Collected:

```
ssCpuStats = poll("ssCpuRawUser,
                 ssCpuRawNice,
                 ssCpuRawSystem,
                 ssCpuRawIdle,
                 ssCpuRawWait,
                 ssCpuRawKernel,
                 ssCpuRawInterrupt,
                 ssCpuRawSoftIRQ,
                 ssRawInterrupts,
                 ssRawContexts");
```

=====

CSV File Format For cpu.xml Reports

Report ID: CPUOLD

MIB Used: CISCO_PROCESS_MIB_CPUOLD

=====

1 CSV Filename Prefix: CPU

1 TimeStamp	TimeStamp
2 Node	fqdnid
3 Slot	CPUSlot
4 Number	CPUNum
5 Description	CPUDescr
6 CPUUtilMax5min	Max(cpmCPUTotal5min / 100)
7 CPUUtilAvg5min	Avg(cpmCPUTotal5min / 100)
8 CPUUtilMax1min	Max(cpmCPUTotal1min / 100)
9 CPUUtilAvg1min	Avg(cpmCPUTotal1min / 100)
10 CPURisingThreshold	cpmCPURisingThresholdValue / 100
11 CPUFallingThreshold	cpmCPUFallingThresholdValue / 100
12 ProcessorIndex	cpmCPUTotalIndex

=====

Web Reports For cpu.xml Reports

Report ID: CPUOLD

MIB Used: CISCO_PROCESS_MIB_CPUOLD

=====

1 CPU Activity Type Percentage

User Code	UserCPUPercent
System Code	SystemCPUPercent
Kernel Code	KernelCPUPercent
Nice Code	NiceCPUPercent
Idle	IdleCPUPercent
IO Wait	WaitCPUPercent
H/W Interrupts	HWInterruptCPUPercent
S/W Interrupts	SWInterruptCPUPercent
Device	fqdnid
Interrupts	Interrupts
Context Switches	ContextSwitches

2 CPU Utilization

Average Utilization	Avg(cpmCPUTotal5min / 100)
Peak Utilization	Max(cpmCPUTotal5min / 100)
Device	fqdnid
Slot	CPUSlot
CPU	CPUNum

```
CPU Description          CPUDESCR
Avg                      Avg(cpmCPUTotal1min / 100)
Peak                     Max(cpmCPUTotal1min / 100)
```

=====
 CSV File Format For cpu.xml Reports

Report ID: UCD_CPU MIB Used: UCD_CPU
 =====

1 CSV Filename Prefix: UCD_CPU_ACTIVITY

```
1 TimeStamp          TimeStamp
2 Node              fqdnid
3 UserPercent       Avg(userCPUTicks    / totalTicks)
4 SystemPercent     Avg(systemCPUTicks  / totalTicks)
5 KernelPercent     Avg(kernelCPUTicks  / totalTicks)
6 NicePercent       Avg(niceCPUTicks    / totalTicks)
7 IdlePercent       Avg(idleCPUTicks    / totalTicks)
8 IOWaitPercent     Avg(waitCPUTicks    / totalTicks)
9 H/WInterruptsPercent Avg(hwInterruptCPUTicks / totalTicks)
10 S/WInterruptsPercent Avg(swInterruptCPUTicks / totalTicks)
11 InterruptsCount  Sum(ssRawInterrupts.delta())
12 ContextSwitchesCount Sum(ssRawContexts.delta())
```

=====
 Web Reports For cpu.xml Reports

Report ID: UCD_CPU MIB Used: UCD_CPU
 =====

1 CPU Activity Type Percentage

```
User Code            Avg(userCPUTicks    / totalTicks)
System Code          Avg(systemCPUTicks  / totalTicks)
Kernel Code          Avg(kernelCPUTicks  / totalTicks)
Nice Code            Avg(niceCPUTicks    / totalTicks)
Idle                 Avg(idleCPUTicks    / totalTicks)
IO Wait              Avg(waitCPUTicks    / totalTicks)
H/W Interrupts      Avg(hwInterruptCPUTicks / totalTicks)
S/W Interrupts      Avg(swInterruptCPUTicks / totalTicks)
Device               fqdnid
Interrupts           Sum(ssRawInterrupts.delta())
Context Switches     Sum(ssRawContexts.delta())
```

2 CPU Utilization

```
Average Utilization  Avg(cpmCPUTotal5min / 100)
Peak Utilization     Max(cpmCPUTotal5min / 100)
Device               fqdnid
Slot                 CPUSlot
CPU                  CPUNum
CPU Description      CPUDESCR
Avg                  Avg(cpmCPUTotal1min / 100)
Peak                 Max(cpmCPUTotal1min / 100)
```

=====
 CSV File Format For cpu.xml Reports

Report ID: CPU MIB Used: CISCO_PROCESS_MIB_CPUREV
 =====

1 CSV Filename Prefix: CPU

```
1 TimeStamp          TimeStamp
```

```

2 Node                fqdnid
3 Slot                CPUSlot
4 Number              CPUNum
5 Description          if((cpuDescr == ""), "NoDescription", cpuDescr)
6 CPUUtilMax5min      Max(cpmCPUTotal5minRev / 100)
7 CPUUtilAvg5min      Avg(cpmCPUTotal5minRev / 100)
8 CPUUtilMax1min      Max(cpmCPUTotal1minRev / 100)
9 CPUUtilAvg1min      Avg(cpmCPUTotal1minRev / 100)
10 CPURisingThreshold cpmCPURisingThresholdValue / 100
11 CPUFallingThreshold cpmCPUFallingThresholdValue / 100
12 ProcessorIndex     cpmCPUTotalIndex

```

=====
Web Reports For cpu.xml Reports

Report ID: CPU

MIB Used: CISCO_PROCESS_MIB_CPUREV

=====

1 CPU Activity Type Percentage

```

User Code              Avg(userCPUTicks      / totalTicks)
System Code            Avg(systemCPUTicks    / totalTicks)
Kernel Code            Avg(kernelCPUTicks    / totalTicks)
Nice Code              Avg(niceCPUTicks      / totalTicks)
Idle                   Avg(idleCPUTicks      / totalTicks)
IO Wait                Avg(waitCPUTicks      / totalTicks)
H/W Interrupts         Avg(hwInterruptCPUTicks / totalTicks)
S/W Interrupts         Avg(swInterruptCPUTicks / totalTicks)
Device                 fqdnid
Interrupts             Sum(ssRawInterrupts.delta())
Context Switches       Sum(ssRawContexts.delta())

```

2 CPU Utilization

```

Average Utilization    Avg(cpmCPUTotal5minRev / 100)
Peak Utilization       Max(cpmCPUTotal5minRev / 100)
Device                 fqdnid
Slot                   CPUSlot
CPU                    CPUNum
CPU Description         if((cpuDescr == ""), "NoDescription", cpuDescr)
Avg                     Avg(cpmCPUTotal1minRev / 100)
Peak                   Max(cpmCPUTotal1minRev / 100)

```

=====
CSV File Format For cpu.xml Reports

Report ID: HOST_RESOURCES_CPU

MIB Used: HOST_RESOURCES_CPU_MIB

=====

1 CSV Filename Prefix: CPU

```

1 TimeStamp           TimeStamp
2 Node                fqdnid
3 Slot                0
4 Number              hrDeviceIndex
5 Description          hrDeviceDescr
6 CPUUtilMax5min      Max(oneMinUtil)
7 CPUUtilAvg5min      Avg(oneMinUtil)
8 CPUUtilMax1min      Max(hrProcessorLoad/100)
9 CPUUtilAvg1min      Avg(hrProcessorLoad/100)
10 CPURisingThreshold 0
11 CPUFallingThreshold 999
12 ProcessorIndex     hrDeviceIndex

```

=====

Web Reports For cpu.xml Reports

Report ID: HOST_RESOURCES_CPU

MIB Used: HOST_RESOURCES_CPU_MIB

=====

1 CPU Activity Type Percentage

```
-----
User Code                Avg(userCPUTicks      / totalTicks)
System Code              Avg(systemCPUTicks    / totalTicks)
Kernel Code              Avg(kernelCPUTicks    / totalTicks)
Nice Code                 Avg(niceCPUTicks      / totalTicks)
Idle                     Avg(idleCPUTicks      / totalTicks)
IO Wait                   Avg(waitCPUTicks      / totalTicks)
H/W Interrupts           Avg(hwInterruptCPUTicks / totalTicks)
S/W Interrupts           Avg(swInterruptCPUTicks / totalTicks)
Device                    fqdnid
Interrupts                Sum(ssRawInterrupts.delta())
Context Switches          Sum(ssRawContexts.delta())
```

2 CPU Utilization

```
-----
Average Utilization      Avg(oneMinUtil)
Peak Utilization          Max(oneMinUtil)
Device                    fqdnid
Slot                       0
CPU                        hrDeviceIndex
CPU Description           hrDeviceDescr
Avg                       Avg(hrProcessorLoad/100)
Peak                       Max(hrProcessorLoad/100)
```

Implementation Notes for cpu

=====

Steps for setting CPU to be monitored in snmpd.conf file:

net-snmp should be installed before doing the following steps.

Linux

1. Open the file /etc/snmp/snmpd.conf
2. Add the following line


```
load 90 80 70
```
3. Save the file and restart as follows


```
/etc/init.d/snmpd restart
```

Solaris

1. Open the file /etc/sma/snmp/snmpd.conf
2. Add the following line


```
load 90 80 70
```
3. Save the file and restart as follows


```
/etc/init.d/init.sma restart
```

レポートのヘルプ

オンライン レポートのヘルプを表示するには、次を選択します：

- [Help] > [READMEs] を選択し CLI コマンド：README、CHANGE、デバイス情報、CLI コマンド、CLI コマンドのヘルプ、リリース ノート、クイック スタートを表示します。
- [Help] > [Reports]：表示システムは、『System Reports ReadmeReadme』、『User Reports Readme』、XML 定義のレポート、『Reports List Readme』、IETF RFC、SNMP MIB、システム機能定義、ユーザ機能定義を表示します。

指定可能な項目

ポーリングされる MIB とレポートされる MIB 変数のほかに、レポート ビュー、レポート時間間隔、データのソート順序を指定できます。

レポート ビュー：グラフ、テーブル、および CSV ファイル

XML インターフェイスでは、ユーザに次のビューを提供するレポートをコーディングできます。

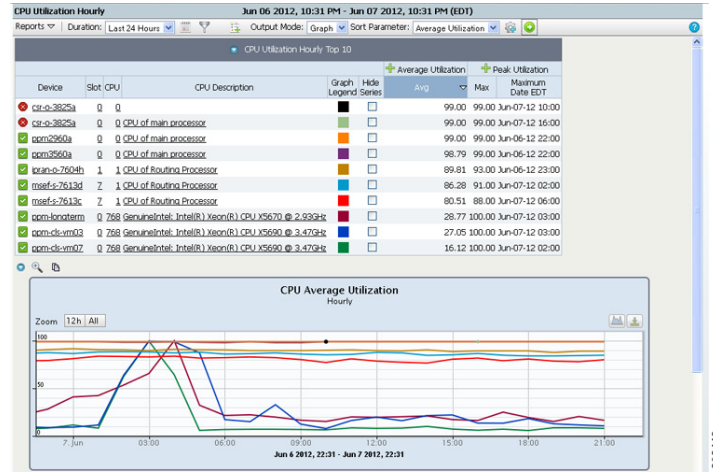
- **グラフ ビュー**：指定された時間間隔のパフォーマンスのグラフを表示します。
- **テーブル ビュー**：パフォーマンスを表形式で表示します。
- **CSV ファイル ビュー**：ユーザは、スプレッドシートやテキスト エディタを使用して表示可能なカンマ区切り値 (CSV) ファイルにレポートを保存できます。

このガイドのチュートリアル第 4 章（「[レポートの記述](#)」）では、cpu.xml レポートのコーディングについて一通り説明します。このレポートでは、ネットワーク全体またはユーザが選択したデバイスの CPU 使用率が示されます。

グラフ ビュー

図 1-1 に、CPU 使用率レポートのグラフ表示ビューの例を示します。

図 1-1 グラフ表示ビューの CPU 使用率レポート



独自レポートをコーディングする場合、ユーザが選択できる時間間隔を制御できます。

テーブル ビュー

図 1-2 に、CPU 使用率レポートのテーブル出力ビューの例を示します。

図 1-2 テーブル ビュー

Device	Slot	CPU	CPU Description	Timestamp EDT	5 Min Util	1 Min Util	Peak	Avg
rom0-prim0	0	756	GenuineIntel: Intel(R) Xeon(R) CPU X5670 @ 2.93GHz	Jun-07-12 04:00	100.00	100.00	100.00	100.00
rom0-sl0m07	0	756	GenuineIntel: Intel(R) Xeon(R) CPU X5650 @ 3.47GHz	Jun-07-12 03:00	99.60	100.00	99.60	100.00
rom0-sl0m02	0	756	GenuineIntel: Intel(R) Xeon(R) CPU X5650 @ 3.47GHz	Jun-07-12 03:00	99.60	100.00	99.60	100.00
rom0560a	0	0	CPU of main processor	Jun-06-12 22:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-06-12 22:00	99.00	99.00	99.00	99.00
rom0560a	0	0	CPU of main processor	Jun-06-12 23:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-06-12 23:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 00:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 01:00	99.00	99.00	99.00	99.00
rom0560a	0	0	CPU of main processor	Jun-07-12 01:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 01:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 02:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 03:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 04:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 04:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 05:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 06:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 07:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 08:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 09:00	99.00	99.00	99.00	99.00
cr-c-3825a	0	0		Jun-07-12 10:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 10:00	99.00	99.00	99.00	99.00
rom0560a	0	0	CPU of main processor	Jun-07-12 10:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 11:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 11:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 12:00	99.00	99.00	99.00	99.00
rom0560a	0	0	CPU of main processor	Jun-07-12 12:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 13:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 13:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 14:00	99.00	99.00	99.00	99.00
rom0560a	0	0	CPU of main processor	Jun-07-12 14:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 15:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 16:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 16:00	99.00	99.00	99.00	99.00
rom0560a	0	0	CPU of main processor	Jun-07-12 16:00	99.00	99.00	99.00	99.00
rom0260a	0	0	CPU of main processor	Jun-07-12 17:00	99.00	99.00	99.00	99.00
rom0560a	0	0	CPU of main processor	Jun-07-12 17:00	99.00	99.00	99.00	99.00

レポート間隔

Cisco Prime レポートの XML スキーマにより、レポートのレポート間隔を複数指定できます。

ソート順序

レポート上でデータがソートされる順序を指定できます。

基本的なレポート カテゴリ

基本的なレポート カテゴリには、ネットワーク レベル レポート、デバイス レベル レポート、および特定の変数に関するレポートの 3 種類があります。レポートを初めて表示したときは、検出されたネットワーク全体の統計情報が表示されます。その後、特定のデバイスを選択して、単一デバイスのレポートを表示することができます。

レポート管理インターフェイス

次のレポート管理タブが、[Performance] メニューおよび次のいずれかから [Reports] を選択することで GUI Prime Performance Manager から使用可能です。

- [Report Settings] : さまざまなレポート間隔を可能にしレポートのエージングを制御できます。
- [Report Status] : レポートを有効または無効にできます。

- [Report Policies] : レポート ポリシーを作成または管理できます。
ユーザ インターフェイスの詳細については、『[Cisco Prime Performance Manager 1.4 User Guide](#)』を参照してください。

