

EmbedWare/SysMon Ver 7.1 for Linux

説明書

このページは空白です。

ごあいさつ

このたびは、「EmbedWare/SysMon® Ver 7.1」をご利用いただき、誠にありがとうございます。
本書は、ハードウェアの動作状態を監視する EmbedWare/SysMon® Ver 7.1 のインストール方法や、
操作方法などについて記述しています。本書をよくお読みになって正しくお使いください。また、
本書を大切に保管していただきますようお願いいたします。
本書が、EmbedWare/SysMon® Ver 7.1 を活用していただくために、みなさまのお役に立つことを
願っております。

2019年11月

EmbedWare、EmbedWare/SysMon、EmbedWare/SysMon Entry は、株式会社 PFU の日本国における登録商標です。

Linux® は、Linus Torvalds 氏の米国およびその他の国における登録商標あるいは商標です。

Red Hat および Red Hat をベースとしたすべての商標とロゴは、米国その他の国で Red Hat, Inc. の登録商標若しくは商標です。

Ubuntu は、Canonical Ltd. の登録商標です。

その他の会社名、製品名などは、各社の商標または、登録商標です。

© PFU Limited 2018-2019

●本書の構成について

第1部 *EmbedWare/SysMon Entry*

第1章 概要

EmbedWare/SysMon Entry の概要について説明します。

第2章 インストールとアンインストール

EmbedWare/SysMon Entry のインストール／アンインストール方法について説明します。

第3章 *EmbedWare/SysMon Entry を利用する*

EmbedWare/SysMon Entry で表示、設定可能な情報について説明します。

第2部 *EmbedWare/SysMon SDK*

第1章 概要

EmbedWare/SysMon SDK の概要について説明します。

第2章 インストールとアンインストール

EmbedWare/SysMon SDK のインストール／アンインストール方法について説明します。

第3章 API 仕様

EmbedWare/SysMon SDK を利用するにあたって、必要な API 仕様について説明します。

第4章 API 使用方法

EmbedWare/SysMon SDK の API 使用法について説明します。

付録

EmbedWare/SysMon Entry の通知メッセージ、コマンドラインインターフェイスメッセージ、EmbedWare/SysMon SDK API ディスク情報取得、およびシステムエラー監視の仕組みについて説明します。

●本書の表記について

本書で使用している記号とその意味を以下に示します。

 注意	お使いになるときに注意していただきたいことや、してはいけないことを記述しています。 必ずお読みください。
 備考	操作に関する補足事項を記述しています。 必要に応じてお読みください。
 参照	関連する情報が記載されているマニュアルや、参照先を示しています。 必要に応じてお読みください。

●略称

本書では、以下の用語について省略して表記する場合がありますのでご了承ください。

正式名称	略称
Red Hat® Enterprise Linux®	Red Hat Enterprise Linux
EmbedWare/SysMon®	EmbedWare/SysMon

目次

第1部 EmbedWare/SysMon Entry

第1章 概要

1.1	EmbedWare/SysMon Entry とは	3
1.2	機能	3
1.3	動作環境	5

第2章 インストールとアンインストール

2.1	インストール	7
2.2	アンインストール	9
2.3	版数確認	10

第3章 EmbedWare/SysMon Entry を利用する

3.1	温度監視	11
(1)	現在の温度の表示	11
(2)	過去最高値・過去最低値の表示	11
(3)	監視条件の設定	11
3.2	電圧監視	12
(1)	現在の電圧の表示	12
(2)	過去最高値・過去最低値の表示	12
(3)	監視条件の設定	12
3.3	ファン監視	12
(1)	現在のファン回転数の表示	12
(2)	過去最高値・過去最低値の表示	12
(3)	監視条件の設定	12
3.4	S.M.A.R.T. 監視	13
(1)	S.M.A.R.T. ステータスの表示	13
(2)	ハードディスクモデル、シリアル番号の表示	13
(3)	S.M.A.R.T. アトリビュート情報の表示	13
(4)	各ディスクのモニタリング オン／オフ	13
3.5	RAID 監視	13
(1)	RAID 情報の表示	13
(2)	監視機能の有効・無効切替え	13
3.6	SVRx RAID 監視	14
(1)	SVRx RAID 情報の表示	14

(2) 監視機能の有効・無効切替え	14
(3) 保守情報採取	14
3.7 システムエラー監視.....	14
(1) PCI エラー情報の表示	14
(2) メモリエラー情報の表示	14
(3) BMC エラー情報の表示	14
(4) 監視機能の有効・無効切替え	15
3.8 システム情報の表示・ログの表示.....	15
(1) BIOS 版数の表示	15
(2) 総通電時間の表示	15
(3) EmbedWare/SysMon イベントログの表示	15
(4) BIOS イベントログの表示	15
3.9 パフォーマンス情報の表示.....	16
(1) CPU 周波数の表示	16
(2) 監視機能の有効・無効切替え	16
3.10 ウオッチドッグタイマ設定.....	16
(1) ウオッチドッグタイマ設定.....	16
3.11 設定の表示	16
(1) 各種オプション設定	16
3.12 BMC SEL	17
(1) BMC SEL の表示	17
(2) BMC SEL 設定	17
3.13 コマンドラインインターフェイス	17
(1) ewsmondisp	17
(2) ewsmonset	19
(3) ewssmart	22
(4) ewssraid	24
(5) ewssvraid	25
(6) ewssyerror	26
(7) ewssinfo	27
(8) ewslog	28
(9) ewsperform	29
(10)ewsctl	30
(11)ewsbmcsel	32
(12)ewssvlog	34

第2部 EmbedWare/SysMon SDK

第1章 概要

1.1	EmbedWare/SysMon SDK とは	37
1.2	機能	37
1.3	アプリケーション開発環境	38
(1)	インストール環境	38
(2)	必要ファイル	38
1.4	アプリケーション実行環境	38

第2章 インストールとアンインストール

2.1	インストール	39
2.2	アンインストール	40
2.3	版数確認	41

第3章 API仕様

3.1	ESM_Open() 関数	43
3.2	ESM_Close() 関数	44
3.3	ESM_Get() 関数	45
3.4	ESM_GetSmart() 関数	50
3.5	ESM_RAIDInfo() 関数	51
3.6	ESM_SvRAIDInfo() 関数	53
3.7	ESM_QueryError() 関数	54
3.8	ESM_GetWDTTimeOut() 関数	55
3.9	ESM_SwitchWDT() 関数	56
3.10	ESM_WDTHeartbeat() 関数	57
3.11	ESM_ReadGPI() 関数	57
3.12	ESM_ReadGPO() 関数	58
3.13	ESM_WriteGPO() 関数	58
3.14	ESM_GetCpuCount() 関数	59
3.15	ESM_GetCpuPerform() 関数	59
3.16	ESM_GetSmartEx() 関数	60
3.17	ESM_GetSvDiskType() 関数	61
3.18	ESM_GetSvDiskSmartAttr() 関数	62

第4章 API 使用方法

4.1 API マルチスレッド対応	63
4.2 API 使用例.....	64

付録

付録 A 通知メッセージ一覧	65
付録 B コマンドラインインターフェイスメッセージ一覧	69
付録 C EmbedWare/SysMon SDK API ディスク情報取得について	71
付録 D システムエラー監視の仕組みについて	72

第1部 EmbedWare/ SysMon Entry

第1章 概要

1

概要

この章では、EmbedWare/SysMon Entry の概要について説明します。

1.1 EmbedWare/SysMon Entry とは

コンピューターのハードウェアが正常な動作状態にあるかを監視するソフトウェアです。EmbedWare/SysMon Entry は、コンピューターの状態を常時監視し、異常を検出した場合に、コンピューターの利用者に通知します。

1.2 機能

EmbedWare/SysMon Entry には、以下の機能があります。

機能	説明
状態監視	コンピューターの温度、電圧値、ファン回転数の現在の値を表示します。これらの値には、上限または下限値があらかじめ設定されており、異常の有無を容易に確認できます。
S.M.A.R.T. 監視	内蔵ディスクデバイスの S.M.A.R.T. (Self-Monitoring, Analysis and Reporting Technology) 機能による異常検出の有無を表示します。
RAID 監視	RAID カード (タイプ 1)、RAID カード (タイプ 2) の状態を表示します。RAID カード (タイプ 2) は、保守情報を採取できます。
システムエラー監視	BIOS イベントログの記録に基づいて、PCI バス、メモリ、および BMC エラーの有無を表示します。
パフォーマンス監視	CPU の実周波数情報を表示します。
異常通知	状態監視、ディスク監視、システムエラー監視で異常を検出した場合、ビープ音を鳴らし、シリログにメッセージを格納します。
ログ記録、表示	状態監視、ディスク監視、システムエラー監視で異常を検出した場合、EmbedWare/SysMon のイベントログファイルに記録します。EmbedWare/SysMon のイベントログのほかに、シリログへの記録もできます。また、ログ表示機能では、EmbedWare/SysMon のイベントログ、BIOS イベントログ、および BMC SEL (Baseboard Management Controller System Event Log) を表示します。 EmbedWare/SysMon イベントログは EmbedWare/SysMon Entry で独自に記録しているログです。異常を検出した場合にメッセージを記録します。
シャットダウン処理	状態監視で異常を検出した場合、コンピューターの電源を切斷します。

機能	説明
トレース機能	状態監視の履歴をトレースファイルに記録します。
過去最高値・過去最低値の表示	状態監視で記録した、過去最高値・過去最低値を表示します。
ウォッチ ドッグタイマ	EmbedWare/SysMon Entry のサービスプログラムが、一定時間動作できない状態になった場合、事前に指定されているアクションを実施します。 ほかのアプリケーションでウォッチ ドッグタイマを制御する場合は、EmbedWare/SysMon SDK を使用してください。その場合、EmbedWare/SysMon Entry のウォッチ ドッグタイマ設定をオフにしてください。
システム情報表示	コンピューターの BIOS 版数、総通電時間を表示します。
BMC SEL 制御	BMC SEL とは、BMC 内部で保持されるシステムイベントログです。 BMC SEL のクリア実行および、記録サイズが最大値に達した場合の動作を設定します。



- EmbedWare/SysMon Entry は、監視対象項目のエラーを検出しても、装置のアラームランプを点灯しません。
- BMC に関する機能は BMC 搭載機種で使用可能です。
- RAID 監視で監視対象としている RAID カードは、2 種類存在します。
 - RAID カード（タイプ 1）は PCIe インターフェイスカードタイプです。
 - RAID カード（タイプ 2）は SATA インターフェイスカードタイプです。

機種によって搭載される RAID カードは異なります。
搭載 RAID カードが不明な場合は、担当営業にご相談ください。

1.3 動作環境

EmbedWare/SysMon Entry の動作環境を以下に示します。

ディストリビューション : Red Hat Enterprise Linux
Ubuntu

ディスク資源 : 約 4.2MB ~ 約 67MB

メモリ資源 : 監視デーモン 約 1.6MB ~ 約 5.0MB

BMC 搭載の有無	IPMI ドライバ
BMC 搭載機種	必須 (IPMI ドライバは、OS インストール時に標準インストールされます)
BMC 非搭載機	不要



- SMBus (System Management Bus) にアクセスするアプリケーションがある場合、動作保証されません。
- ウオッチドッグタイマ設定を行う場合は、本製品以外にウォッチドッグタイマへのアクセスを行っていないことを確認してください。本製品以外にアクセスがある場合、動作保証されません。
- 対応ディストリビューションおよびバージョンについての詳細は、ハードウェアのカタログまたはマニュアルでご確認ください。
- 上記以外の環境をご使用のお客様は、担当営業にご相談ください。

第2章 インストールとアンインストール

この章では、EmbedWare/SysMon Entry のインストールとアンインストールの方法について説明します。

2.1 インストール

◇ ipmi_devintf、ipmi_si のロード方法

BMC 搭載機種の場合、ipmi_devintf、ipmi_si のモジュールを永続的にロードする必要があります。Red Hat Enterprise Linux7.2 および Ubuntu16.04 LTS の手順例を、次に示します。



ipmi_devintf、ipmi_si のモジュールは、OS インストール時に標準でインストールされています。

1 lsmod コマンドを実行し、ipmi_devintf、ipmi_si が永続的にロードされているかを確認します。

コマンドの実行結果に ipmi_devintf、ipmi_si の表示がある場合は、モジュールが永続的にロードされていますので以降の作業は不要です。

コマンドの実行結果に ipmi_devintf、ipmi_si の表示がない場合は、モジュールが永続的にロードされていません。

/etc/modules-load.d/ ディレクトリに conf ファイルを作成し、以降の作業を行ってください。



conf ファイル名は任意です。

参考

2 conf ファイルを編集し、永続的にロードするモジュール (ipmi_devintf、ipmi_si) を記載します。

```
ipmi_devintf
ipmi_si
```

3 装置を再起動します。

◇ OS のウォッチドッグタイマを無効にする

EmbedWare/SysMon でウォッチドッグタイマ設定を行う場合、本製品以外のウォッチドッグタイマへのアクセスを無効にする必要があります。OS のウォッチドッグタイマを無効にする手順例を、次に示します。

1 /etc/modprobe.d/blacklist.conf を編集し、以下記述を追加します。

```
blacklist iTCO_wdt
```



/etc/modprobe.d/blacklist.conf が存在しない場合は、新規作成してください。

◇ Red Hat Enterprise Linux でインストールする

Red Hat Enterprise Linux でインストールする方法を以下に示します。

- 1 root 権限でログインします。**

```
$ su -  
Password: ***** ←root のパスワードを入力
```

- 2 EmbedWare/SysMon Entry のインストールプログラムがあるディレクトリに移動して、以下のコマンドでファイルの有無を確認します。**

```
# ls  
install-ews-entry.sh
```

- 3 EmbedWare/SysMon Entry のインストールを開始します。**

```
# bash install-ews-entry.sh
```

◇ Ubuntu でインストールする

Ubuntu でインストールする方法を以下に示します。

- 1 EmbedWare/SysMon Entry のインストールプログラムがあるディレクトリに移動して、以下のコマンドでファイルの有無を確認します。**

```
$ ls  
install-ews-entry.sh
```

- 2 EmbedWare/SysMon Entry のインストールを開始します。**

```
$ sudo bash install-ews-entry.sh
```



インストールに失敗した場合は、製品格納ディレクトリおよびその配下にファイルが残る場合があります。再度インストールを行い、インストールを完了させてください。

2.2 アンインストール

◇ Red Hat Enterprise Linux でアンインストールする

Red Hat Enterprise Linux でアンインストールする方法を以下に示します。

- 1 root 権限でログインします。

```
$ su -
Password: ***** ←root のパスワードを入力
```

- 2 パッケージの問い合わせをします。以下のコマンドを使用して、EmbedWare/SysMon Entry がインストールされているかどうかを確認します。

```
# rpm -q ews-entry
ews-entry-7.x.x.x
```

- 3 EmbedWare/SysMon Entry をアンインストールします。

```
# rpm -e ews-entry
```

◇ Ubuntu でアンインストールする

Ubuntu でアンインストールする方法を以下に示します。

- 1 パッケージの問い合わせをします。以下のコマンドを使用して、EmbedWare/SysMon Entry がインストールされているかどうかを確認します。

```
$ dpkg -l ews-entry
||/ Name      Version       Description
+++-=====
ii  ews-entry   7.x.x.x     xxxx
```

- 2 EmbedWare/SysMon Entry をアンインストールします。

```
$ sudo dpkg -r ews-entry
```

◇ ipmi_devintf、ipmi_si のロードの解除方法

BMC 搭載機種で、ipmi_devintf、ipmi_si のモジュールの永続的ロードが不要な場合は、解除してください。

Red Hat Enterprise Linux7.2 および Ubuntu16.04 LTS の手順例を、次に示します。

- 1 /etc/modules-load.d/ ディレクトリに作成した conf ファイルを削除します。
- 2 装置を再起動します。

2.3 版数確認

◇ Red Hat Enterprise Linux で版数を確認する

以下のコマンドで版数を確認します。

```
# rpm -q ews-entry  
ews-entry-7.x-x.x
```

◇ Ubuntu で版数を確認する

以下のコマンドで版数を確認します。

```
$ dpkg -l ews-entry  
||/ Name      Version       Description  
=====  
ii  ews-entry   7.x.x.x     xxxx
```

第3章 EmbedWare/SysMon Entry を利用する



この章では、EmbedWare/SysMon Entry の利用方法について説明します。

EmbedWare/SysMon Entry では、コマンドラインインターフェイスで情報の表示および各種設定が可能です。



- 機種によって、一部の機能が存在しない場合があります。
- root 権限を持たない場合は、情報の参照だけ可能です。設定の変更はできません。
- BMC 搭載機種の温度監視、電圧監視、ファン監視で異常が発生したあとに、現在値が正常範囲に戻ったにもかかわらず、状態判定が異常のままになることがあります。状態判定に使用されるしきい値がヒステリシスを持っているためです。

3.1 温度監視

温度監視で使用できる機能は、以下のとおりです。詳細については、各コマンドラインインターフェイスを参照してください。

(1) 現在の温度の表示



「3.13 コマンドラインインターフェイス」の「(1) ewsmondisp」(17 ページ)

(2) 過去最高値・過去最低値の表示



「3.13 コマンドラインインターフェイス」の「(1) ewsmondisp」(17 ページ)

(3) 監視条件の設定



「3.13 コマンドラインインターフェイス」の「(2) ewsmonset」(19 ページ)

3.2 電圧監視

電圧監視で使用できる機能は、以下のとおりです。詳細については、各コマンドラインインターフェイスを参照してください。

(1) 現在の電圧の表示



「3.13 コマンドラインインターフェイス」の「(1) ewsmondisp」(17 ページ)

(2) 過去最高値・過去最低値の表示



「3.13 コマンドラインインターフェイス」の「(1) ewsmondisp」(17 ページ)

(3) 監視条件の設定



「3.13 コマンドラインインターフェイス」の「(2) ewsmonset」(19 ページ)

3.3 ファン監視

ファン監視で使用できる機能は、以下のとおりです。詳細については、各コマンドラインインターフェイスを参照してください。

(1) 現在のファン回転数の表示



「3.13 コマンドラインインターフェイス」の「(1) ewsmondisp」(17 ページ)

(2) 過去最高値・過去最低値の表示



「3.13 コマンドラインインターフェイス」の「(1) ewsmonisp」(17 ページ)

(3) 監視条件の設定



「3.13 コマンドラインインターフェイス」の「(2) ewsmonset」(19 ページ)

3.4 S.M.A.R.T. 監視

S.M.A.R.T. 監視で使用できる機能は、以下のとおりです。詳細については、各コマンドラインインターフェイスを参照してください。

(1) S.M.A.R.T. ステータスの表示



「3.13 コマンドラインインターフェイス」の「(3) ewssmart」(22 ページ)

(2) ハードディスクモデル、シリアル番号の表示



「3.13 コマンドラインインターフェイス」の「(3) ewssmart」(22 ページ)

(3) S.M.A.R.T. アトリビュート情報の表示



「3.13 コマンドラインインターフェイス」の「(3) ewssmart」(22 ページ)

(4) 各ディスクのモニタリング オン／オフ



「3.13 コマンドラインインターフェイス」の「(3) ewssmart」(22 ページ)

3.5 RAID 監視

RAID カード（タイプ 1）監視で使用できる機能は、以下のとおりです。詳細については、各コマンドラインインターフェイスを参照してください。

(1) RAID 情報の表示



「3.13 コマンドラインインターフェイス」の「(4) ewsraids」(24 ページ)

(2) 監視機能の有効・無効切替え



「3.13 コマンドラインインターフェイス」の「(4) ewsraids」(24 ページ)

3.6 SVRx RAID 監視

RAID カード（タイプ 2）監視で使用できる機能は、以下のとおりです。詳細については、各コマンドラインインターフェイスを参照してください。

(1) SVRx RAID 情報の表示



「3.13 コマンドラインインターフェイス」の「(5) ewssvraid」(25 ページ)

(2) 監視機能の有効・無効切替え



「3.13 コマンドラインインターフェイス」の「(5) ewssvraid」(25 ページ)

(3) 保守情報採取



「3.13 コマンドラインインターフェイス」の「(12) ewssvlog」(34 ページ)

3.7 システムエラー監視

システムエラー監視で使用できる機能は、以下のとおりです。詳細については、各コマンドラインインターフェイスを参照してください。

(1) PCI エラー情報の表示



「3.13 コマンドラインインターフェイス」の「(6) ewssyserror」(26 ページ)

(2) メモリエラー情報の表示



「3.13 コマンドラインインターフェイス」の「(6) ewssyserror」(26 ページ)

(3) BMC エラー情報の表示



「3.13 コマンドラインインターフェイス」の「(6) ewssyserror」(26 ページ)

(4) 監視機能の有効・無効切替え



「3.13 コマンドラインインターフェイス」の「(6) ewssyserror」(26 ページ)

3.8 システム情報の表示・ログの表示

(1) BIOS 版数の表示



「3.13 コマンドラインインターフェイス」の「(7) ewssinfo」(27 ページ)

(2) 総通電時間の表示



「3.13 コマンドラインインターフェイス」の「(7) ewssinfo」(27 ページ)

(3) EmbedWare/SysMon イベントログの表示

- ログ世代管理について

EmbedWare/SysMon イベントログは /var/opt/EmbedWare_SysMon/Entry/log/sysmon.log に記録されます。ファイルサイズが 1MB に達すると、sysmon.log の末尾に数字を追加した名前に変名し、新たに sysmon.log が作成されます。ファイル名の数字が大きい方が、古い世代となります。

作成されるファイルの例 : sysmon.log.0



「3.13 コマンドラインインターフェイス」の「(8) ewslog」(28 ページ)

(4) BIOS イベントログの表示

- ログ世代管理について

BIOS イベントログは /var/opt/EmbedWare_SysMon/Entry/log/biosevent.log に記録されます。ファイルサイズが 1MB に達すると、biosevent.log の末尾に数字を追加した名前に変名し、新たに biosevent.log が作成されます。ファイル名の数字が大きい方が、古い世代となります。

作成されるファイルの例 : biosevent.log.0



「3.13 コマンドラインインターフェイス」の「(8) ewslog」(28 ページ)

3.9 パフォーマンス情報の表示

(1) CPU 周波数の表示



「3.13 コマンドラインインターフェイス」の「(9) ewsperform」(29 ページ)

(2) 監視機能の有効・無効切替え



「3.13 コマンドラインインターフェイス」の「(9) ewsperform」(29 ページ)

3.10 ウオッチドッグタイマ設定

(1) ウオッチドッグタイマ設定



「3.13 コマンドラインインターフェイス」の「(10) ewsctl」(30 ページ)

3.11 設定の表示

(1) 各種オプション設定



「3.13 コマンドラインインターフェイス」の「(10) ewsctl」(30 ページ)

3.12 BMC SEL

(1) BMC SEL の表示



「3.13 コマンドラインインターフェイス」の「(11) ewsbtmcsl」(32 ページ)

(2) BMC SEL 設定



「3.13 コマンドラインインターフェイス」の「(11) ewsbtmcsl」(32 ページ)

3.13 コマンドラインインターフェイス

EmbedWare/SysMon Entry のコマンドラインインターフェイスの詳細を記述します。すべてのコマンドの実行には、root (システム管理者) 権限が必要です。

(1) ewsmondisp

[名前]

`ewsmondisp`　温度、電圧、ファン回転数情報の表示

[書式]

`ewsmondisp [-w [-r]] [-t] [-v] [-f]`

[説明]

温度、電圧、ファン回転数の現在の値とステータス、または過去最高値・過去最低値を表示します。オプションの指定によって表示する情報を絞り込むことができます。

[オプション]

- w このオプションを指定した場合は温度、電圧、ファン回転数の過去最高値・過去最低値と発生日時を表示します。指定しない場合は現在の値とステータスを表示します。
- r -w と一緒に指定する場合に有効です。現在記録されている過去最高値・過去最低値をリセットします。
- t 温度についてだけ、情報表示または最高値・最低値をリセットします。動作内容は -w、-r オプションの有無に依存します。-v、-f との同時指定も可能です。
- v 電圧についてだけ、情報表示または最高値・最低値をリセットします。動作内容は -w、-r オプションの有無に依存します。-t、-f との同時指定も可能です。
- f ファン回転数についてだけ、情報表示または最高値・最低値をリセットします。動作内容は -w、-r オプションの有無に依存します。-t、-v との同時指定も可能です。

[実行例]

ewsmondisp

	Name	Value	Status	
	temp1	MB	32	NORMAL
	temp2	CPU	47	NORMAL
	voltage1	CPU core	1.1771	NORMAL
	voltage2	5VL	3.3460	ERROR
	voltage3	3.3V	3.2701	NORMAL
	voltage4	3.3VL	3.2873	NORMAL
	voltage5	5V	4.9800	NORMAL
	voltage6	12V	12.0157	NOT MONITORING
	voltage7	5V (MEM)	5.0015	NORMAL
	fan1	SYSFAN	3970	NORMAL

ewsmondisp -w

	Name	Lower date	Lower time	Upper date	Upper time	
	temp1	MB	12		40	
		160704	01:05:25	160704	01:02:25	
	temp2	CPU	25		51	
		160704	01:03:25	160704	01:02:25	
	voltage1	CPU core	1.2100		1.1600	
		160710	11:10:10	160710	09:51:45	
	voltage2	5VL	3.3460		3.6300	
		160710	11:10:10	160710	09:51:45	
	voltage3	3.3V	3.2652		3.3140	
		160710	11:10:10	160710	09:51:45	
	voltage4	3.3VL	3.2743		3.3065	
		160710	11:10:10	160710	09:51:45	
	voltage5	5V	4.9800		5.0010	
		160710	11:10:10	160710	09:51:45	
	voltage6	12V	12.0157		12.1157	
		160710	11:10:10	160710	09:51:45	
	voltage7	5V (MEM)	4.9800		5.1200	
		160710	11:10:10	160710	09:51:45	
	fan1	SYSFAN	3620		5201	
		160710	03:45:08	160710	03:45:08	

(2) ewsmonset

[名前]

ewsmonset 温度、電圧、ファン回転数監視の設定表示、変更

[書式]

```
ewsmonset [[ -t key1 on | off [key1 on | off] ... ]
           [-m key1 on | off [key1 on | off] ... ]
           [-u key1 value [key1 value] ... ]
           [-l key1 value [key1 value] ... ]
           [-s key2 on value | off [key2 on value | off] ... ] ] |
           [-d [key2]] | [-r]
```

[説明]

温度、電圧、ファン回転数監視について、以下の設定を表示、設定します。

- モニタリング設定（監視対象を実際にモニタするかどうか）
- トレース設定
- 異常検出のしきい値
- 異常検出時の電源断設定

オプションなしでこのコマンドを実行した場合、すべての設定を表示します。

設定を変更する場合、オプションと監視対象を示すキーワード（key1、key2）を指定します。key1はtemp1～、voltage1～、fan1～の中から指定します（実際に監視対象となっているセンサーは機種により異なります）。key2はtemp、voltage、fanの中から指定します。

[オプション]

- t key1 で指定した監視項目のトレースのオン／オフを設定します。key1は同時に複数指定できます。トレース情報（センサー名、温度・電圧・ファン回転数、記録した日付時刻）は、1秒間隔で「temp.trc」、「voltage.trc」、「fan.trc」ファイルに出力されます。ファイルは /var/opt/EmbedWare_SysMon/Entry/trc ディレクトリに格納されます。



トレースを採取するためには、トレース機能全体のオン／オフ設定を別途行う必要があります。詳細は、「(10) ewsctl」(30 ページ) を参照してください。

- m key1 で指定した監視項目のモニタリングのオン／オフを設定します。key1は同時に複数指定できます。
- u 温度、ファン回転数を正常と判断する上限のしきい値を変更します。key1にはtemp1～、fan1～を指定可能です。key1は同時に複数指定できます。指定可能なしきい値の有効範囲は ewsmonset -r を実行すれば表示されます。



- 機種によって、ファン回転数の上限のしきい値は存在しない場合があります。
- BMC 搭載機種の場合、しきい値はセンサーの分解能に合わせて調整される場合があります。

- l 温度、ファン回転数を正常と判断する下限のしきい値を指定します。key1にはtemp1～、fan1～を指定可能です。key1は同時に複数指定できます。指定可能なしきい値の有効範囲は、ewsmonset -r を実行すると表示されます。
- 備考**
 - 機種によって、温度の下限のしきい値は存在しない場合があります。
 - BMC 搭載機種の場合、しきい値はセンサーの分解能に合わせて調整される場合があります。
- s key2 で指定した監視カテゴリーで異常を検出した場合の装置電源断のオン／オフを設定します。オンを指定した場合、異常検出後、電源断開始までの時間を 0～60 秒の範囲で指定する必要があります。key2 は同時に複数指定できます。
- d key2 で指定した監視カテゴリーのしきい値設定、異常検出時の電源断設定の電源断開始までの時間を初期値に戻します。key2 を指定しない場合は、すべてのしきい値設定、異常検出時の電源断設定の電源断開始までの時間を初期値に戻します。このオプションは、ほかのオプションと一緒に指定できません。
- r 指定可能なしきい値の有効範囲を表示します。このオプションは、ほかのオプションとは同時に指定できません。

[実行例]

```
# ewsmonset
```

```
---Temperature-----
| Monitoring DispName Lower Upper Trace |
-----|
| temp1      on     MB      -      57    off   |
| temp2      on     CPU     20     115   off   |
-----|
```

if a threshold is breached, shutdown the system delay of 5 (0-60 sec) : off

```
----Voltage-----
| Monitoring DispName Lower Upper Trace |
-----|
| voltage1   on     CPU core  0.8060  1.5160 off   |
| voltage2   on     5VL     4.5000  5.5000 off   |
| voltage3   on     3.3V    2.9700  3.6300 off   |
| voltage4   on     3.3VL   2.9700  3.6300 off   |
| voltage5   on     5V      4.5000  5.5000 off   |
| voltage6   on     12V    10.8000 13.2000 off   |
| voltage7   on     5V (MEM) 4.5000  5.5000 off   |
-----|
```

if a threshold is breached, shutdown the system delay of 5 (0-60 sec) : off

```
----Fan-----
| Monitoring DispName Lower Upper Trace |
-----|
| fan1       on     SYSFAN  2656    -      off   |
-----|
```

if a threshold is breached, shutdown the system delay of 5 (0-60 sec) : off

```
# ewsmonset -r
```

```
---Temperature-----
| DispName Lower (Range) Upper (Range) |
-----|
| temp1    MB        -          0-100   |
-----|
```

temp2	CPU	0-120	0-120	
<hr/>				
--Fan				
DispName		Lower (Range)	Upper (Range)	

```
# ewsmonset -m temp1 off voltage1 off -t temp2 on voltage2 on -u temp1 40 temp2 50 -l fan1 4000 -s voltage
on 10
```

```
# ewsmonset
```

<hr/>				
--Temperature				
Monitoring	DispName	Lower	Upper	Trace

temp1	off	MB	-	40	off
temp2	on	CPU	20	50	on

if a threshold is breached, shutdown the system delay of 5 (0-60 sec) : off

<hr/>				
--Voltage				
Monitoring	DispName	Lower	Upper	Trace

voltage1	off	CPU core	0.8060	1.5160	off
voltage2	on	5VL	4.5000	5.5000	on
voltage3	on	3.3V	2.9700	3.6300	off
voltage4	on	3.3VL	2.9700	3.6300	off
voltage5	on	5V	4.5000	5.5000	off
voltage6	on	12V	10.8000	13.2000	off
voltage7	on	5V (MEM)	4.5000	5.5000	off

if a threshold is breached, shutdown the system delay of 10 (0-60 sec) : on

<hr/>				
--Fan				
Monitoring	DispName	Lower	Upper	Trace

fan1	on	SYSFAN	4000	-	off
------	----	--------	------	---	-----

if a threshold is breached, shutdown the system delay of 5 (0-60 sec) : off

```
# ewsmonset -d
```

```
# ewsmonset
```

<hr/>				
--Temperature				
Monitoring	DispName	Lower	Upper	Trace

temp1	off	MB	-	57	off
temp2	on	CPU	20	115	on

if a threshold is breached, shutdown the system delay of 5 (0-60 sec) : off

<hr/>				
--Voltage				
Monitoring	DispName	Lower	Upper	Trace

voltage1	off	CPU core	0.8060	1.5160	off
voltage2	on	5VL	4.5000	5.5000	on
voltage3	on	3.3V	2.9700	3.6300	off

```
| voltage4      on       3.3VL    2.9700   3.6300   off   |
| voltage5      on        5V     4.5000   5.5000   off   |
| voltage6      on       12V    10.8000  13.2000   off   |
| voltage7      on      5V(MEM) 4.5000   5.5000   off   |
-----
```

if a threshold is breached, shutdown the system delay of 5 (0-60 sec) : on

```
-----Fan-----
|           Monitoring      DispName    Lower     Upper     Trace   |
-----|
| fan1        on          SYSFAN     2656      -       off   |
-----
```

if a threshold is breached, shutdown the system delay of 5 (0-60 sec) : off

(3) ewssmart

[名前]

ewssmart S.M.A.R.T. ステータスとアトリビュート情報の表示

[書式]

ewssmart [key on | off [key on | off ...]] | [-a key] | [-v]

[説明]

引数、オプションなしで実行した場合、すべてのディスクの S.M.A.R.T. ステータス一覧を表示します。

引数として key を指定した場合、各ディスクのモニタリングのオン／オフを設定します。key には slot0 ~ slot7 を指定できます。複数 key の同時指定も可能です。

-a オプションと 1 個の key を指定した場合、指定したディスクの S.M.A.R.T. アトリビュートを表示します。

-v オプションを指定した場合、すべてのディスクの S.M.A.R.T. 詳細情報を表示します。



- EmbedWare/SysMon Ver7.0以前に使用していた disk1 ~ disk8 も key として指定可能です。disk1 ~ disk8 の番号は、slot の番号に 1 を加えた値です。
- 装置が AR シリーズの場合、SLOT の番号は装置に印字されているハードディスクの番号に対応します。

[オプション]

-v すべてのディスクの S.M.A.R.T. ステータス、モデル名、シリアル番号一覧を表示します。

-a key で指定したディスクの Attribute ID、Attribute Name、Flag、Value、Worst、Threshold、Raw Value の一覧を表示します。

[実行例]

ewssmart

	DispName	Status	Monitoring
	SLOT0	OK	on

#ewssmart -v

DispName,Status,Model,SerialNumber,Monitoring

SLOT0,OK,SATA589940,D8488EXEE32,on

#ewssmart -a slot0

ID	Attribute Name	Flag(hex)	Val	Wor	Thr	Raw	Value(hex)
1	Raw Read Error Rate	0b 00	100	100	50 00	00 00	00 00 00 00 00 00
2	Throughput Performance	05 00	100	100	50 00	00 00	00 00 00 00 00 00
3	Spin Up Time	27 00	100	100	1 a3	0c 00	00 00 00 00 00 00
4	Start Stop Count	32 00	100	100	0 e9	01 00	00 00 00 00 00 00
5	Reallocated Sector Count	33 00	100	100	50 00	00 00	00 00 00 00 00 00
7	Seek Error Rate	0b 00	100	100	50 00	00 00	00 00 00 00 00 00
8	Seek Time Performance	05 00	100	100	50 00	00 00	00 00 00 00 00 00
9	Power On Hours	32 00	98	98	0 9c	03 00	00 00 00 00 00 00
10	Spin Retry Count	33 00	109	100	30 00	00 00	00 00 00 00 00 00
12	Power Cycle Count	32 00	100	100	0 db	01 00	00 00 00 00 00 00
191	Mechanical Shock	32 00	100	100	0 04	00 00	00 00 00 00 00 00
192	Power-Off Retract Count	32 00	100	100	0 dd	00 00	00 00 00 00 00 00
193	Load Cycle Count	32 00	100	100	0 fb	01 00	00 00 00 00 00 00
194	Temperature Celsius	22 00	100	100	0 24	00 0f	00 30 00 00
196	Reallocated Event Count	32 00	100	100	0 00	00 00	00 00 00 00 00 00
197	Current Pending Sector	32 00	100	100	0 00	00 00	00 00 00 00 00 00
198	Offline Uncorrectable	30 00	100	100	0 00	00 00	00 00 00 00 00 00
199	UDMA CRC Error Count	32 00	200	253	0 1e	00 00	00 00 00 00 00 00
220	Disk Shift	02 00	100	100	0 00	00 00	00 00 00 00 00 00
222	Loaded Hours	32 00	98	98	0 90	03 00	00 00 00 00 00 00
223	Load/Unload Retry Count	32 00	100	100	0 00	00 00	00 00 00 00 00 00
224	Load Friction	22 00	100	100	0 00	00 00	00 00 00 00 00 00
226	Load-in Time	26 00	100	100	0 c6	00 00	00 00 00 00 00 00
240	Head Flying Hours	01 00	100	100	1 00	00 00	00 00 00 00 00 00

ewssmart disk1 off

ewssmart

	DispName	Status	Monitoring
	SLOT0	OK	off

(4) ewsraids

[名前]

ewsraids RAID カード（タイプ 1）のステータス表示と監視設定

[書式]

ewsraids [on | off]

[説明]

引数を指定しない場合は、RAID カード（タイプ 1）のステータスを表示します。デバイス名は CNTL（コントローラー番号）、CN（コネクタ番号）、DEV（デバイス番号）で表示します。

引数として on または off を指定した場合は、監視機能のオン／オフを切り替えます。

[実行例]

ewsraids

	Status	S.M.A.R.T.	Status	Array	RAID	
	(physical)		(logical)	No.	level	
CNTL01	CN00	DEV00	ONLINE	NORMAL	OK	0 RAID1
CNTL01	CN00	DEV01	ONLINE	NORMAL	OK	0 RAID1

ewsraids off

ewsraids

Monitoring off

(5) ewssvraid

[名前]

ewssvraid RAID カード（タイプ 2）のステータス表示と監視設定

[書式]

ewssvraid [on | off]

[説明]

引数を指定しない場合は、RAID カード（タイプ 2）のステータスを表示します。デバイス名は Drive（ディスク番号）で表示します。

引数として on または off を指定した場合は、監視機能のオン／オフを切り替えます。

[実行例]

ewssvraid

	Status	S.M.A.R.T.	Status	Array	RAID	
	(physical)		(logical)	No.	level	
Drive0	ONLINE	NORMAL	OK	0	RAID1	
Drive1	ONLINE	NORMAL	OK	0	RAID1	

ewssvraid off

ewssvraid

Monitoring off

(6) ewssyserror

[名前]

ewssyserror システムエラー監視情報の表示と設定変更

[書式]

ewssyserror {-p [reset | on | off] | -m [reset | on | off] | -b [reset | on | off]}

[説明]

指定したオプションに従って、装置起動時に取得した BIOS イベントログからどれかの動作を行います。

- 検出した PCI、メモリまたは BMC エラーを表示
- 検出したエラー状態のリセット
- 監視機能のオン／オフ



- BIOS イベントログを消去しても検出済みの PCI、メモリ、BMC エラーは、エラー状態のリセットをしない限り、エラー検出状態を解除しません。詳細は「付録 D システムエラー監視の仕組みについて」(72 ページ) を参照ください。
- エラー検出状態である場合、同一のエラーが再度発生しても無視されます。

[オプション]

- p パラメーターなしで実行した場合、PCI エラーの有無と発生日時を表示します。パラメーター reset を指定した場合、エラー記録をリセットします。パラメーター on または off を指定した場合、PCI エラー監視機能のオン／オフを切り替えます。
- m パラメーターなしで実行した場合、メモリエラーの有無と発生日時を表示します。パラメーター reset を指定した場合、エラー記録をリセットします。パラメーター on または off を指定した場合、メモリエラー監視機能のオン／オフを切り替えます。
- b パラメーターなしで実行した場合、BMC エラーの有無と発生日時を表示します。パラメーター reset を指定した場合、エラー記録をリセットします。パラメーター on または off を指定した場合、BMC エラー監視機能のオン／オフを切り替えます。

[実行例]

```
#ewssyserror -p  
NORMAL
```

```
#ewssyserror -p off  
#ewssyserror -p  
Monitoring off
```

```
#ewssyserror -m  
ERROR 09/02/14 16:23:02
```

```
#ewssyserror -m reset  
#ewssyserror -m  
NORMAL
```

```
#ewssyserror -b
NORMAL
```

```
#ewssyserror -b off
#ewssyserror -b
Monitoring off
```

(7) ewssinfo

[名前]

ewssinfo システム情報の表示

[書式]

ewssinfo {-b | -t [reset]}

[説明]

指定したオプションに従って、BIOS 版数情報または装置の総通電時間を表示します。装置の総通電時間は、0 時間にリセットできます。

[オプション]

- b BIOS カスタマイズコードと版数を表示します。
- t パラメーターなしで実行した場合、装置の総通電時間を 0.1 時間単位で表示します。パラメーター reset を指定した場合、総通電時間の記録をリセットします。

[実行例]

```
# ewssinfo -b
PFU Customize Code: TN
BIOS Version: 1.04
```

```
# ewssinfo -t
17.5
```

```
# ewssinfo -t reset
# ewssinfo -t
0.0
```

(8) ewslog

[名前]

ewslog BIOS イベントログと EmbedWare/SysMon イベントログの表示

[書式]

```
ewslog {-b [-s|-r ソート種別] [-f FilePath] |  
-e [-s|-r ソート種別] [-f FilePath]}
```

[説明]

指定したオプションに従って、BIOS イベントログ、EmbedWare/SysMon イベントログを表示します。情報のソート表示、およびファイルへの保存ができます。

[オプション]

- b BIOS イベントログ (biosevent.log) を表示します。
- e EmbedWare/SysMon イベントログ (sysmon.log) を表示します。
- s 表示内容を以下のソート種別に従ってソートします。
 - time 時刻が古い順にソート表示。
 - type メッセージのアルファベット順にソート表示。
- r 表示内容を以下のソート種別に従ってソートします。
 - time 時刻が新しい順にソート表示。
 - type メッセージのアルファベット逆順にソート表示。
- f 情報を FilePath で指定したファイルに保存します。FilePath で指定したファイルがすでに存在する場合は確認メッセージが表示されます (y or Y 入力で上書きを実行)。FilePath で指定したパスがディレクトリ名の場合はエラーになります。



ソートをしない場合は、時刻が古い順での出力となります。

注意

[実行例]

```
# ewslog -b
```

```
16/07/19 14:31:03 : POST Error. : Keyboard Not Functional
```

```
# ewslog -e -s type
```

```
16/06/15 15:14:53 12V voltage lower threshold breached: 8.0311[V]  
16/06/15 12:45:05 12V voltage lower threshold breached: 8.0311[V]  
16/06/15 11:46:34 12V voltage lower threshold breached: 8.0311[V]  
16/06/15 11:24:06 3.3V voltage lower threshold breached: 1.3758[V]  
16/06/15 11:20:22 3.3VL voltage lower threshold breached: 0.0343[V]  
16/06/15 14:11:23 5V(MEM) voltage lower threshold breached: 3.1356[V]  
16/06/15 15:15:13 5V(MEM) voltage lower threshold breached: 3.3430[V]  
16/06/15 11:20:19 5VL voltage lower threshold breached: 0.0519[V]  
16/06/15 11:45:43 CPU core voltage upper threshold breached: 1.8967[V]  
16/06/15 11:45:07 CPU core voltage upper threshold breached: 1.8967[V]
```

```
16/06/15 13:50:04 CPU temperature upper threshold breached: 90[degrees C]
16/06/15 11:20:22 MB temperature upper threshold breached: 35[degrees C]
16/06/15 15:16:32 MB temperature upper threshold breached: 31[degrees C]
16/06/15 11:19:39 MB temperature upper threshold breached: 30[degrees C]
16/06/15 15:14:54 SYSFAN fan speed lower threshold breached: 2888[rpm]
```

(9) ewsperform

[名前]

ewsperform パフォーマンス情報の表示

3

[書式]

ewsperform [-c on | off]

[説明]

オプションなしでこのコマンドを実行した場合、CPU スロットごとの実周波数 (MHz) を表示します。引数に -c オプションを指定した場合、監視機能のオン／オフを切り替えます。

[オプション]

-c CPU パフォーマンス機能のオン／オフを切り替えます。

[実行例]

#ewsperform

CPU

SLOT0

CPU Clock: 800MHz

SLOT1

CPU Clock: 900MHz

#ewsperform -c off

#ewsperform

CPU

Monitoring off

EmbeddedWare/SysMon Entry を利用する

(10) ewsctl

[名前]

ewsctl 状態監視に関する各種設定

[書式]

```
ewsctl [ [-a on [value] | off][-t on | off][-b [value]]  
[-e on [value] | off][-w [on [value] [action] | off]] ]
```

[説明]

オプションなしでこのコマンドを実行した場合、以下の項目について現在の設定を表示します。

- Trace

温度、電圧、ファン回転数監視機能で、測定値の履歴をトレースファイルに残すかどうかの設定を on/off で表示します。初期値は off です。

- Alert

すべての監視項目について、異常検出時に、ビープ音を鳴らすかどうか、状態変化時にシリログにメッセージを格納するかどうかを設定します。この機能が無効な場合は off、有効な場合はビープ音を鳴らす時間を表示します。初期値は off です。

- Log Generation (EmbedWare/SysMon Log、BIOS Event Log)

EmbedWare/SysMon イベントログ、BIOS イベントログの保存世代数を表示します。ログ保存世代数の初期値は 1 です。

- 監視設定 (EmbedWare/SysMon Log)

EmbedWare/SysMon イベントログにメッセージを出力するかどうかの設定を on/off で表示します。初期値は on です。

- WDT

ウォッチドッグタイマの設定を表示します。この機能が無効な場合は off、有効な場合はタイムアウト時間およびタイムアウト発生時のアクションを表示します。初期値は off です。

[オプション]

-a ビープ音とシリログ出力の設定のオン／オフを指定します。オンの場合、ビープ音を鳴らす時間を 0～60 秒の範囲で指定します。シリログにもメッセージが格納されます。ビープ音の時間を省略した場合は 3 秒に設定されます。



ビープ音を鳴らす時間が 0 秒で異常検出した場合は、システムが終了するまでビープ音を鳴らし続けます。

-t 温度、電圧、ファン回転数測定値のトレース機能のオン／オフを指定します。

-b BIOS イベントログファイルの保存世代数を 1～10 世代の範囲で指定します。value を省略した場合は 1 世代となります。

-e EmbedWare/SysMon イベントログのメッセージ出力のオン／オフおよびファイルの保存世代数を指定します。オンの場合、ファイル保存世代数を 1～10 世代の範囲で指定します。value を省略した場合は 1 世代となります。

-w 引数なしでこのオプションを指定した場合、ウォッチドッグタイマのタイムアウト発生時のアクション一覧を表示します。表示されるアクションは、次のとおりです。

- hardreset : 装置再起動を実施します。
- noaction : 何も実施しません。
- powerdown : 装置のシャットダウンを実施します。
- powercycle : 装置のシャットダウンを実施したあとに装置起動を実施します。



装置でサポートしているアクションだけが表示されます。

引数に on/off を指定した場合、ウォッチドッグタイマのオン／オフを設定できます。

引数に on を指定した場合、タイムアウト時間の秒数、タイムアウト発生時のアクションを指定できます。

value を省略すると最大値が設定されます（設定可能な値の範囲秒数は機種によって異なります。ewsctl をオプションなしで実行することにより確認できます）。

action を省略すると hardreset が設定されます（設定可能なアクションは機種によって異なります。ewsctl -w を実行することにより確認できます）。



CPU 使用率が継続して高負荷になるような操作およびOS 停止要因となる操作を実施する場合は、ウォッチドックタイマの設定をオフにするか、タイムアウト値を適切な値に変更してください。EmbeddedWare/SysMon サービスプログラムがウォッチドッグタイマにアクセスできず、装置がシャットダウンまたは再起動する可能性があります。

[実行例]

```
# ewsctl
```

Options

Trace	off
Alert	off
EmbeddedWare/SysMon Log	on
Log Generation(EmbeddedWare/SysMon Log)	1 (1-10)
Log Generation(BIOS Event Log)	1 (1-10)
WDT	off (10-600)

```
# ewsctl -w
```

WDT Action

- hardreset
- noaction
- powerdown
- powercycle

```
# ewsctl -t on -a on 10 -e off -b 5 -w on 60 hardreset
```

```
# ewsctl
```

Options

Trace	on
Alert	10 sec
EmbedWare/SysMon Log	off
Log Generation(EmbedWare/SysMon Log)	1 (1-10)
Log Generation(BIOS Event Log)	5 (1-10)
WDT	60 sec (10-600)
WDT Action	hardreset

(11) **ewsbmcsel**

[名前]

ewsbmcsel BMC SEL の表示、設定

[書式]

ewsbmcsel [[-s] | [-o [on | off]] | [-c] | [-f FilePath]]

[説明]

オプションなしでこのコマンドを実行した場合、採取済みの BMC SEL を表示します。採取していない場合、BMC SEL は表示されません。オプションの指定によって BMC SEL の設定の変更、BMC SEL の採取・保存ができます。

[オプション]

-s BMC SEL の採取を開始します。



採取済みのデータは、再び BMC SEL を採取することで上書きされます。

-o 引数なしでこのオプションを指定した場合、BMC SEL の記録サイズが最大値に達したときの、記録の上書き設定の状態を表示します。引数に on/off を指定した場合、BMC SEL の記録サイズが最大値に達したときに記録を上書きするかを、on/off で指定できます。初期値は on です。



記録を上書きしない運用にした場合、記録サイズが最大値に達しても、BMC SEL に新たに記録することができません。

-c BMC 内の SEL を削除します。



BMC 内の SEL の削除は途中で停止できません。また、採取済みの BMC SEL はこの操作で削除されません。

-f FilePath で指定したファイルに、採取済みの BMC SEL を保存します。FilePath で指定したファイルがすでに存在する場合は、確認メッセージが表示されます (y or Y 入力で上書きを実行)。FilePath で指定したパスがディレクトリ名の場合は、エラーになります。

[実行例]

```
# ewsbmcsel -o
```

Setting

Overwrite on

```
# ewsbmcsel -o off
```

```
# ewsbmcsel -o
```

Setting

Overwrite off

```
# ewsbmcsel -s
```

Collecting 100%

```
# ewsbmcsel
```

ID	Timestamp	Typ	Gene	SensorType	SensorName	EvnetDir	Description
0001	17/03/20 16:34:20	SYS	BMC	Temperature	MB	Assert	Lower Critical...
0002	17/03/20 16:34:20	SYS	BMC	Voltage	3.3V	Assert	Upper Critical...

CollectionTime: 2017/03/22 11:04:50

```
# ewsbmcsel -c
```

```
# ewsbmcsel -s
```

Collecting 100%

```
# ewsbmcsel
```

CollectionTime: 2017/03/23 22:30:50

(12) ewssvlog

[名前]

ewssvlog RAID カード（タイプ 2）の保守情報採取

[書式]

ewssvlog [{-o | -output-dir} <OutputDirPath>]

ewssvlog {-h | -help}

[説明]

オプションなしでこのコマンドを実行した場合、採取した RAID カード（タイプ 2）の保守情報がカレントディレクトリに保存されます。オプションの指定によって採取した RAID カード（タイプ 2）の保守情報保存先の指定および使用方法の参照ができます。

[オプション]

-o | -output-dir 採取した RAID カード（タイプ 2）の保守情報を OutputDirPath で指定したディレクトリに保存します。



採取した RAID カード（タイプ 2）の保守情報の保存先をカレントディレクトリにしたい場合はオプションなしでコマンドを実行してください。

-h|help コマンドの使用方法を表示します。

[実行例]

```
# ewssvlog -o /home/pfu
Device /dev/sda(XXXXXX) is ....
Card Maintenance Info print start.
Card Maintenance Info print OK.
Device1 Maintenance Info print start.
Device1 Maintenance Info print OK.
Device2 Maintenance Info print start.
Device2 Maintenance Info print OK.
Card Error Log dump start.
Card Error Log dump OK.
Device1 Error Log dump start.
Device1 Error Log dump OK.
Device2 Error Log dump start.
Device2 Error Log dump OK.
Complete->"/home/pfu/2017-12-20_22-36-33_SV_RAID_MaintenanceInfo_dev_sda".
```

第 2 部 EmbedWare/ SysMon SDK

第1章 概要



この章では、EmbedWare/SysMon SDK の概要について説明します。

1.1 EmbedWare/SysMon SDK とは

EmbedWare/SysMon SDK では、EmbedWare/SysMon Entry の機能をアプリケーションから利用するための API を提供しています。

この API を利用することで、アプリケーションに容易にハードウェア監視機能を追加できます。

1.2 機能

EmbedWare/SysMon SDK には、以下の機能があります。

機能	説明
コンピューターの状態の取得	コンピューターの温度、電圧値、ファン回転数の異常の有無、現在の値、過去最高値・過去最低値、異常判定の際の上限および下限値を取得します。
S.M.A.R.T. 監視	内蔵ディスクデバイスの S.M.A.R.T. 機能による異常検出の有無、および S.M.A.R.T. アトリビュートデータを取得します。
RAID 監視	RAID カード（タイプ 1）、RAID カード（タイプ 2）の状態を取得します。RAID カード（タイプ 2）は、接続ディスクの S.M.A.R.T. アトリビュートデータおよびモデル名、シリアル番号を取得できます。
システムエラーに関する情報の取得	PCI バス、メモリ、BMC エラーの有無、エラー発生時の BIOS イベントログデータを取得します。
パフォーマンス情報の取得	CPU の実周波数情報を取得します。
動作設定の取得	各種監視機能の有効・無効、異常通知設定を確認します。温度、電圧値、ファン監視で異常検出時のシャットダウン設定を確認します。
ウォッチドッグタイマの操作	アプリケーションから直接ウォッチドッグタイマを制御します。
GPIO (General Purpose Input / Output) の操作	システム オン モジュールの GPIO を操作します。



RAID 監視で監視対象としている RAID カードは、2 種類存在します。

備考

- RAID カード（タイプ 1）は PCIe インターフェイスカードタイプです。
- RAID カード（タイプ 2）は SATA インターフェイスカードタイプです。

機種によって搭載される RAID カードは異なります。

搭載 RAID カードが不明な場合は、担当営業にご相談ください。



- GPIO操作は、システムオンモジュールで使用可能です。エンベデッドコンピューターでは使用できません。
- 監視情報を取得するAPIは、過渡的な情報を取得する場合があります。例えば、ESM_RAIDInfo()関数で物理ディスクステータスがRebuildingを示しているにもかかわらず、論理ボリュームステータスがDegradedを示すなどの場合があります。過渡的な情報が得られた場合は5秒後に再度APIを実行してください。

1.3 アプリケーション開発環境

(1) インストール環境

EmbedWare/SysMon SDKがインストール可能な環境を以下に示します。

ディストリビューション : Red Hat Enterprise Linux
Ubuntu

ディスク資源 : 約0.2MB

(2) 必要ファイル

EmbedWare/SysMon SDKを利用するアプリケーションの開発には、EmbedWare/SysMon SDKをインストールしてください。必要に応じて以下のファイルをコピーしてください。

- /usr/include/esm_api.h
- /opt/EmbedWare_SysMon/Entry/lib/libesm_api.so



- 対応ディストリビューションおよびバージョンについての詳細は、ハードウェアのカタログまたはマニュアルでご確認ください。
- 上記以外の環境をご使用のお客様は、担当営業にご相談ください。

1.4 アプリケーション実行環境

EmbedWare/SysMon SDKで開発したアプリケーションを実行するためには、EmbedWare/SysMon SDKとEmbedWare/SysMon Entryの両方をインストールする必要があります。

第2章 インストールとアンインストール

この章では、EmbedWare/SysMon SDK のインストールとアンインストールの方法について説明します。

2.1 インストール

◇ Red Hat Enterprise Linux でインストールする

Red Hat Enterprise Linux でインストールする方法を以下に示します。

1 root 権限でログインします。

```
$ su -
Password: ***** ←root のパスワードを入力
```

2 EmbedWare/SysMon SDK のインストールプログラムがあるディレクトリに移動して、以下のコマンドでファイルの有無を確認します。

```
# ls
install-ews-sdk.sh
```

3 EmbedWare/SysMon SDK のインストールを開始します。

```
# bash install-ews-sdk.sh
```

◇ Ubuntu でインストールする

Ubuntu でインストールする方法を以下に示します。

1 EmbedWare/SysMon SDK のインストールプログラムがあるディレクトリに移動して、以下のコマンドでファイルの有無を確認します。

```
$ ls
install-ews-sdk.sh
```

2 EmbedWare/SysMon SDK のインストールを開始します。

```
$ sudo bash install-ews-sdk.sh
```



インストールに失敗した場合は、製品格納ディレクトリおよびその配下にファイルが残る場合があります。再度インストールを行い、インストールを完了させてください。

2.2 アンインストール

◇ Red Hat Enterprise Linux でアンインストールする

Red Hat Enterprise Linux でアンインストールする方法を以下に示します。

- 1 root 権限でログインします。

```
$ su -  
Password: ***** ←root のパスワードを入力
```

- 2 パッケージの問い合わせをします。以下のコマンドを使用して、EmbedWare/SysMon SDK がインストールされているかどうかを確認します。

```
# rpm -q ews-sdk  
ews-sdk-7.x-x.x
```

- 3 EmbedWare/SysMon SDK をアンインストールします。

```
# rpm -e ews-sdk
```

◇ Ubuntu でアンインストールする

Ubuntu でアンインストールする方法を以下に示します。

- 1 パッケージの問い合わせをします。以下のコマンドを使用して、EmbedWare/SysMon SDK がインストールされているかどうかを確認します。

```
$ dpkg -l ews-sdk  
||/ Name      Version       Description  
+++-==  
ii  ews-sdk    7.x.x.x     xxxx
```

- 2 EmbedWare/SysMon SDK をアンインストールします。

```
$ sudo dpkg -r ews-sdk
```

2.3 版数確認

◇ Red Hat Enterprise Linux で版数を確認する

以下のコマンドで版数を確認します。

```
# rpm -q ews-sdk  
ews-sdk-7.x-x.x
```

◇ Ubuntu で版数を確認する

以下のコマンドで版数を確認します。

```
$ dpkg -l ews-sdk  
||/ Name          Version       Description  
====-  
ii  ews-sdk      7.x.x.x     xxxx
```


第3章 API 仕様



この章では、EmbedWare/SysMon SDK を利用するにあたって、必要な API 仕様について説明します。

3.1 ESM_Open() 関数

機能説明

EmbedWare/SysMon Entry へのアクセスを開始します。

呼び出し形式

```
#include "esm_api.h"  
int ESM_Open(void);
```

パラメーター

なし

復帰値

- | | |
|---------------|---|
| 0 以上 | : 正常終了。EmbedWare/SysMon Entry へのアクセス開始を行うための識別子を返します。 |
| ESM_ERROR(-1) | : システムエラーが発生。 |



EmbedWare/SysMon Entry へのアクセスは、root (システム管理者) 権限が必要です。

使用例

```
#include "esm_api.h"  
  
int descriptor; /* EmbedWare/SysMon アクセス識別子 */  
descriptor = ESM_Open(); /* EmbedWare/SysMon アクセス識別子獲得 */  
if ( descriptor < 0 ) {  
    /* エラー処理 */  
}  
/* EmbedWare/SysMon アクセス要求開始 */  
:
```

3.2 ESM_Close() 関数

機能説明

EmbedWare/SysMon Entryへのアクセスを終了します。

ESM_Open()と対で必ず実行してください。

呼び出し形式

```
#include "esm_api.h"  
int ESM_Close(int descriptor);
```

パラメーター

int descriptor : ESM_Open()で獲得した識別子を指定します。

復帰値

ESM_SUCCESS(0) : 正常終了。

ESM_ERROR(-1) : システムエラーが発生。

使用例

```
#include "esm_api.h"  
int descriptor; /* EmbedWare/SysMon アクセス識別子 */  
descriptor = ESM_Open(); /* EmbedWare/SysMon アクセス識別子獲得 */  
if( descriptor < 0 ) {  
    /* エラー処理 */  
}  
/* EmbedWare/SysMon アクセス要求開始 */  
:  
:  
/* EmbedWare/SysMon アクセス要求終了 */  
ESM_Close( descriptor );  
:
```

3.3 ESM_Get() 関数

機能説明

指定された監視項目の現在の状態を獲得します。

呼び出し形式

```
#include "esm_api.h"
int ESM_Get(int descriptor, char *Key1, char *Key2, char *Rtn_Value);
```

パラメーター

int descriptor	: ESM_Open() で獲得した識別子を指定します。
char *Key1	: 監視または設定項目を示すキーワードへのポインターを指定します。指定できる文字列は以下のとおりです。
TEMP1 ~ TEMP48	: 個々の温度センサーに関する情報を取得します。
 TEMP1 ~ TEMP48	TEMP1 ~ TEMP48 に対応する温度センサーは、EmbedWare/SysMon Entry で確認してください。番号は表示順に対応しています。

VOLTAGE1 ~ VOLTAGE48 : 個々の電圧センサーに関する情報を取得します。

 VOLTAGE1 ~ VOLTAGE48 に対応する電圧センサーは、EmbedWare/SysMon Entry で確認してください。番号は表示順に対応しています。

FAN1 ~ FAN48 : 個々のファンに関する情報を取得します。

 FAN1 ~ FAN48 に対応するファンは、EmbedWare/SysMon Entry で確認してください。番号は表示順に対応しています。

SLOT0 ~ SLOT7 : S.M.A.R.T. 監視に関する情報を取得します。

-  • SLOT0 ~ SLOT7 に対応する S.M.A.R.T. 監視情報は、EmbedWare/SysMon Entry で確認してください。
- EmbedWare/SysMon Ver7.0 以前に使用していた DISK1 ~ DISK8 も指定可能です。DISK1 ~ DISK8 の番号は、SLOT の番号に 1 を加えた値です。

UPTIME : 総通電時間を取得します。

TEMP : 温度監視全般に関する設定を取得します。

VOLTAGE : 電圧監視全般に関する設定を取得します。

FAN : ファン監視全般に関する設定を取得します。

RAID : チップセット RAID および RAID カード（タイプ 1）監視の共通設定を取得します。

 共通設定は、チップセット RAID および RAID カード（タイプ 1）監視の各設定の論理和になります。

CHIPSETRAID : チップセット RAID 監視の設定を取得します。

RAIDCARD : RAID カード（タイプ 1）監視の設定を取得します。

SVRAID : RAID カード（タイプ 2）監視の設定を取得します。

PCI : BIOS イベントログによる PCI エラー監視の設定を取得します。

MEMORY : BIOS イベントログによるメモリエラー監視の設定を取得します。

BMC : BIOS イベントログによる BMC エラー監視の設定を取得します。

ALL : すべての監視項目に共通した情報を取得します。

char *Key2	: 取得したい情報種別を示すキーワードへのポインターを指定します。指定できる文字列は以下のとおりです。実際に指定できる値は、Key1 の内容に依存します。※「補足説明」参照
Name	: センサー名です。
Value	: 現在の値です。
Status	: 現在の状態です。
Threshold	: 現在設定されているしきい値です。
Worst	: 過去最高値・過去最低値の記録と発生日時です。
Action	: 監視の有効・無効設定または異常発生時の動作設定です。
Log	: 異常通知に関する設定です。
Hddinfo	: ハードディスクのモデル名、シリアル番号です。
char *Rtn_Value	: 指定されたキーワードに対応した情報を格納する領域へのポインターを指定します。アプリケーションは最低 64byte (終端ヌル文字を含む) の格納領域を用意する必要があります。

補足説明

Key1、Key2 で指定可能なパラメーターの組み合わせと、Rtn_Value に返されるデータを以下に示します。

Key1	Key2	Rtn_Value (*1)	Rtn_Value の説明
TEMP1 ~ TEMP48	Name	(センサー名)	EmbedWare/SysMon Entry の ewsmondisp コマンドで表示される Name と同じ
	Value	(整数)	現在値 (*2)
	Status	NORMAL	正常
		ERROR	異常発生
	Threshold	(整数) , (整数)	上限しきい値、下限しきい値 下限しきい値が存在しない場合、 右側の整数は省略されます。 例) 100,
	Action	ON	監視有効
		OFF	監視無効
	Worst	(整数) , (整数) , (YY/ MM/DD hh:mm:ss) , (YY/MM/DD hh:mm:ss)	過去最高温度、最低温度、最高温 度の記録日時、最低温度の記録日 時 (*2)

Key1	Key2	Rtn_Value (*1)	Rtn_Value の説明
VOLTAGE1 ~ VOLTAGE48	Name	(センサー名)	EmbedWare/SysMon Entry の ewsmondisp コマンドで表示される Name と同じ
	Value	(小数)	現在値 (出力形式 : 小数第 4 位まで表示) (*2)
	Status	NORMAL	正常
		ERROR	異常発生
	Threshold	(小数) , (小数)	上限しきい値、下限しきい値 (出力形式 : 小数第 4 位まで表示)
	Action	ON	監視有効
		OFF	監視無効
FAN1 ~ FAN48	Worst	(小数) , (小数) , (YY/MM/DD hh:mm:ss) , (YY/MM/DD hh:mm:ss)	過去最高電圧、最低電圧 (出力形式 : 小数第 4 位まで表示)、最高電圧の記録日時、最低電圧の記録日時 (*2)
	Name	(センサー名)	EmbedWare/SysMon Entry の ewsmondisp コマンドで表示される Name と同じ
	Value	(整数)	現在値 (*2)
	Status	NORMAL	正常
		ERROR	異常発生
	Threshold	(整数) , (整数)	上限しきい値、下限しきい値 上限しきい値が存在しない場合、左側の整数は省略されます。 例) ,1000
	Action	ON	監視有効
		OFF	監視無効
SLOT0 ~ SLOT7 DISK1 ~ DISK8	Worst	(整数) , (整数) , (YY/MM/DD hh:mm:ss) , (YY/MM/DD hh:mm:ss)	過去最高回転数、最低回転数、最高回転数の記録日時、最低回転数の記録日時 (*2)
	Name	(監視項目名)	EmbedWare/SysMon Entry の ewssmart コマンドで表示される DispName と同じ
	Status	OK	正常
		WARNING	異常発生
		UNKNOWN	S.M.A.R.T. ステータスが取得できない

Key1	Key2	Rtn_Value (*1)	Rtn_Value の説明
SLOT0 ~ SLOT7 DISK1 ~ DISK8	Hddinfo	(Model) , (SerialNumber)	ハードディスクのモデル名、シリアル番号
	Action	ON	監視有効
		OFF	監視無効
UPTIME	Value	(小数)	総通電時間（単位：時間、出力形式：小数第一位まで表示）
TEMP	Action	ON, (整数 : 0 ~ 60)	温度異常発生時シャットダウンする
		OFF	温度異常発生時シャットダウンしない
VOLTAGE	Action	ON, (整数 : 0 ~ 60)	電圧異常発生時シャットダウンする
		OFF	電圧異常発生時シャットダウンしない
FAN	Action	ON, (整数 : 0 ~ 60)	ファン異常発生時シャットダウンする
		OFF	ファン異常発生時シャットダウンしない
RAID (チップセット RAID および RAID カード (タイプ1) 監視の共通設定)	Action	ON	監視有効
		OFF	監視無効
CHIPSETRAID	Action	ON	監視有効
		OFF	監視無効
RAIDCARD	Action	ON	監視有効
		OFF	監視無効
SVRAID	Action	ON	監視有効
		OFF	監視無効
PCI (PCI エラー監視)	Action	ON	監視有効
		OFF	監視無効
MEMORY (メモリエラー監視)	Action	ON	監視有効
		OFF	監視無効
BMC (BMC エラー監視)	Action	ON	監視有効
		OFF	監視無効
ALL	Action または Log	ON, (整数 : 0 ~ 60)	異常発生時、ビープ音を指定秒数鳴らし、システムログにメッセージを格納します。

Key1	Key2	Rtn_Value (*1)	Rtn_Value の説明
ALL	Action または Log	OFF	異常発生時、ビープ音を鳴らさず、 シスログにもメッセージを格納しません。

*1 : 括弧付きのものは、値の内容を示します。括弧なしのものは、その値がそのまま格納されることを示します。

*2 : センサー非搭載時または監視準備中は、値が空になります。センサー搭載しているにもかかわらず値が空の場合は、監視準備中の可能性がありますので再取得してください。

復帰値

- ESM_SUCCESS(0) : 正常終了。
- ESM_ERROR(-1) : システムエラーが発生。
- ESM_PRMERR(-2) : 指定パラメーターに誤りがある。
- ESM_NODATA1(-9) : Key1 に誤りがある。
- ESM_NODATA2(-10) : Key1、Key2 の組み合わせに誤りがある。

使用例

```
#include "esm_api.h"
int rtn;
char rtnstr[64];
int descriptor; /* EmbedWare/SysMon アクセス識別子 */
descriptor = ESM_Open(); /* EmbedWare/SysMon アクセス識別子獲得 */
if ( hesm == (HANDLE)ESM_ERROR ) {
    /* エラー処理 */
}
/* EmbedWare/SysMon アクセス要求開始 */
:
memset( rtnstr, 0, 64 ); /* 獲得域の初期化 */
rtn = ESM_Get( descriptor, "TEMP1", "Value", rtnstr ); /* TEMP1 の現在値獲得 */
if ( rtn < 0 ) {
    /* エラー処理 */
}
/* 空文字（センサー非搭載または監視準備中）でない場合に表示する */
if ( strcmp( "", rtnstr ) != 0 ) {
    printf( "TEMP1 = %s°C", rtnstr );
}
:
/* 次の監視項目へ */
:
/* EmbedWare/SysMon アクセス要求終了 */
ESM_Close( descriptor );
```

上記使用例をコンソールアプリケーションとして作成し実行した場合、以下のように出力されます。

TEMP1 = 40°C

3.4 ESM_GetSmart() 関数

機能説明

指定したディスクの S.M.A.R.T. ステータスと S.M.A.R.T. アトリビュートデータを取得します。本関数には拡張関数が用意されており、「3.16 ESM_GetSmartEx() 関数」(60 ページ) の使用を推奨しています。

呼び出し形式

```
#include "esm_api.h"
int ESM_GetSmart(int descriptor, char *Key1, smart_attributes_t *Get_Value);
```

パラメーター

int descriptor	: ESM_Open() で獲得した識別子を指定します。
char *Key1	: 監視または設定項目を示すキーワードへのポインターを指定します。指定できる文字列は以下のとおりです。
DISK1 ~ DISK8	: 指定されたディスクに関する情報を取得します。
smart_attributes_t *Get_Value	: 情報を格納する領域へのポインター。以下に示す smart_attributes_t 構造体へのポインターを指定します。

smart_attr_t, smart_attributes_t 構造体

```
typedef struct smart_attr {
    unsigned char id;
    unsigned char flags[2];
    unsigned char value;
    unsigned char worst;
    unsigned char threshold;
    unsigned char raw[6];
} smart_attr_t;
```

```
typedef struct smart_attributes {
    int status;
    smart_attr_t attribute[30];
} smart_attributes_t;
```

補足説明

smart_attributes_t 構造体に格納される status の意味は、以下のとおりです。

0	: S.M.A.R.T. アトリビュートデータが取得できません（この場合 attribute の格納は行われません）
1	: S.M.A.R.T. ステータスは正常です
2	: S.M.A.R.T. ステータスの異常が検出されています

復帰値

ESM_SUCCESS(0)	: 正常終了。
ESM_ERROR(-1)	: システムエラーが発生。
ESM_PRMERR(-2)	: 指定パラメーターに誤りがある。

3.5 ESM_RAIDInfo() 関数

機能説明

RAID カード（タイプ 1）のステータス情報を取得します。

呼び出し形式

```
#include "esm_api.h"
int ESM_RAIDInfo(int descriptor, int Controller, int Port, unsigned int *status, char *raidlevel);
```

パラメーター

int descriptor	: ESM_Open() で獲得した識別子を指定します。
int Controller	: RAID コントローラーの種類を指定します。
1	: RAID カード（タイプ 1）の情報を取得します。
int Port	: RAID カード構成情報を指定します。RAID カード構成情報はデバイス番号、コネクター番号、コントローラー番号の論理和で表現されます。
0x000000XX : XX	: デバイス番号。
0x0000XX00 : XX	: コネクター番号。
0x00XX0000 : XX	: コントローラー番号。ここで指定するコントローラー番号の開始番号は 00 です。EmbedWare/SysMon Entry の表示より 1 小さいのでご注意ください。
unsigned int *status	: ステータス情報を格納する領域へのポインターを指定します。ステータス情報は、以下の物理ディスクステータス、物理ディスク S.M.A.R.T. ステータス、論理ボリュームステータス、構成情報の論理和で表現されます。
物理ディスクステータス	
None : 0x00	: パラメーター Port で指定したディスクが存在しない
Online : 0x01	: RAID 構成されているディスク
HotSpare : 0x02	: ホットスペアディスク
Failed : 0x04	: ディスクの異常が検出された
Rebuilding : 0x08	: リビルド中
NoArray : 0x10	: 論理ボリュームに所属していないディスクが存在する
物理ディスク S.M.A.R.T. ステータス	
NORMAL : 0x00	: ディスクの S.M.A.R.T. 異常が検出されなかった（ディスクの S.M.A.R.T. 状態が正常または不明）
ERROR : 0x80	: ディスクの S.M.A.R.T. 異常が検出された
論理ボリュームステータス	
None/NoArray : 0x0000	: 物理ディスクが RAID 構築されていないか、ディスクが存在しない
OK : 0x0100	: 論理ボリュームが正常
Degraded : 0x0200	: 論理ボリュームの冗長性が失われている
Rebuilding : 0x0400	: 論理ボリュームがリビルド中
Failed : 0x0800	: 論理ボリュームが異常

構成情報

0xXXXX0000	: XXXX : 論理ボリューム番号
char *raidlevel	: RAID レベル情報を格納する領域へのポインターを指定します。 この情報が不要な場合は NULL を指定します。 指定した領域に RAID レベル情報を "0"、"1"、"5"、"10" などの文字列で格納します。領域のサイズは終端ヌル文字を含め最低 8byte 準備する必要があります。指定した場所が RAID 構成ではない場合、長さ 0 の文字列が格納されます。

復帰値

ESM_SUCCESS(0)	: 正常終了。
ESM_ERROR(-1)	: システムエラーが発生。
ESM_PRMERR(-2)	: 指定パラメーターに誤りがある。

留意事項

監視が有効ではない場合（装置が RAID 構成ではない、または、監視を無効に設定している）でもこの関数がエラーとなることはありませんが、すべてのステータス情報は 0 となります。

3.6 ESM_SvRAIDInfo() 関数

機能説明

RAID カード（タイプ 2）のステータス情報を取得します。

呼び出し形式

```
#include "esm_api.h"
int ESM_SvRAIDInfo(int descriptor, int Array, int Drive, unsigned int *status, char *raidlevel);
```

パラメーター

int descriptor	: ESM_Open() で獲得した識別子を指定します。
int Array	: 論理ボリュームの番号を指定します。0～3 の値が指定可能です。
int Drive	: ディスク番号を指定します。0～1 の値が指定可能です。
unsigned int *status	: ステータス情報を格納する領域へのポインターを指定します。 ステータス情報は、以下の物理ディスクステータス、物理ディスク S.M.A.R.T. ステータス、論理ボリュームステータス、構成情報の論理和で表現されます。

物理ディスクステータス

None : 0x00	: RAID カードが検出できなかった
Online : 0x01	: RAID 構成されているディスク
Failed : 0x02	: ディスクの異常が検出された
Rebuilding : 0x04	: リビルド中

物理ディスク S.M.A.R.T. ステータス

NORMAL : 0x00	: ディスクの S.M.A.R.T. 異常が検出されなかった（ディスクの S.M.A.R.T. 状態が正常または不明）
ERROR : 0x80	: ディスクの S.M.A.R.T. 異常が検出された

論理ボリュームステータス

None : 0x0000	: 物理ディスクが RAID 構築されていないか、ディスクが存在しない
OK : 0x0100	: 論理ボリュームが正常
Degraded : 0x0200	: 論理ボリュームの冗長性が失われている
Rebuilding : 0x0400	: 論理ボリュームがリビルド中
Failed : 0x0800	: 論理ボリュームが異常

構成情報

0xXXXX0000	: XXXX : 論理ボリューム番号
char *raidlevel	: RAID レベル情報を格納する領域へのポインターを指定します。 この情報が不要な場合は NULL を指定します。指定した領域に RAID レベル情報を "0"、"1"、"5"、"10" などの文字列で格納します。領域のサイズは終端ヌル文字を含め最低 8 バイト準備する必要があります。

復帰値

ESM_SUCCESS(0)	: 正常終了。
ESM_ERROR(-1)	: システムエラーが発生。
ESM_PRMERR(-2)	: 指定パラメーターに誤りがある。

留意事項

監視が有効ではない場合（RAID カードが存在しない、または、監視を無効に設定している）でもこの関数がエラーとなることはありませんが、すべてのステータス情報は 0 となります。

3.7 ESM_QueryError() 関数

機能説明

PCI エラー、メモリエラー、BMC エラー情報を取得します。

呼び出し形式

```
#include "esm_api.h"  
int ESM_QueryError(int descriptor, char *Key1, int *status, char *detail_info);
```

パラメーター

int descriptor	: ESM_Open() で獲得した識別子を指定します。
char *Key1	: 監視または設定項目を示すキーワードへのポインターを指定します。指定できる文字列は以下のとおりです。 PCI : PCI エラー情報を取得 MEMORY : メモリエラー情報を取得 BMC : BMC エラー情報を取得
int *status	: ステータス情報を格納する領域へのポインターを指定します。格納される値は以下のとおりです。 0 : 異常は検出されていません 1 : 異常が検出されました
char *detail_info	: 詳細情報を格納する領域へのポインターを指定します。上記 Status に 1 が返された場合に、この領域に値が格納されます。異常検出時の BIOS イベントログのバイナリデータを格納します。最低 256byte の領域を準備する必要があります。この情報が不要な場合は NULL を指定します。

詳細情報の格納形式は以下のとおりです。

Byte1	: イベントログタイプ
Byte2	: 有効レコード長 Byte1～8 を含むログ全体の長さを示す 7bit の値が格納されます (最上位ビットはリザーブ)。
Byte3	: タイムスタンプ (年)
Byte4	: タイムスタンプ (月)
Byte5	: タイムスタンプ (日)
Byte6	: タイムスタンプ (時)
Byte7	: タイムスタンプ (分)
Byte8	: タイムスタンプ (秒)
Byte9～有効レコード長	: ログ固有のデータ領域

復帰値

- ESM_SUCCESS(0) : 正常終了。
- ESM_ERROR(-1) : システムエラーが発生。
- ESM_PRMERR(-2) : 指定パラメーターに誤りがある。

3.8 ESM_GetWDTTimeOut() 関数

機能説明

ウォッチドッグタイマのタイムアウト時間に設定可能な最大値（秒）、最小値（秒）を取得します。

呼び出し形式

```
#include "esm_api.h"
int ESM_GetWDTTimeOut(int descriptor, int *Max_Time, int *Min_Time);
```

パラメーター

int descriptor	: ESM_Open() で獲得した識別子を指定します。
int *Max_Time	: ウォッチドッグタイマのタイムアウト時間に設定可能な最大値（秒）を格納する領域へのポインターを指定します。
int *Min_Time	: ウォッチドッグタイマのタイムアウト時間に設定可能な最小値（秒）を格納する領域へのポインターを指定します。

復帰値

ESM_SUCCESS(0)	: 正常終了。
ESM_ERROR(-1)	: システムエラーが発生。
ESM_PRMERR(-2)	: 指定パラメーターに誤りがある。

3.9 ESM_SwitchWDT() 関数

機能説明

SDK のウォッチドッグタイマのオン／オフを設定します。EmbedWare/SysMon Entry のウォッチドッグタイマがオフの場合に設定できます。

本関数と「3.10 ESM_WDTHeartbeat() 関数」(57 ページ) は同一プロセスで実行する必要があります。ウォッチドッグタイマをオンに設定し、タイムアウト時間が経過した場合、指定したアクション動作が実行されます。

異なるプロセスでウォッチドッグタイマを操作した場合、動作は保証されません。

呼び出し形式

```
#include "esm_api.h"
int ESM_SwitchWDT(int descriptor, int time, int action, int status);
```

パラメーター

int descriptor	: ESM_Open() で獲得した識別子を指定します。
int time	: ウォッチドッグタイマのタイムアウト時間 (秒) を指定します。タイムアウト時間の指定範囲は、ESM_GetWDTTimeOut() から取得します。
int action	: ウォッチドッグタイマのタイムアウト発生時のアクションを指定します。指定できる値は以下のとおりです。
0	: 装置再起動を実施します (Hard Reset)。
1	: 何も実施しません (No Action)。
2	: 装置をシャットダウンします (Power Down)。
3	: 装置のシャットダウンを実施したあとに装置起動を実施します (Power Cycle)。



装置でサポートしているアクションは EmbedWare/SysMon Entry で確認してください。
詳細は、「3.13 コマンドラインインターフェイス」の「(10) ewsctl」(30 ページ) を参照してください。

int status	: ウォッチドッグタイマのオン／オフを設定します。指定できる値は以下のとおりです。
0	: オフ
1	: オン



ウォッチドッグタイマがすでにオンの状態でタイムアウト時間およびタイムアウト発生時のアクションを変更する場合は、一度ウォッチドッグタイマをオフにしてください。

復帰値

ESM_SUCCESS(0)	: 正常終了。
ESM_ERROR(-1)	: システムエラーが発生。
ESM_PRMERR(-2)	: 指定パラメーターに誤りがある。
ESM_WDTERR1(-11)	: EmbedWare/SysMon Entry のウォッチドッグタイマ設定がオンである。
ESM_NOSUPPORT(-13)	: サポートしていないアクションが指定された。

3.10 ESM_WDTHeartbeat() 関数

機能説明

ウォッチドッグタイマのカウントをリセットします。SDK のウォッチドッグタイマがオンの場合に有効です。

本関数と「3.9 ESM_SwitchWDT() 関数」(56 ページ) は同一プロセスで実行する必要があります。異なるプロセスでウォッチドッグタイマを操作した場合、動作は保証されません。

呼び出し形式

```
#include "esm_api.h"
int ESM_WDTHeartbeat(int descriptor);
```

パラメーター

int descriptor : ESM_Open() で獲得した識別子を指定します。

復帰値

ESM_SUCCESS(0)	: 正常終了。
ESM_ERROR(-1)	: システムエラーが発生。
ESM_PRMERR(-2)	: 指定パラメーターに誤りがある。
ESM_WDTER2(-12)	: ESM_SwitchWDT() 関数を使用してウォッチドッグタイマをオンにしていない。

3.11 ESM_ReadGPI() 関数

機能説明

GPI (General Purpose Input) の情報を取得します。

呼び出し形式

```
#include "esm_api.h"
int ESM_ReadGPI(int descriptor, BYTE *Read_Value);
```

パラメーター

int descriptor	: ESM_Open() で獲得した識別子を指定します。
BYTE *Read_Value	: GPI のデータを格納する領域へのポインターを指定します。 データの有効範囲は bit0 ~ 3 の 4 ビット。

復帰値

ESM_SUCCESS(0)	: 正常終了。
ESM_ERROR(-1)	: システムエラーが発生。
ESM_PRMERR(-2)	: 指定パラメーターに誤りがある。

3.12 ESM_ReadGPO() 関数

機能説明

GPO (General Purpose Output) の情報を取得します。

呼び出し形式

```
#include "esm_api.h"  
int ESM_ReadGPO(int descriptor, BYTE *Read_Value);
```

パラメーター

int descriptor	: ESM_Open() で獲得した識別子を指定します。
BYTE *Read_Value	: GPO のデータを格納する領域へのポインターを指定します。 データの有効範囲は bit0 ~ 3 の 4 ビットです。

復帰値

ESM_SUCCESS(0)	: 正常終了。
ESM_ERROR(-1)	: システムエラーが発生。
ESM_PRMERR(-2)	: 指定パラメーターに誤りがある。

3.13 ESM_WriteGPO() 関数

機能説明

GPO に情報を設定します。

呼び出し形式

```
#include "esm_api.h"  
int ESM_WriteGPO(int descriptor, BYTE Write_Value);
```

パラメーター

int descriptor	: ESM_Open() で獲得した識別子を指定します。
BYTE Write_Value	: GPO に設定するデータを指定します。データの有効範囲は bit0 ~ 3 の 4 ビットです。有効範囲の各ビットに 0 または 1 を指定 します。

復帰値

ESM_SUCCESS(0)	: 正常終了。
ESM_ERROR(-1)	: システムエラーが発生。
ESM_PRMERR(-2)	: 指定パラメーターに誤りがある。

3.14 ESM_GetCpuCount() 関数

機能説明

物理 CPU 数を取得します。物理 CPU の数を元に、「3.15 ESM_GetCpuPerform() 関数」(59 ページ) を利用することで CPU の実周波数情報を取得できます。

呼び出し形式

```
#include "esm_api.h"
int ESM_GetCpuCount(int descriptor, int *cpu_num);
```

パラメーター

int descriptor	: ESM_Open() で獲得した識別子を指定します。
int *cpu_num	: 物理 CPU 数を格納する領域へのポインターを指定します。

復帰値

ESM_SUCCESS(0)	: 正常終了。
ESM_ERROR(-1)	: システムエラーが発生。
ESM_PRMERR(-2)	: 指定パラメーターに誤りがある。

3.15 ESM_GetCpuPerform() 関数

機能説明

CPU の実周波数を取得します。

呼び出し形式

```
#include "esm_api.h"
int ESM_GetCpuPerform(int descriptor, int slot, int *Get_Value);
```

パラメーター

int descriptor	: ESM_Open() で獲得した識別子を指定します。
int slot	: 実周波数を取得する CPU の物理 CPU 番号 (0 ~) を指定します。物理 CPU の数は ESM_GetCpuCount() で取得します。
int *Get_Value	: CPU 実周波数 (MHz) の情報を格納する領域へのポインターを指定します。

復帰値

ESM_SUCCESS(0)	: 正常終了。
ESM_ERROR(-1)	: システムエラーが発生。
ESM_PRMERR(-2)	: 指定パラメーターに誤りがある。
ESM_NODATA1(-9)	: CPU の実周波数情報が存在しない。

3.16 ESM_GetSmartEx() 関数

機能説明

指定したディスクの S.M.A.R.T. ステータスと S.M.A.R.T. アトリビュートデータを取得します。ESM_GetSmartEx() は ESM_GetSmart() の拡張関数です。取得できる S.M.A.R.T. アトリビュートデータに違いがあります。

書式

```
#include "esm_api.h"
int ESM_GetSmartEx(int descriptor, char *Key1, smart_data_t *Get_Value);
```

パラメーター

int descriptor	: ESM_Open() で獲得した識別子を指定します。
char *Key1	: 監視または設定項目を示すキーワードへのポインターを指定します。指定できる文字列は以下のとおりです。
SLOT0 ~ SLOT7	: 指定されたディスクに関する情報を取得します。
smart_data_t *GetValue	: 情報を格納する領域へのポインター。以下に示す smart_data_t 構造体へのポインターを指定します。

smart_attribute_t, smart_data_t 構造体

```
typedef struct smart_attribute {
    unsigned char id;
    unsigned char flags[2];
    unsigned char value;
    unsigned char worst;
    unsigned char threshold;
    unsigned char raw[7];
    unsigned char reserved[3];
} smart_attribute_t;
```

```
typedef struct smart_data {
    int status;
    smart_attribute_t attributes[30];
} smart_data_t;
```



ESM_GetSmart() で扱う smart_attr_t, smart_attributes_t 構造体との差異は、以下のとおりです。

- ESM_GetSmart() と ESM_GetSmartEx() では、raw データおよび flags データの並びが違います。
- ESM_GetSmart() には raw データのうち 6 バイト分が格納されますが、ESM_GetSmartEx() には 7 バイト分すべてが格納されます。

flags データ Offset (ディスクから取得した時点での並び)	+0	+1
ESM_GetSmart()	flags[1]	flags[0]
ESM_GetSmartEx()	flags[0]	flags[1]

raw データ Offset (ディスクから取得した時 点での並び)	+0	+1	+2	+3	+4	+5	+6
ESM_GetSmart()	raw[5]	raw[4]	raw[3]	raw[2]	raw[1]	raw[0]	-
ESM_GetSmartEx()	raw[0]	raw[1]	raw[2]	raw[3]	raw[4]	raw[5]	raw[6]

補足説明

smart_data_t 構造体に格納される status の意味は、以下のとおりです。

- 0 : S.M.A.R.T. アトリビュートデータが取得できません
(この場合 attributes の格納は行われません)
- 1 : S.M.A.R.T. ステータスは正常です
- 2 : S.M.A.R.T. ステータスの異常が検出されています

復帰値

- ESM_SUCCESS(0) : 正常終了。
- ESM_ERROR(-1) : システムエラーが発生。
- ESM_PRMERR(-2) : 指定パラメーターに誤りがある。

3.17 ESM_GetSvDiskType() 関数

機能説明

RAID カード（タイプ 2）に接続されているディスクのモデル名およびシリアル番号を取得します。

書式

```
#include "esm_api.h"
int ESM_GetSvDiskType(int descriptor, int Array, int Drive, char *model, char *serial);
```

パラメーター

- int descriptor : ESM_Open() で獲得した識別子を指定します。
- int Array : 論理ボリュームの番号を指定します。0 ~ 3 の値が指定可能です。
- int Drive : ディスク番号を指定します。0 ~ 1 の値が指定可能です。
- char *model : モデル名を格納する領域へのポインターを指定します。最低 64byte (終端ヌル文字を含む) の領域を準備する必要があります。
- char *serial : シリアル番号を格納する領域へのポインターを指定します。最低 64byte (終端ヌル文字を含む) の領域を準備する必要があります。

復帰値

- ESM_SUCCESS(0) : 正常終了。
- ESM_ERROR(-1) : システムエラーが発生。
- ESM_PRMERR(-2) : 指定パラメーターに誤りがある。

留意事項

装置が RAID 構成でないまたは RAID 監視を無効に設定している場合、本関数は正常終了しますが、モデル名およびシリアル番号には空文字が格納されます。

3.18 ESM_GetSvDiskSmartAttr() 関数

機能説明

RAID カード（タイプ 2）に接続されているディスクの S.M.A.R.T. アトリビュートデータを取得します。

書式

```
#include "esm_api.h"
int ESM_GetSvDiskSmartAttr(int descriptor, int Array, int Drive, smart_attribute_t *attributes);
```

パラメーター

int descriptor	: ESM_Open() で獲得した識別子を指定します。
int Array	: 論理ボリュームの番号を指定します。0～3 の値が指定可能です。
int Drive	: ディスク番号を指定します。0～1 の値が指定可能です。
smart_attribute_t *attributes	: 情報を格納する領域へのポインター。以下に示す smart_attribute_t 構造体へのポインターを指定します。 smart_attribute_t 配列 30 個分の格納領域を用意する必要があります。
smart_attribute_t 構造体	
typedef struct smart_attribute {	
unsigned char id;	
unsigned char flags[2];	
unsigned char value;	
unsigned char worst;	
unsigned char threshold;	
unsigned char raw[7];	
unsigned char reserved[3];	
} smart_attribute_t;	

復帰値

ESM_SUCCESS(0)	: 正常終了。
ESM_ERROR(-1)	: システムエラーが発生。
ESM_PRMERR(-2)	: 指定パラメーターに誤りがある。

留意事項

装置が RAID 構成でないまたは RAID 監視を無効に設定している場合、本関数は正常終了しますが、S.M.A.R.T. アトリビュートデータはすべて 0 の値が格納されます。

第4章 API 使用方法



この章では、EmbedWare/SysMon SDK の API 使用法について説明します。

4.1 API マルチスレッド対応

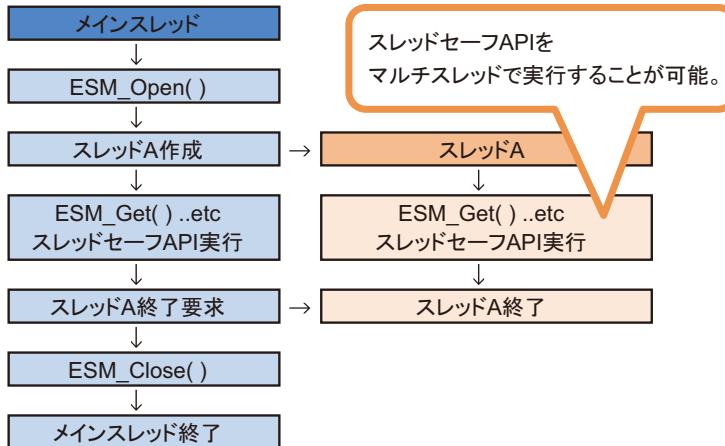
マルチスレッドでほかの API と同時に使用可能な API（以下、スレッドセーフ API）と、マルチスレッドでほかの API と一緒に使用できない API（以下、スレッドアンセーフ API）が存在します。

スレッドセーフ API	スレッドアンセーフ API
ESM_Get()	ESM_Open()
ESM_GetSmart()	ESM_Close()
ESM_RAIDInfo()	ESM_SwitchWDT()
ESM_SvRAIDInfo()	ESM_WDTHeartbeat()
ESM_QueryError()	
ESM_GetWDTTimeOut()	
ESM_ReadGPI()	
ESM_ReadGPO()	
ESM_WriteGPO()	
ESM_GetCpuCount()	
ESM_GetCpuPerform()	
ESM_GetSmartEx()	
ESM_GetSvDiskType()	
ESM_GetSvDiskSmartAttr()	

4.2 API 使用例

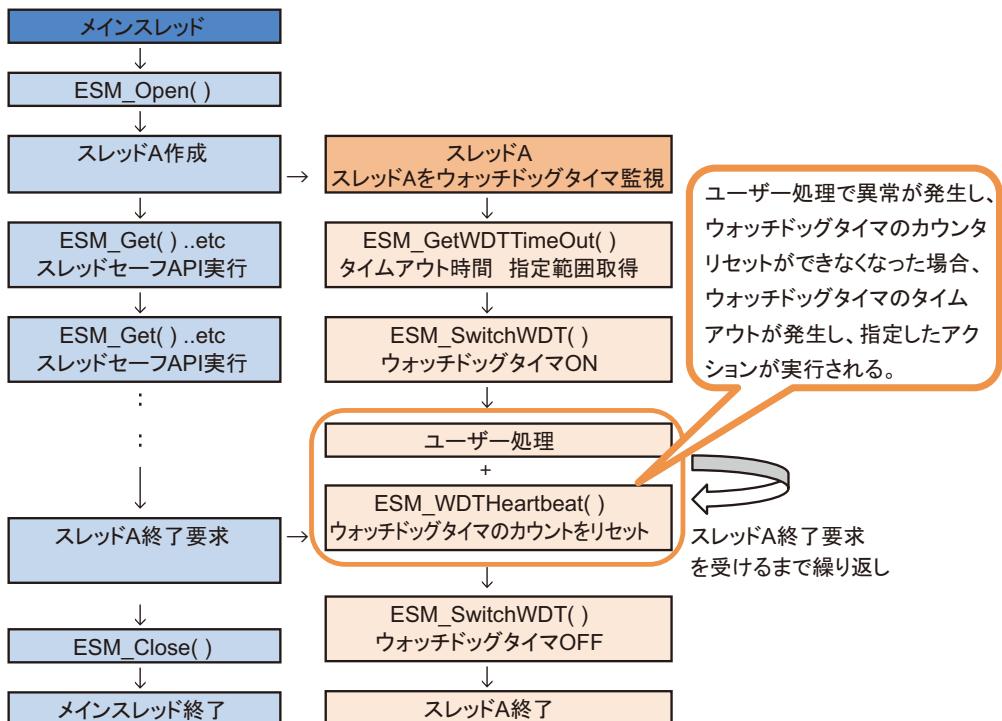
API 使用例を以下に示します。

◇ API 実行



◇ API を使用したウォッチドッグタイマ監視

スレッド A (ユーザー処理) のハングアップをウォッチドッグタイマで監視する例



付録

付

付録 A 通知メッセージ一覧

EmbedWare/SysMon Entry でコンピューターの動作状態が変更された場合に通知されるメッセージを以下に示します。

◇情報

通知される情報メッセージを以下に示します。Alert 機能有効時にシステムログにメッセージが格納されます。ビープ音は鳴りません。

No.	メッセージ	意味
1	<センサー名> temperature is normal.	<センサー名> の温度が正常に戻った。
2	<センサー名> voltage is normal.	<センサー名> の電圧が正常に戻った。
3	<センサー名> fan speed is normal.	<センサー名> のファン回転数が正常に戻った。
4	SLOTX disk is normal.	HDD（非 RAID SATA ディスク）が復旧した。
5	PCI Error was reset.	PCI エラー検出状態をリセットした。
6	Memory Error was reset.	メモリエラー検出状態をリセットした。
7	BMC Error was reset.	BMC エラー検出状態をリセットした。
8	RAID Card: Information [CNTLXX CNXX DEVXX: <Status>]…	RAID カード（タイプ1）のディスク情報。 装置起動時に出力される。ディスク情報は監視デバイスの数分表示される。
9	SVRx RAID ArrayX: Information [Drive0: <Model>, <SerialNumber>, <Status>] [Drive1: <Model>, <SerialNumber>, <Status>]	RAID カード（タイプ2）のディスク情報。 装置起動時に出力される。
10	RAID CNTLXX CNXX DEVXX Rebuild start	RAID カード（タイプ1）のディスクがリビルトを開始した。
11	SVRx RAID ArrayX: Rebuild start [Drive0: <Model>, <SerialNumber>, <Status>] [Drive1: <Model>, <SerialNumber>, <Status>]	RAID カード（タイプ2）のディスクがリビルトを開始した。
12	RAID CNTLXX CNXX DEVXX Rebuild completed	RAID カード（タイプ1）のディスクがリビルトを完了した。
13	SVRx RAID ArrayX: Rebuild completed [Drive0: <Model>, <SerialNumber>, <Status>] [Drive1: <Model>, <SerialNumber>, <Status>]	RAID カード（タイプ2）のディスクがリビルトを完了した。

No.	メッセージ	意味
14	RAID CNTLXX CNXX DEVXX is normal.	RAID カード（タイプ 1）のディスクが故障、または S.M.A.R.T. による異常から復旧。 リビルド完了時も復旧したとみなす。
15	SVRx RAID ArrayX: Normal [Drive0: <Model>, <SerialNumber>, <Status>] [Drive1: <Model>, <SerialNumber>, <Status>]	RAID カード（タイプ 2）のディスクが故障、または S.M.A.R.T. による異常から復旧。 リビルド完了時も復旧したとみなす。
16	EWS Entry Service Start	EmbedWare/SysMon Entry のサービスが開始した。
17	EWS Entry Service Stop	EmbedWare/SysMon Entry のサービスが停止した。 シャットダウン時、syslog デーモンが先に終了する場合、情報出力されない。



- "X" は可変の数値を表します。
- "<"、">" で囲んでいる箇所は可変の文字列を表します。

◇警告

通知される警告メッセージを以下に示します。Alert 機能有効時にシステムログにメッセージが格納されます。ビープ音が鳴ります。

No.	メッセージ	意味	対処
1	<センサー名> temperature upper threshold breached: XXX[degrees C]	<センサー名> で温度異常を検出した。	吸気温度の場合は空調を、CPU 温度などの場合はファン状態を確認するか、または担当営業にご相談ください。
2	<センサー名> temperature is abnormal.		
3	<センサー名> temperature lower threshold breached: XXX[degrees C]		
4	<センサー名> voltage upper threshold breached: XXX[V]	<センサー名> で電圧異常を検出した。	担当営業にご相談ください。
5	<センサー名> voltage lower threshold breached: XXX[V]		
6	<センサー名> fan speed lower threshold breached: XXX[rpm]	<センサー名> でファン回転数異常を検出した。	ファン故障の可能性があります。ファン回転数をご確認のうえ、担当営業にご相談ください。
7	<センサー名> fan speed upper threshold breached: XXX[rpm]		

付

No.	メッセージ	意味	対処
8	SLOTX disk is abnormal.	S.M.A.R.T. 機能による故障予測を検出した。	ディスク交換を推奨します。詳細については、担当営業にご相談ください。
9	PCI Error [yy/mm/dd hh:mm:ss < 詳細データ >]	PCI エラーを検出した。	BIOS イベントログに PCI エラー発生記録が存在します。 BIOS イベントログをご確認のうえ、担当営業にご相談ください。
10	Memory Error [yy/mm/dd hh:mm:ss < 詳細データ >]	メモリエラーを検出した。	BIOS イベントログにメモリエラー発生記録が存在します。 BIOS イベントログをご確認ください。マルチビット ECC メモリエラーが発生している場合、またはシングルビット ECC メモリエラーが頻発している場合はメモリ故障の可能性があるため、担当営業にご相談ください。
11	BMC Error [yy/mm/dd hh:mm:ss < 詳細データ >]	BMC エラーを検出した。	BIOS イベントログに BMC エラー発生記録が存在します。 BIOS イベントログをご確認のうえ、担当営業にご相談ください。
12	RAID CNTLXX CNXX DEVXX is Failed.	RAID カード(タイプ1) のディスク故障を検出した。	ディスクを交換してください。

No.	メッセージ	意味	対処
13	RAID CNTLXX CNXX DEVXX is S.M.A.R.T. ERROR.	RAID カード (タイプ 1) のディスクが S.M.A.R.T. による異常を検出した。	ディスク交換を推奨します。詳細については、担当営業にご相談ください。
14	SVRx RAID ArrayX: DriveX is Failed [Drive0: <Model>, <SerialNumber>, <Status>] [Drive1: <Model>, <SerialNumber>, <Status>]	RAID カード (タイプ 2) のディスク故障を検出した。	ディスクを交換してください。
15	SVRx RAID ArrayX: DriveX is S.M.A.R.T. ERROR [Drive0: <Model>, <SerialNumber>, <Status>] [Drive1: <Model>, <SerialNumber>, <Status>]	RAID カード (タイプ 2) のディスクが S.M.A.R.T. による異常を検出した。	ディスク交換を推奨します。詳細については、担当営業にご相談ください。
16	SVRx RAID ArrayX: DriveX is SKIP RECON [Drive0: <Model>, <SerialNumber>, <Status>] [Drive1: <Model>, <SerialNumber>, <Status>]	RAID カード (タイプ 2) のディスクがスキップ・リコンを検出した。	担当営業にご相談ください。



- "X" は可変の数値を表します。
- "<"、">" で囲んでいる箇所は可変の文字列を表します。
- "yy/mm/dd hh:mm:ss" は日時を表します。

◇エラー

通知される異常メッセージを以下に示します。Alert 機能の有効・無効にかかわらず、出力されます。ビープ音は鳴りません。

No.	メッセージ	意味	対処
1	BMC open error.	BMC のオープンに失敗した。	システムに IPMI モジュールがロードされているか確認してください。システムに IPMI モジュールをロードしてください。それでも問題が発生する場合は、担当営業にご相談ください。
2	Configuration file is broken.	設定ファイルが壊れている。	担当営業にご相談ください。
3	Internal error.	内部異常を検出した。	担当営業にご相談ください。

付録B コマンドラインインターフェイスメッセージ一覧

EmbedWare/SysMon Entry のコマンドラインインターフェイスで表示される可能性のあるエラーメッセージを以下に示します。

No.	メッセージ	意味	対処
1	-x option argument not found.	-x オプションに引数が見つからない。	-x オプションに引数を指定してください。
2	-x option too many argument.	-x オプションの引数の数が多すぎる。	-x オプションの引数の数を確認してください。
3	-x option argument can't specify.	-x オプションは引数を指定できない。	-x オプションに引数を指定しないでください。
4	-x option is no argument.	-x オプションの引数がない。	-x オプションに有効な引数を指定してください。
5	-x option can't specify simultaneously with other options.	-x オプションはほかのオプションと同時に指定できない。	-x オプションとほかのオプションを同時に指定しないでください。
6	-x and -y options can't be specified simultaneously.	-x と -y オプションは同時に指定できない。	-x オプションと -y オプションを同時に指定しないでください。
7	One of -x and the -y needs to specify.	-x と -y オプションのどちらかを指定する必要がある。	-x と -y オプションのどちらかを指定してください。
8	Options can't specify simultaneously with other options.	オプションはほかのオプションと同時に指定できない。	オプションはほかのオプションと同時に指定しないでください。
9	Options needs to specify.	オプションを指定する必要がある。	オプションを指定してください。
10	XXX and -x option and -y options can't be specified simultaneously.	XXX と -x と -y オプションは同時に指定できない。	XXX と -x オプションと -y オプションを同時に指定しないでください。
11	XXX is not exist.	XXX という監視項目は存在しない。	引数 XXX に有効な監視項目を入力してください。
12	XXX option are few arguments.	XXX の引数が足りない。	XXX の引数の数を確認してください。
13	XXX is invalid argument.	XXX は無効な引数。	引数 XXX を確認してください。
14	XXX is out of range.	XXX は設定範囲外。	XXX に設定範囲内の数値を入力してください。
15	XXX is a directory.	XXX はディレクトリ。	指定したパスを確認してください。
16	XXX directory is not exists.	XXX というディレクトリが存在しない。	

No.	メッセージ	意味	対処
17	path is too long.	指定されたパスが長すぎる。	パスを 128 字以内で指定してください。
18	File:Failed to save XXX. YYY.	ファイルの保存に失敗した。 XXX にはファイル名が、YYY には失敗した理由が表示される。	YYY の内容に応じて処置を行ってください。
19	UpperThreshold cannot changed in XXX.	監視項目 XXX の上限値は変更できない。	監視項目 XXX の上限値を変更しないでください。
20	LowerThreshold cannot changed in XXX.	監視項目 XXX の下限値は変更できない。	監視項目 XXX の下限値を変更しないでください。
21	An incorrect threshold value was specified.	しきい値が不正。	指定したしきい値を確認してください。
22	sysmond isn't started.	sysmond デーモンが起動していない。	EmbedWare/SysMon Entry のサービスプログラムが初期処理中の可能性があります。しばらく待ってから、コマンド操作を実施してください。コマンド再操作で再びエラーが発生した場合は、担当営業にご相談ください。
23	Permission Denied	コマンドのアクセス権限がない。	root (システム管理者) 権限でコマンドを実行してください。
24	BMC is not supported.	BMC はサポートされていない。	BMC 搭載有無を確認してください。
25	BMC access error< エラーコード >.	BMC アクセスでエラーが発生した。	実行中の BMC 処理終了後、再度実行してください。 再実行でエラーが発生した場合は、担当営業にご相談ください。
26	xml open error.	XML のオープンエラーが発生した。	担当営業にご相談ください。
27	internal error< エラーコード >.	内部エラーが発生した。	
28	Initialize < リソース名 > error.	リソースの初期化に失敗した。	
29	Unexpected option "< オプション >" is there.	期待外のオプションが指定された。	コマンド仕様をご確認ください。
30	Expected argument for option "< オプション >" is missing.	オプションに引数が指定されていない。	

No.	メッセージ	意味	対処
31	Skipped since device <デバイス名> is not support ATA IdentifyDevice.	処理対象外のデバイスをスキップした。	対処不要です。
32	Skipped since device <デバイス名>(<モデル名>) is not <RAID カード種別>.		
33	-x option can't be specified more than once.	-x オプションは複数回指定できません。	-x オプションを複数回指定しないでください。



- ・ "<"、">" で囲んでいる箇所は、可変の文字列を表します。
- ・ 担当営業にご相談ください。
- ・ "-x"、"-y" は可変のオプションを表します。

付録 C EmbedWare/SysMon SDK API ディスク情報取得について

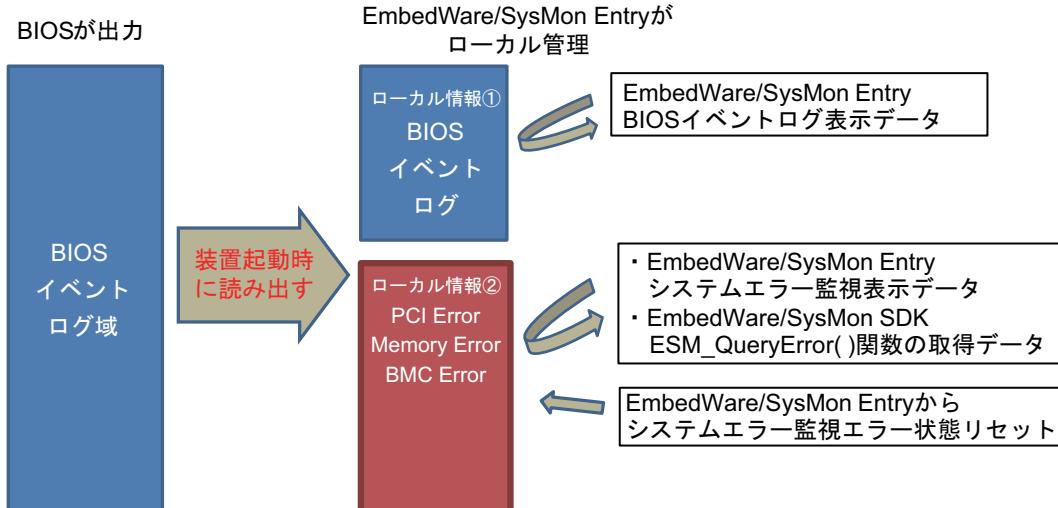
ディスク情報取得で使用する API を以下に示します。

取得できる情報	非 RAID ディスクの場合	RAID カード（タイプ1）、チップセット RAID の場合	RAID カード（タイプ2）の場合
監視項目名	ESM_Get()	—	—
モデル名	ESM_Get()	—	ESM_GetSvDiskType()
シリアル番号	ESM_Get()	—	ESM_GetSvDiskType()
監視の有効・無効	ESM_Get()	ESM_Get()	ESM_Get()
S.M.A.R.T. ステータス	ESM_Get() ESM_GetSmart() ESM_GetSmartEx()	ESM_RAIDInfo() ※S.M.A.R.T. ステータス 以外に RAID 状態も取得できます。	ESM_SvRAIDInfo() ※S.M.A.R.T. ステータス以外に RAID 状態も取得できます。
S.M.A.R.T. アトリビュート データ	ESM_GetSmart() ESM_GetSmartEx()	—	ESM_GetSvDiskSmartAttr()

付録 D システムエラー監視の仕組みについて

EmbedWare/SysMon Entry のシステムエラー監視の仕組みを、以下に示します。

- EmbedWare/SysMon Entry は、装置起動時に BIOS イベントログ域からログを読み出し、ローカル情報①として保持します。この中でエラーを検出した場合、ローカル情報②にエラー情報を格納します。
- 以降の装置起動では、BIOS イベントログ域に新たに追加されたログだけを読み出し、ローカル情報①へ追加します。追加ログの中でエラーを検出した場合、ローカル情報②にエラー情報を格納します。
- BIOS イベントログ域をクリアしても、ローカル情報①②は削除しません。
- エラー検出時、すでに同種のシステムエラーがローカル情報②に格納済みの場合、新しく検出したエラーは格納しません。
- ローカル情報②に格納したシステムエラーは、EmbedWare/SysMon Entry からリセットできます。



EmbedWare/SysMon Ver 7.1 for Linux 説明書
P2U3-0220-02Z0

発行日 2019年11月

発行責任 株式会社 PFU

- 本書の内容は、改善のため事前連絡なしに変更することがあります。
- 本書に記載されたデータの使用に起因する、第三者の特許権およびその他の権利の侵害については、当社はその責を負いません。
- 無断転載を禁じます。