



OSS を使用した Express サーバの運用管理 (Linux 環境)

2013/07/05 Rev. 2.0

目次

目次	II
■商標について	IV
■ご注意	IV
■本書について	IV
■用語	V
■ REVISION HISTORY	VI
1. <u>サーバ管理概要</u>	- 1 -
2. <u>使用するツール</u>	- 2 -
2.1. ツール概要	- 2 -
2.2. 構成	- 2 -
2.3. ツール使用方法	- 3 -
2.3.1. ローカルコンピュータ上での使用方法	- 3 -
2.3.2. リモート管理 PC 上からの使用方法(ユーザ認証機能を使用する場合)	- 3 -
2.3.3. リモート管理 PC 上からの使用方法(ユーザ認証/暗号化機能を使用する場合)	- 3 -
3. <u>初期設定</u>	- 5 -
3.1. 管理対象サーバ設定	- 6 -
3.1.1. OpenIPMI 関連パッケージのインストールおよび起動設定	- 6 -
3.1.2. IPMITool のインストール	- 6 -
3.1.3. 再起動	- 6 -
3.1.4. BMC アクセス用アカウント作成	- 6 -
3.1.4.1. ユーザ情報のサマリ確認	- 6 -
3.1.4.2. 現在定義されているユーザ情報確認	- 7 -
3.1.4.3. 新規ユーザアカウント作成	- 7 -
3.1.4.4. パスワード設定	- 7 -
3.1.4.5. 権限設定	- 8 -
3.1.4.6. アカウントを有効にする	- 8 -
3.1.4.7. アカウントを削除する場合	- 8 -
3.1.5. BMC の LAN 設定	- 8 -
3.1.5.1. 使用する LAN Channel 確認	- 8 -
3.1.5.2. IP アドレス設定	- 8 -
3.1.5.3. 設定した項目確認	- 9 -

3.1.5.4. LAN Channel を有効にする	- 9 -
3.2. 管理 PC 設定	- 10 -
3.2.1. IPMITool のインストール	- 10 -
3.3. 管理対象サーバにおける BMC 設定初期化(EXPRESSSCOPE エンジン 2)	- 10 -
3.4. 管理対象サーバにおける BMC 設定初期化(EXPRESSSCOPE エンジン 3)	- 12 -
4. <u>サーバ監視の活用例</u>	- 13 -
4.1. IPMI 情報の表示	- 13 -
4.1.1. BMC 情報	- 13 -
4.1.2. SDR 情報	- 13 -
4.1.3. FRU 情報	- 15 -
4.1.4. SEL 情報	- 15 -
4.2. センサ状態表示	- 17 -
4.3. 電源制御	- 18 -
4.4. 任意の IPMI コマンド実行	- 19 -
4.5. その他のコマンド	- 19 -
4.5.1. BMC の自己診断	- 19 -
5. <u>ウォッチドッグタイマを使用した OS のストール監視／再起動</u>	- 20 -
5.1. 機能概要	- 20 -
5.2. 設定手順	- 20 -
5.3. 監視を無効化する手順	- 23 -
5.4. タイムアウト時に NMI を発生させる設定	- 23 -
6. <u>SERIAL OVER LAN 機能を使用する</u>	- 27 -
6.1. システム BIOS 設定	- 27 -
6.2. OS の設定ファイル編集	- 28 -
6.2.1. /etc/inittab 編集	- 28 -
6.2.2. /etc/securetty 編集	- 28 -
6.3. GRUB の設定ファイル編集	- 28 -
6.4. EXPRESSSCOPE エンジンの設定	- 29 -
6.5. SOL を使用する	- 30 -
6.6. SOL を終了する	- 32 -
■ <u>APPENDIX</u>	- 33 -

■商標について

Linux は、Linus Torvalds 氏の米国およびその他の国における商標または登録商標です。
ESMPRO、DianaScope および EXPRESSBUILDER は、日本電気株式会社の登録商標です。
その他、本書で記載されている会社名および製品名は各社の登録商標または商標です。

■ご注意

本書の内容の一部または全部を無断転載することは禁止されています。
本書の内容に関しては将来予告なしに変更することがあります。
日本電気株式会社の許可なく複製・改変などを行うことはできません。
本書の内容については万全を期しておりますが、日本電気株式会社は、本書中の技術的あるいは編集上の誤り、欠落に対して、責任を負いかねます。

■本書について

本書では、オープンソースソフトウェア(以下 OSS)を使用して、IPMI (Intelligent Platform Management Interface)に準拠した Express サーバを管理する方法および使用例についてご紹介します。通常、Express サーバの管理のためには装置に添付されている ESMPRO、DianaScope または BMC が直接提供する Web サーバ機能を使用しますが、ESMPRO をご使用にならない Linux 環境(例えば、ESMPRO が未サポートの Linux ディストリビューションをご使用になる場合等)において、コマンドラインから直接 IPMI 機能を利用したサーバ管理を行う際にご参照ください。

本書における動作例の記載で使⽤した Express サーバ及び Linux ディストリビューション

サーバ : *Express5800/iR110a-1H、Express5800/R120d-1M*

OS 環境 : *CentOS 5.2*

■用語

用語	説明
管理対象サーバ	管理対象となる BMC を搭載した Express5800 シリーズのサーバ(本書の動作例では Express5800/iR110a-1H を使用)
管理 PC	リモートから管理対象サーバを運用管理する PC 相当の端末
BMC	Baseboard Management Controller システムの状態や OS に依存することなく、システムのハードウェアの監視機能を提供する IPMI 仕様に準拠した管理用コントローラ
FRU	Field Replaceable Unit IPMI 仕様にて規定された保守交換ユニット
IPMI	Intelligent Platform Management Interface システムの状態や OS に依存することなく、サーバのハードウェアを監視するための標準インターフェース仕様
NMI	Non-Maskable Interrupt マスク不可能なハードウェア割込み
OSS	Open Source Software インターネットなどを通じてソースコードが無償公開されており、誰でも改良、再配布が行えるようになっているソフトウェア
RMCP	Remote Management Control Protocol IPMI Ver.1.5 で規定されているリモート管理するための通信プロトコル(ユーザ認証機能をサポート)
RMCP+	Remote Management Control Protocol Plus IPMI Ver.2 で規定されているリモート管理するための通信プロトコル(ユーザ認証/暗号化機能をサポート)
SDR	Sensor Data Record IPMI 仕様にて規定されたハードウェア監視に使われるセンサ情報レコード
SEL	System Event Log IPMI 仕様にて規定されたハードウェアログ情報
SOL	Serial Over LAN Serial に出力されるデータを BMC が UDP Packet 化して LAN に流す機能。
BMC Configuration Tool	EXPRESSSCOPE エンジン 3 搭載装置に対応したツール。POST 中に「F4」を押して起動するオフライン版と、OS 上にインストールするオンライン版が存在。

■ Revision History

- | | | |
|-----|------------|--|
| 1.0 | 2009/04/27 | 新規作成 |
| 1.1 | 2010/06/23 | 3.1.4 章 BMC アクセス用アカウント作成 <ul style="list-style-type: none">➤ ヒント追加➤ アカウント削除方法追加 |
| | | 4.1.4 章 SEL 情報 <ul style="list-style-type: none">➤ ヒント追加 |
| | | 6章 Serial Over LAN (新規追加) |
| 2.0 | 2013/07/05 | EXPRESSSCOPE エンジン 3 に対応 |

1. サーバ管理概要

IPMI(Intelligent Platform Management Interface)は、特定のハードウェアシステムや OS に依存することなく、温度、電圧、ファン、電源、筐体等といったシステムの物理的な状態を監視するための共通インターフェース仕様です。IPMI を使用することで、システムの電源状態に依存せず、ハードウェア状態の監視、リモートからのセキュアな診断、復旧、資産管理等を行うことができます。

IPMI 仕様は現在、200 を超す主要なサーバベンダにて採用されており、NEC、Intel、HP、DELL の 4 社のプロモータにて仕様作成が行われています(Appendix 参照)。

NEC が提供する Express サーバの大部分には、IPMI 仕様に準拠した BMC (Baseboard Management Controller)が搭載されています。なお、Express サーバの IPMI 対応状況に関しては、Express5800 シリーズポータルサイト(Appendix 参照)に掲載されている対象装置の「システム構成ガイド」または「テクニカルガイド サーバマネージメント編」を参照する事により確認することができます。

Express サーバでは ESMPRO、DianaScope 等、IPMI に準拠した管理ソフトウェアを標準提供していますが、お客様の運用環境に応じて、IPMI に対応した OSS を使用する事により、一部のサーバ管理機能を実現する事も可能となってきています。

2. 使用するツール

本書では、Linux 環境における IPMI 機能を使用した OSS で現在、最もポピュラーなコマンドラインツールである **IPMITool** と、**OpenIPMI ドライバ** を使用したサーバ管理方法について紹介します。

2.1. ツール概要

IPMITool は、Linux ディストリビューションに標準で提供されていることもあり、本書では以下のパッケージを使用しています。

OpenIPMI-tools-2.0.6-6.el5.[i386|x86_64].rpm

また、**IPMITool** は Web 上からも入手可能です。(Appendix 参照)

なお、**IPMITool**をローカル環境で使用する際は、同様にOSSとして提供されている**OpenIPMI ドライバ**が必要となります(詳細は 2.2 章を参照)。

本書では Linux ディストリビューションに含まれている以下の **OpenIPMI ドライバ**関連のパッケージを使用しています。

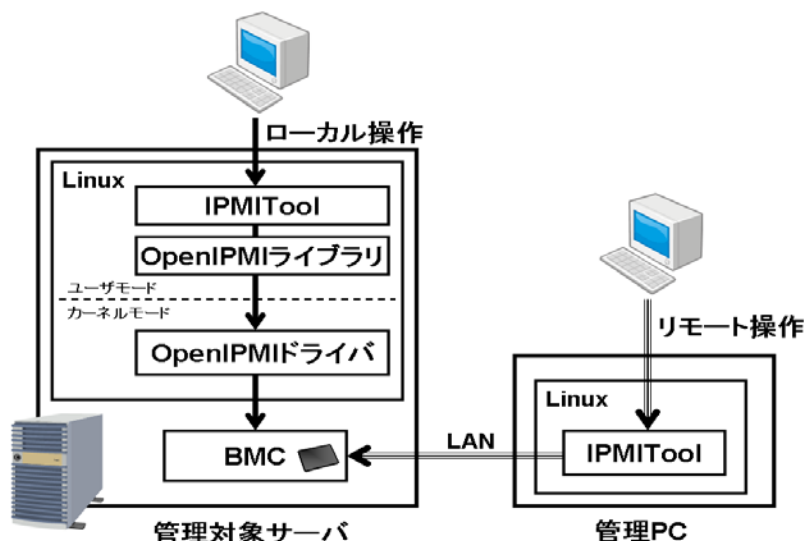
◇ OpenIPMI ドライバ : OpenIPMI-2.0.6-6.el5.[i386|x86_64].rpm

◇ OpenIPMI ライブラリ : OpenIPMI-libs-2.0.6-6.el5.[i386|x86_64].rpm

ディストリビューションに含まれていない場合は Web 上からも入手可能です。(Appendix 参照)

2.2. 構成

IPMITool は、下図のように管理対象サーバ単体(ローカル環境)、及び管理 PC を使用したリモート環境で使うことができます。



ローカル環境で使用する場合、管理対象サーバ上のコンソールで操作します。
リモート環境の場合、管理対象サーバに所定の設定を行うことで、Linux が動作する一般的な管理 PC 相当から操作を行うことができます。



本書は Express サーバにおいて OSS (IPMITool、OpenIPMI ドライバ等) の動作保証を行うものではありません。

2.3. ツール使用方法

IPMITool の使用方法について、説明します。

なお、IPMITool を使用する際のコマンド”ipmitool”の実行例に記載する引数表記については、以下の通りとしています。

<value>	<>内は省略不可
[value]	[]内は省略可能
value1 value2	value1 または value2 のどちらかを入力

2.3.1. ローカルコンピュータ上での使用方法

```
# ipmitool <command option>
```

2.3.2. リモート管理 PC 上からの使用方法(ユーザ認証機能を使用する場合)

```
# ipmitool -I lan -H <IP Address> -U <userid> -P <password> <command option>
```

※ IP Address/userid/password は、管理対象サーバの設定(3.1 章参照)で登録するものを使用します。下線部分以外はローカルコンピュータ上で実行する場合と同様の形式です。



◇ セッション確立時のユーザ認証には MD5 が使用されます (認証設定は -A オプションにて変更可能)

◇ 管理者権限にて実行されます (実行権限は -L オプションで変更可能)
詳細は IPMI 仕様および IPMITool のマニュアルを参照してください
(Appendix 参照)

2.3.3. リモート管理 PC 上からの使用方法(ユーザ認証/暗号化機能を使用する場合)

```
# ipmitool -I lanplus -H <IP Address> -U <username> -P <password> <command option>
```

※ IP Address/userid/password は、管理対象サーバの設定(3.1 章参照)で登録するものを使用します。下線部分以外はローカルコンピュータ上で実行する場合と同様の形式です。



◇ ユーザ認証/暗号化には RMCP+ の暗号スイート 3 (以下の設定) が使用されま
す (暗号スイートは -C オプションで変更可能)

セッション確立時の認証 : RAKP-HMAC-SHA1

ペイロードデータの認証 : HMAC-SHA1-96

ペイロードデータの暗号化 : AES-CBC-128

◇ 管理者権限にて実行されます (実行権限は -L オプションで変更可能)
詳細は IPMI 仕様および IPMITool のマニュアルを参照してください
(Appendix 参照)

以降は、特に記載が無い場合、IPMITool の実行例は、`ipmitool <command option>` の形式で記載します。リモートの管理 PC から実行する場合は、2.3.2 章および 2.3.3 章の説明に従って引数(-I/-H/-U/-P)を指定してください。

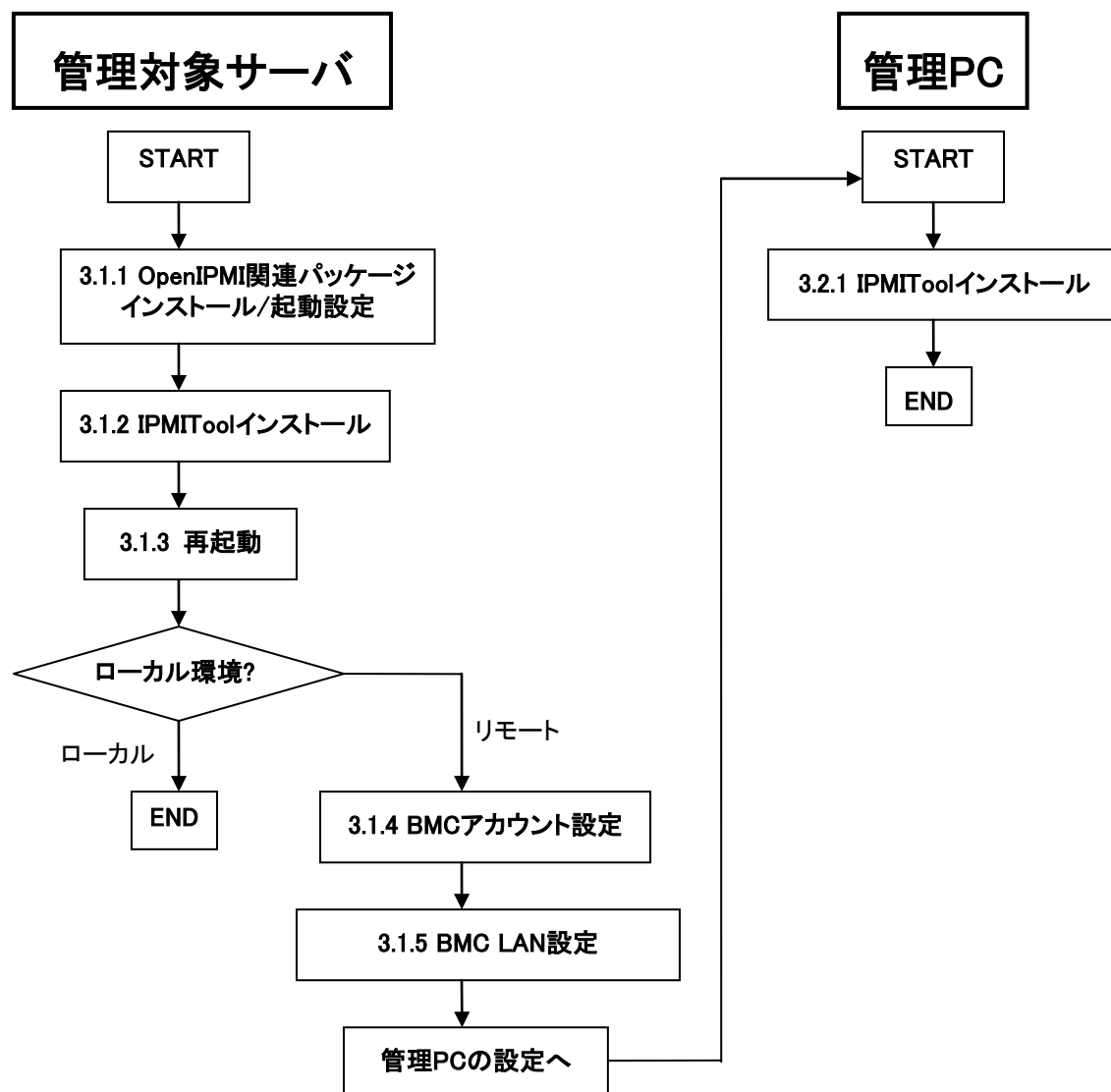
また、IPMITool は、本書で紹介するコマンドやオプション以外にも数多くの機能があります。IPMITool の詳細については man page などで確認することが出来ます。

`# man ipmitool`

3. 初期設定

ローカル及びリモート環境で IPMITool を使用したサーバ管理を行うために、管理対象サーバや管理 PC に、OSS のインストールや BMC の初期設定を行います。

以下に、基本的な初期設定手順のフローを記します。



3.1. 管理対象サーバ設定

パッケージのインストールや設定作業は管理者権限のあるユーザで行います。また、OS(本書の実行例では CentOS 5.2)がインストールされている事を前提としますので OS インストール方法に関しては説明を省略します。

なお、rpm パッケージ/コマンドを使用しないディストリビューション上でのインストール/起動設定は、各ディストリビューションの仕様に従ってください。

3.1.1. OpenIPMI 関連パッケージのインストールおよび起動設定

OpenIPMI 関連パッケージ(OpenIPMI ドライバおよびライブラリ)のインストール

```
# rpm -ivh OpenIPMI-libs-(version).rpm
```

```
# rpm -ivh OpenIPMI-(version).rpm
```

※ OpenIPMI は OpenIPMI-libs と依存関係があります。

OpenIPMI ドライバ起動設定

```
# chkconfig ipmi on
```

3.1.2. IPMITool のインストール

```
# rpm -ivh OpenIPMI-tools-(version).rpm
```

3.1.3. 再起動

```
# reboot
```

ローカル環境で使用する場合は、ここまでの設定で完了です。

リモート環境で使用する場合は、引き続き以下の設定を行います。

3.1.4. BMC アクセス用アカウント作成

3.1.4.1. ユーザ情報のサマリ確認

```
# ipmitool user summary
```

実行結果例

```
Maximum IDs          : 16
Enabled User Count    : 0
Fixed Name Count      : 1
```

3.1.4.2. 現在定義されているユーザ情報確認

ipmitool user list

実行結果例(EXPRESSSCOPE エンジン 2 の場合)

ID	Name	Callin	Link	Auth	IPMI Msg	Channel Priv	Limit
4	administrator	true	false	false	false	Unknown	(0x0)

※ ID の 1 から 4 までは、NEC が提供する管理ツールが使用する場合があります。
ID4 の「administrator」は、デフォルト状態で設定されています。

実行結果例(EXPRESSSCOPE エンジン 3 の場合)

ID	Name	Callin	Link	Auth	IPMI Msg	Channel Priv	Limit
5	Administrator	true	true	true	true	ADMINISTRATOR	

※ ID の 1 から 5 までは、NEC が提供する管理ツールが使用する場合があります。
ID5 の「Administrator」は、デフォルト状態で設定されています。初期パスワードについては、EXPRESSSCOPE エンジン 3 ユーザーズガイドを参照して下さい。

3.1.4.3. 新規ユーザアカウント作成

ipmitool user set name <userid> <username>

※ userid については、EXPRESSSCOPE エンジン 2 の場合は 5 以降、EXPRESSSCOPE エンジン 3 の場合は 6 以降で、現在未定義、且つ、summary で確認した Maximum IDs までの ID を指定

username については 15byte 以内の文字列

実行例: userid を 6、username を “testuser” とする場合

ipmitool user set name 6 testuser


3.1.4.4. パスワード設定


ipmitool user set password <userid> [<password>]

※ password を省略した場合、パスワードは消去される

実行例: userid “6”に対して、“expressOpenIPMIuser”を password として設定する場合

ipmitool user set password 6 expressOpenIPMIuser

◇  パスワード長が 16Byte 以下の場合、RMCP 用に 16Byte Password が作成されます。リモート操作時は -l lan オプションを使用します。(2.3.2 章参照)

 パスワード長が 16Byte よりも大きい場合、RMCP+用に 20Byte Password が作成されます。リモート操作時は -l lanplus オプションを使用します。(2.3.3 章参照)

3.1.4.5. 権限設定

```
# ipmitool channel setaccess <channel number> <userid> [callin=on|off]  
[ipmi=on|off] [link=on|off] [privilege=level]
```

※ channel number については 3.1.5.1 を参照。

privilege level は User は 2、Operator は 3、Administrator は 4 を指定。

3.1.4.6. アカウントを有効にする

```
# ipmitool user enable <userid>
```

※ 無効にする場合

```
# ipmitool user disable <userid>
```

3.1.4.7. アカウントを削除する場合

以下のコマンドを実行することにより、不要なアカウントを削除することができます。

```
# ipmitool user set name <userid> ""
```

3.1.5. BMC の LAN 設定

3.1.5.1. 使用する LAN Channel 確認

```
# ipmitool channel info <channel number>
```

※ channel number として、1 から 7 まで実行して、Medium Type が

“802.3 LAN” の時の number を調べる

実行例: channel number 2 の情報取得

```
# ipmitool channel info 2
```

Channel 0x2 info:

Channel Medium Type : 802.3 LAN

<===== この出力を確認

Channel Protocol Type : IPMB-1.0

Session Support : multi-session

Active Session Count : 0

Protocol Vendor ID : 7154

<以下省略>

この場合、以降の設定において、channel number は 2 となる。

3.1.5.2. IP アドレス設定

◇ IP アドレスを dhcp で設定する場合

```
# ipmitool lan set <channel number> ipsrc dhcp
```

◇ IP アドレスを手動で設定する場合

```
# ipmitool lan set <channel number> ipsrc static
```

IP アドレス設定

```
# ipmitool lan set <channel number> ipaddr <x.x.x.x>
ネットマスク設定
# ipmitool lan set <channel number> netmask <x.x.x.x>
デフォルトゲートウェイ設定
# ipmitool lan set <channel number> defgw ipaddr <x.x.x.x>
```



ヒント

IP アドレスの設定は、EXPRESSSCOPE エンジン 2 搭載装置の場合はシステム BOIS のセットアップメニューから、EXPRESSSCOPE エンジン 3 搭載装置の場合は EXPRESSSCOPE エンジン 3 の WebConsole またはオンライン・オフライン BMC Configuration Tool から行うことができます。設定詳細については、装置のユーザーズガイドを参照してください。

3.1.5.3. 設定した項目確認

```
# ipmitool lan print
```

実行結果例

```
Set in Progress      : Set Complete
Auth Type Support    : NONE MD5 PASSWORD
Auth Type Enable     : Callback : NONE MD5 PASSWORD
                     : User      : NONE MD5 PASSWORD
                     : Operator  : NONE MD5 PASSWORD
                     : Admin     : NONE MD5 PASSWORD
                     : OEM       : NONE MD5 PASSWORD

IP Address Source    : DHCP Address
IP Address           : 192.168.1.72
Subnet Mask          : 255.255.255.0
MAC Address          : xx:xx:xx:xx:xx:xx
SNMP Community String : public
IP Header            : TTL=0x40 Flags=0x40 Precedence=0x00 TOS=0x10
Default Gateway IP   : 192.168.1.1
802.1q VLAN ID       : Disabled
802.1q VLAN Priority : 0
RMCP+ Cipher Suites  : 0,1,2,3
Cipher Suite Priv Max : 0000XXXXXXXXXX
                     : X=Cipher Suite Unused
                     : c=CALLBACK
                     : u=USER
                     : o=OPERATOR
                     : a=ADMIN
                     : 0=OEM
```

3.1.5.4. LAN Channel を有効にする

```
# ipmitool lan set <channel number> access on
```

以上で管理対象サーバの設定は終了です。

3.2. 管理 PC 設定

パッケージのインストールは管理者権限のあるユーザで行います。また、OS(本書の実行例では CentOS 5.2)がインストールされている事を前提としますので OS インストール方法に関しては説明を省略します。

なお、rpm パッケージ/コマンドを使用しないディストリビューション上でのインストールは、各ディストリビューションの仕様に従ってください。

3.2.1. IPMITool のインストール

```
# rpm -ivh OpenIPMI-tools-(version).rpm
```

以上で管理 PC の設定は終了です。

3.3. 管理対象サーバにおける BMC 設定初期化(EXPRESSSCOPE エンジン 2)

3.1 章で行った BMC のアカウントや LAN 設定を初期化する方法について説明します。



BMC の設定値が初期化されます。
本書 3.1 章でご紹介した以外の方法(NEC 提供の管理ツールなど)で
BMC 設定が行われていた場合も初期化されます。

- (1) Express サーバに添付されている EXPRESSBUILDER からブート
- (2) Tool Menu (Normal Mode) を選択

```
----- Boot selection -----
| Os installation *** default *** |
| Tool menu (Normal mode)         |
| Tool menu (Redirection mode)    |
|-----|
```

- (3) 言語(Japanese) を選択

```
----- Language selection -----
| Japanese *** default ***       |
| English                        |
| Return to previous menu        |
|-----|
```


- (4) TOOL MENU からシステムマネージメント を選択

TOOL MENU
Maintenance Utility
BIOS/FW Updating
ROM-DOS Startup FD
Test and Diagnostics
System Management
Return to previous menu

- (5) システムマネージメント機能から BMC の初期化 を選択

システムマネージメント機能 Rev. xx
ファームウェア管理情報の表示
システムマネージメントの設定
BMC web サーバの設定
LAN ポートの設定
BMC 設定の初期化
終了

- (6) BMC の初期化 を実行

ENTER を押下して、初期化を実行します。

BMC 設定の初期化
BMC 設定を初期化しますか?
実行: [ENTER] 中止: [ESC]

- (7) 初期化が終了したら、自動で再起動されます

- (8) EXPRESSBUILDER を取り出します

3.4. 管理対象サーバにおける BMC 設定初期化(EXPRESSSCOPE エンジン 3)

3.1 章で行った BMC のアカウントや LAN 設定を、初期化する方法について説明します。



注意

BMC の設定値が初期化されます。

本書 3.1 章でご紹介した以外の方法(NEC 提供の管理ツールなど)で BMC 設定が行われていた場合も初期化されます。初期化される項目については EXPRESSSCOPE エンジン 3 ユーザーズガイドを参照して下さい。

- (1) 装置を起動し、POST 中に「F4」キーを押下する。
- (2) POST 完了後に「Off-line TOOL MENU」が起動するので、「BMC Configuration」を選択して下さい。
- (3) 「BMC Initialization」を選択すると、BMC 設定の初期化を実行できます。

4. サーバ監視の活用例

4.1. IPMI 情報の表示

4.1.1. BMC 情報

ファームウェアのリビジョンや IPMI のバージョン等を確認することができます。

実行例

```
# ipmitool bmc info
Device ID                : 56                [BMC のデバイス ID]
Device Revision          : 1                [BMC デバイスのリビジョン]
Firmware Revision        : 1.4              [BMC ファームウェアのリビジョン]
IPMI Version             : 2.0              [IPMI のバージョン]
Manufacturer ID          : 119              [製造元 ID] ← 119 は NEC の ID
Manufacturer Name        : Unknown (0x77)   [製造元名称]
Product ID               : 1250 (0x04e2)     [プロダクト ID]
Device Available         : yes              [デバイスの状態]
Provides Device SDRs     : no
Additional Device Support :                 [サポートデバイス情報]
  Sensor Device
  SDR Repository Device
  SEL Device
  FRU Inventory Device
  IPMB Event Receiver
  Bridge
  Chassis Device
Aux Firmware Rev Info    :
  0x00                   [0x00 と 0x15 でブートブロックファームウェアの
  0x15                   レビジョン 00.15 を表す]
  0x01                   [0x01 と 0x02 で装置固有情報 (PIA) の
  0x02                   レビジョン 1.02 を表す]
```



[Manufacturer ID]および[Manufacturer Name]の表示について

[Manufacturer ID]には、IANA(Internet Assigned Number Authority)で定義された Enterprise Number が格納されます。

119 は NEC の Enterprise Number です。

(<http://www.iana.org/assignments/enterprise-numbers> 参照)

[Manufacturer Name]の表示が Unknown と表示される場合がありますが、IPMITool の問題である事が判明しています。

4.1.2. SDR 情報

センサーデータレコード (SDR) 情報にはハードウェアシステムのセンサに関する情報や保守交換ユニット (FRU) の種類、データ格納位置などの情報が定義されています。

センサには大きく分けて 2 つのクラスに分けられます。Threshold タイプのセンサは、閾値と比較することで現在の状態を示します。Discrete タイプのセンサは、自身の状態をビット列のオン/オフで表現し、同時に複数の状態を示すこともあります。

実行例 1: SDR のサマリ情報表示

```
# ipmitool sdr info
SDR Version           : 0x51           [SDR のフォーマットバージョン]
Record Count          : 51             [SDR の登録件数]
Free Space             : 49205 bytes    [SDR 領域の空き容量]
Most recent Addition   :               [最後に SDR を追加した日時]
Most recent Erase      :               [最後に SDR を削除した日時]
以降は省略
```

実行例 2: Threshold タイプのセンサ情報表示

```
# ipmitool sdr list full
Processor Vcore       | 0.99 Volts          | ok
Baseboard P_Vtt       | 1.07 Volts          | ok
以降は省略
```

※ それぞれ、1 列目から順に以下の状態を表します

- ◇ SDR に定義されている ID 文字列
 - ◇ センサの現在値
 - ◇ センサの正常/異常状態
- “ok”はセンサの状態が警告または危険を示す閾値に達していない事を示します。

実行例 3: Discrete タイプのセンサ情報表示

```
# ipmitool sdr list compact
Power Unit           | 0x00           | ok
BMC Watchdog         | 0x00           | ok
HDD 0 Status         | 0x01           | ok
以降は省略
```

※ それぞれ、1 列目から順に以下の状態を表します

- ◇ SDR に定義されている ID 文字列
 - ◇ bit 単位で定義されたセンサ固有の状態
- ”HDD 0 Status”に関しては、以下の通りセンサ状態が定義されており、bit 0 (Offset 01h) が 1 であるため、“Drive Present”である事を示します。Discrete タイプの各種センサ定義に関しては IPMI 仕様を参照してください(Appendix 参照)。
- センサタイプ (Drive Slot)
- Offset 00h : Drive Presence
 - Offset 01h : Drive Fault
 - Offset 02h : Predictive Failure
 - Offset 03h : Hot Spare
 - Offset 04h : Consistency Check / Parity Check in progress
 - Offset 05h : In Critical Array
 - Offset 06h : In Failed Array
 - Offset 07h : Rebuild/Remap in progress
 - Offset 08h : Rebuild/Remap Aborted
- ◇ センサの正常/異常状態
- “ok”と表示されていますが、Discrete タイプのセンサでは正しく反映されませんので無視して下さい(IPMITool の問題であることが判明しています)。

4.1.3. FRU 情報

保守交換ユニット(FRU)情報には、装置の筐体情報、ボード情報、製品情報等、詳細なコンポーネント情報が格納されています。

実行例

```
# ipmitool fru print
```

```
FRU Device Description : Builtin FRU Device (ID 0)
Chassis Type           : Rack Mount Chassis      [筐体種別]
Chassis Part Number    : 856-127810-001          [筐体の指定番号]
Chassis Serial         : 01
Chassis Extra          : 01
Board Mfg              : INVENTEC                [ボードの製造元]
Board Product          : 5021                    [ボードの製造元 ID]
Board Serial           : 1234567890              [ボードのシリアル番号]
Board Part Number      : 856-127810-001          [ボードの指定番号]
Product Manufacturer   : NEC                    [製品の製造元]
Product Name           : Express5800/iR110a-1H    [製品のモデル名]
Product Part Number    : [N8100-1488Y]          [製品の型番]
Product Version        : FR1.0                  [製品の FR 番号]
Product Serial         : 1234567                [号機番号]
Product Asset Tag      : 243-419170-001          [ユーザ資産番号]
Product Extra          :
```

4.1.4. SEL 情報

システムイベントログ(SEL)には、システムが持つさまざまなセンサが検出したイベント／エラーなど、後の障害復旧作業のために必要な情報が記録されています。

実行例 1: SEL の状態を確認

```
# ipmitool sel info
```

```
SEL Information
Version           : 1.5 (v1.5, v2 compliant) [SEL のフォーマットバージョン]
Entries           : 1482                     [SEL の登録件数]
Free Space        : 35600 bytes               [SEL 領域の空き容量]
Percent Used      : 39%                      [SEL の使用率]
Last Add Time     : 03/14/2009 01:12:27       [最後に SEL を追加した日時]
Last Del Time     : 03/10/2009 22:32:31       [最後に SEL を削除した日時]
Overflow          : false                    [SEL の空き容量が無い場合は true]
Supported Cmds    : 'Delete' 'Reserve' 'Get Alloc Info'
# of Alloc Units  : 3263
Alloc Unit Size   : 20
# Free Units      : 1780
Largest Free Blk  : 1780
Max Record Size   : 6
```

実行例 2: SEL の内容を一覧表示

```
# ipmitool sel list
```

```
140 | 03/28/2009 | 18:00:00 | Button #0x88 | Power Button pressed | Asserted  
以降省略
```

※ 各項目はそれぞれ以下の意味を表します

140	レコード ID
03/28/2009 18:00:00	タイムスタンプ
Button	センサタイプ
#0x88	センサ番号
Power Button pressed	イベント内容
Asserted	イベントが発生したことを示します
	Deasserted の場合は、発生したイベント要因が 取り除かれたことを示します



IPMITool で表示される SEL のタイムスタンプについて

Linux におけるハードウェアクロックの設定がローカルタイムになっている場合、IPMITool で表示される SEL のタイムスタンプは、ローカルタイムにタイムゾーンの値が付加されて表示されますので注意して下さい(タイムゾーンが日本(+9 時間)の場合、実際の日本時間よりも更に 9 時間進んだ時間となります)。本件は IPMITool の問題であることが判明しています。

実行例 3: SEL 情報の保存と消去

SEL の空き容量が無くなると、それ以降に発生したイベントを記録することが出来なくなります。このため、定期的に空き容量をチェックするなどして、SEL 情報を保存してから消去します。

Text 形式で保存する場合

```
# ipmitool sel save <text filename>
```

バイナリ形式で保存する場合

```
# ipmitool sel writeraw <binary filename>
```

SEL の消去

```
# ipmitool sel clear
```



EXPRESSSCOPE エンジン 2 搭載装置の場合はシステム BOIS のセットアップメニューから、EXPRESSSCOPE エンジン 3 搭載装置の場合は P EXPRESSSCOPE エンジン 3 の WebConsole またはオンライン・オフライン BMC Configuration Tool から、SEL の空き容量が無くなった場合の動作を変更できます。設定詳細については、装置のユーザズガイドまたは EXPRESSSCOPE エンジンユーザズガイドを参照してください。

4.2. センサ状態表示

SDR にて定義された各種センサ(温度、電圧、ファン、電源など)の状態を表示することができます。

実行例 1: センサー一覧表示

```
# ipmitool sensor list
Processor Vcore | 1.000 | Volts | ok | na | 0.637 | 0.666 | 1.421 | 1.490 | na
Baseboard P_Vtt | 1.068 | Volts | ok | na | 0.853 | 0.902 | 1.205 | 1.264 | na
Power Unit      | 0x0   | discrete| 0x0080| na | na | na | na | na | na
以降は省略
```

※ それぞれ、1 列目から順に以下の状態を表します

- ◇ センサの ID 文字列
 - ◇ センサの現在値
温度、電圧、ファン等の Threshold タイプのセンサではセンサの現在値を返す。
閾値を持たない Discrete タイプのセンサでは有効な値が表示されませんので無視して下さい。
 - ◇ 単位(閾値を持たない Discrete タイプのセンサの場合は discrete と表示される)
 - ◇ センサの状態
温度、電圧、ファン等の Threshold タイプのセンサでは現在値が警告または危険を示す閾値に達していない場合は ok を表示します。
閾値を持たない Discrete タイプのセンサでは bit 単位で定義されたセンサの状態を示します。"Power Unit"に関しては、以下の通りセンサ状態が定義されており、何れの bit (Offset) も 0 であるため、"Power Unit"に関しては異常が無い事を示します。
Discrete タイプの各種センサ定義に関しては IPMI 仕様 (Appendix 参照) を参照してください。
- センサタイプ (Power Unit)
- Offset 00h : Power Down / Power Off
 - Offset 01h : Power Cycle
 - Offset 02h : 240VA Power Down
 - Offset 03h : Interlock Power Down
 - Offset 04h : AC Lost
 - Offset 05h : Soft Power Control Failure
 - Offset 06h : Predictive Failure
- ◇ 回復不能状態の下限閾値 (na は設定されていないことを表す)
 - ◇ 危険状態の下限閾値
 - ◇ 警告状態の下限閾値
 - ◇ 警告状態の上限閾値
 - ◇ 危険状態の上限閾値
 - ◇ 回復不能な状態の上限閾値 (na は設定されていないことを表す)

実行例 2: 特定のセンサ情報のみ表示 (get オプションでセンサの ID 文字列を指定)

```
# ipmitool sensor get "System Fan 1"
Locating sensor record...
Sensor ID       : System Fan 1 (0x40)
Entity ID       : 7.1
Sensor Type (Analog) : Fan
Sensor Reading   : 5807.201 (+/- inf) RPM
Status          : ok
```

```

Lower Non-Recoverable : na
Lower Critical        : na
Lower Non-Critical    : na
Upper Non-Critical    : 4492.363
Upper Critical        : na
Upper Non-Recoverable : na
Assertion Events      :
Assertions Enabled     : unc+
Deassertions Enabled  : unc+

```



上記 FAN の場合は、SDR における Linearization 定義が $1/x$ となっています。従って、センサの生データに対応する Upper Non-Critical Threshold (警告状態の上限閾値) を実際の回転数に変換すると、実際は下限閾値となるため、表示上はセンサの現在値が警告の上限閾値よりも大きくなっているにもかかわらず Status 表示は ok となっています。

FAN でも Linearization が Linear のセンサと、 $1/x$ のセンサの 2 種類が存在します。

4.3. 電源制御

ipmitool は以下の電源制御を行うことができます。

コマンド	説明
chassis status	サーバ筐体の電源状態に関する情報を表示します
chassis poh	通電時間(Power-On Hours)カウンタを表示します
chassis identify [interval]	フロントパネルの識別用 UID (Unit ID) ランプの点滅指示を行います (デフォルトの Interval は 15 秒) UID ランプをオフにするには interval に 0 を指定します
chassis power status	装置の電源状態を表示します
chassis power on	装置の電源をオンします
chassis power off	装置の電源をオフします
chassis power cycle	一旦電源オフし直後に再度電源オンします
chassis power reset	ハードリセットを行います
chassis power diag	NMI を生成します (ダンプスイッチの押下をエミュレート)
chassis power soft	電源ボタンの押下をエミュレートします

実行例:

```
# ipmitool chassis status
```

```

System Power          : on           [電源状態] ← 電源オン状態
Power Overload        : false        [過電流による Shutdown] ← 該当無し
Power Interlock       : inactive     [インターロックでの Shutdown] ← 該当無し
Main Power Fault      : false        [電源故障] ← 異常無し
Power Control Fault   : false        [電源制御失敗] ← 異常無し
Power Restore Policy  : previous     [AC オフ/オン時の電源状態回復設定]
                                   ← AC オフ直前の状態を復帰

```


Last Power Event	:	
Chassis Intrusion	:	inactive [筐体カバーオープン]←該当無し
Front-Panel Lockout	:	inactive [電源ボタン等のマスク]←無効
Drive Fault	:	false [Drive 故障有無]←異常無し
Cooling/Fan Fault	:	false [冷却部故障有無]←異常無し
Sleep Button Disable	:	allowed [Sleep ボタン無効化設定]←可能
Diag Button Disable	:	allowed [NMI ボタン無効化設定]←可能
Reset Button Disable	:	allowed [Reset ボタン無効化設定]←可能
Power Button Disable	:	allowed [Power ボタン無効化設定]←可能
Sleep Button Disabled	:	true [スリープボタン無効化]
Diag Button Disabled	:	true [NMI ボタン無効化] (*注意)
Reset Button Disabled	:	true [リセットボタン無効化] (*注意)
Power Button Disabled	:	true [電源ボタン無効化] (*注意)



上記において Diag Button Disabled、Reset Button Disabled、Power Button Disabled に関しては正しい状態が表示されませんので無視して下さい。
本件は IPMITool の問題であることが判明しています。

```
# ipmitool chassis power status
```

```
Chassis Power is on
```

```
# ipmitool chassis poh
```

```
POH Counter : 1128 hours total (47 days, 0 hours)
```

4.4. 任意の IPMI コマンド実行

ipmitool は、BMC に対して任意の IPMI コマンドを実行することができます。

実行例: Get Device ID コマンドを実行する

```
# ipmitool -v raw 0x6 0x1
```

```
RAW REQ (channel=0x0 netfn=0x6 lun=0x0 cmd=0x1 data_len=0)
```

```
RAW RSP (15 bytes)
```

```
38 01 01 01 02 df 77 00 00 e2 04 00 12 01 02
```



IPMI コマンド詳細については、IPMI 仕様(Appendix 参照)を参照してください。

4.5. その他のコマンド

4.5.1. BMC の自己診断

実行例:

```
# ipmitool bmc selftest
```

```
Selftest: passed [BMC の自己診断が正常終了している事を示します]
```

5. ウォッチドッグタイマを使用した OS のストール監視／再起動

5.1. 機能概要

装置に搭載されている BMC のウォッチドッグタイマ機能 (IPMI 仕様に準拠) を使って、OS (ソフトウェア) のストール監視を行うことができます。

ユーザアプリケーションから定期的に BMC のウォッチドッグタイマを更新します。設定した時間を経過してタイマを更新できなかった場合は OS ストールと判断し、BMC は、設定された動作を実行し、システムの再起動などを行います。

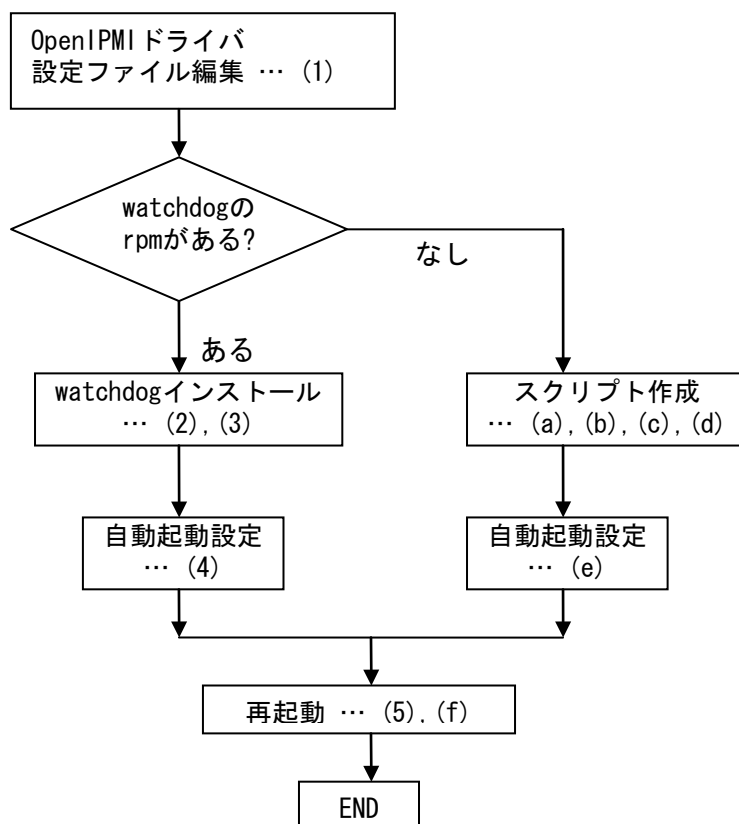


システムの負荷状況によっては、OS がストール状態でなくても、ウォッチドッグタイマの更新ができずにタイムアウトが発生する可能性があります。事前に高負荷状態での評価を行った上でウォッチドッグタイマの設定をして下さい。

5.2. 設定手順

管理対象サーバの設定を行います。

以下に、基本的な設定手順のフローを記します。



(1) OpenIPMI ドライバの設定ファイル編集

/etc/sysconfig/ipmi を以下のような設定値にします。

IPMI_WATCHDOG=yes

IPMI_WATCHDOG_OPTIONS="timeout=600 action=reset start_now=1"

※ IPMI_WATCHDOG_OPTIONS には以下の値を設定できます。

timeout : タイムアウト時間

ストールしたと判定するまでの時間を秒数で設定します。

デフォルト値(指定しない時)は 60 秒となります

最小は 10 秒から設定可能

action : タイムアウト発生後の動作

タイムアウトが発生した時の動作を設定します

設定可能な値

none : 何もしません

reset : システムをリセットし再起動を試みます

power_off : システムの DC をオフします

power_cycle : 一旦 DC オフし、再度 DC オンします

設定を行わない場合、[reset] の設定となります

start_now : 1 : OpenIPMI ドライバのロードと同時にタイマスタート

通常はこの設定で問題ありません



タイマの更新はユーザモードで行われるため、システムの負荷状況によっては更新が遅れることもあります。このためタイムアウト時間の設定値は、事前に評価をおこなったうえで、余裕のある時間を設定してください。

(2) タイマ更新を行うパッケージをインストール

```
# rpm -ivh watchdog-(version).rpm
```

(3) タイマ更新用の設定ファイル編集

/etc/watchdog.conf を以下のような設定値にします。

watchdog-device=/dev/watchdog

interval=30

※ interval に更新間隔を秒数で設定します。

この例では、30 秒毎にタイマ更新を行います。

(4) 自動起動設定

OpenIPMI ドライバ及び watchdog の起動設定を行います。

```
# chkconfig ipmi on
```

```
# chkconfig watchdog on
```

(5) システムを再起動

OS 起動時から OS ストール監視が有効になります

```
# reboot
```



watchdog のパッケージが含まれていないディストリビューションの場合、シェルスクリプトを作成することにより、タイマ更新を行うことができます。以下にスクリプトの作成例を示します。

(a) タイマを更新するプログラムを作成

この例ではファイル名を「ResetWDT」とします。

```
#!/bin/bash
while true
do
    /usr/bin/ipmitool raw 0x6 0x22 > /dev/null 2>&1
    sleep 30
done
```

※ sleep の秒数でタイマ更新間隔を設定します。

この例では更新間隔を 30 秒としています。

(b) プログラムを /usr/sbin 配下にコピー

```
# install -p -m 755 ResetWDT /usr/sbin
```

(c) 作成した更新用プログラムを、OS 起動時に自動起動させるスクリプトを作成

この例ではファイル名を「watchdog」とします。

```
#!/bin/bash
#
# chkconfig: - 27 46
# description: A software watchdog
#
# Source function library.
. /etc/rc.d/init.d/functions
prog=/usr/sbin/ResetWDT
case "$1" in
start)
    echo -n "Starting watchdog daemon: "
    ${prog} &
    success "Starting watchdog daemon"
    echo
    ;;
*)
    echo "Usage: watchdog {start}"
    exit 1
    ;;
esac
```

- (d) 上記起動用スクリプトを /etc/rc.d/init.d 配下にコピー

```
# install -p -m 755 watchdog /etc/rc.d/init.d
```

- (e) 起動用スクリプトを自動起動できるように設定

```
# chkconfig --add watchdog
```

```
# chkconfig watchdog on
```

- (f) システムを再起動

OS 起動時から OS ストール監視が有効になります

```
# reboot
```

5.3. 監視を無効化する手順

- (1) /etc/sysconfig/ipmi 編集

```
IPMI_WATCHDOG=no
```

- (2) 起動用スクリプトが自動起動しないように設定

```
# chkconfig watchdog off
```

- (3) システムを再起動

次の OS 起動から、ストール監視は無効になります。

```
# reboot
```

- (4) 不要な場合、スクリプトを削除

watchdog のパッケージをインストールしていた場合

```
# rpm -e watchdog
```

シェルスクリプトでタイマ更新していた場合

```
# chkconfig --del watchdog
```

```
# rm -f /etc/rc.d/init.d/watchdog
```

```
# rm -f /usr/sbin/ResetWDT
```

5.4. タイムアウト時に NMI を発生させる設定

BMC のウォッチドッグタイマで、タイムアウト検出時に NMI を発生させることができます。

システム上で NMI によるメモリダンプを採取できるような設定がおこなわれている場合に、ウォッチドッグタイマによる OS ストール検出を契機に NMI を発生させ、障害解析に役立つ情報を保存することができます。

OpenIPMI ドライバ の設定ファイル(/etc/sysconfig/ipmi)では NMI を発生させる設定を行うことができないため、IPMITool から直接 IPMI コマンドを発行することによって任意の設定を行います(4.5 章参照)。

本書では、以下の設定を行う手順について記載します

タイムアウト時間 600 秒
タイムアウト発生時の動作 NMI
タイムアウト発生後の動作 none (何もしない)



システム上で、以下の設定が行われていることを想定しています

- NMI によりメモリダンプを採取
- メモリダンプ後、自動的にシステムを再起動

メモリダンプの設定方法は、OS やカーネルのドキュメントを参照してください。
また、5.2 章でご紹介した設定とは共存できません



本章は BMC のウォッチドッグタイマを使用した一般的なメモリダンプの採取方法を
紹介するものであり、NEC にて本章の手順により採取したメモリダンプの解析、調査
サービスを提供するものではありません。

(1) OpenIPMI ドライバの設定ファイル編集

/etc/sysconfig/ipmi を以下の設定値にします。
IPMI_WATCHDOG=no

(2) タイマを更新するプログラムを作成

この例ではファイル名を「ResetWDT-NMI」とします。

```
#!/bin/bash
#
# プログラム起動時に Set Watchdog Timer コマンドを実行する
/usr/bin/ipmitool raw 0x6 0x24 0x4 0x20 0xa 0x3e 0x70 0x17 # ※1 参照
while true
do
    /usr/bin/ipmitool raw 0x6 0x22 > /dev/null 2>&1
    sleep 30
done
```

※1 Set Watchdog Timer コマンド実行時の ipmitool の引数について

raw ... IPMI コマンドを指定して実行する際の引数(固定)
0x6 ... NetFunction (固定)
0x24 ... Command (固定)
 NetFunction(0x6) と Command(0x24) の組合せで、
 Set Watchdog Timer コマンドを表します。
0x4 ... Timer Use
OS 動作中のストール監視の場合 0x4 から変更の必要はありません。
下位 3 ビットで、ストール監視のフェーズを表しています。

```

[2:0]
    000b = reserved
    001b = BIOS FRB2
    010b = BIOS/POST
    011b = OS Load
    100b = SMS/OS
    101b = OEM
    上記以外 = reserved (使用しません)
0x20 ... Timer Actions
    上位 4 ビットでタイムアウト発生時の動作設定を行います。
    [7] reserved
    [6:4] pre-timeout interrupt
        000b = none (何もしません)
        001b = SMI (使用しません)
        010b = NMI/Diagnostic Interrupt (NMI を発生します)
        011b = Messaging Interrupt (使用しません)
        上記以外 = reserved (使用しません)
    下位 4 ビットでタイムアウト発生後の動作設定を行います。
    [3] reserved
    [2:0] timeout action
        000b = no action (何もしません)
        001b = Hard Reset (リセットします)
        010b = Power Down (DC OFF します)
        011b = Power Cycle (DC OFF 後、DC ON します)
        上記以外 = reserved (使用しません)
0xa ... Pre-timeout interval
    タイムアウト検出からタイムアウト後の動作に移行するまでの
    時間を 1 秒単位で指定します。0xa の場合は 10 秒となります。
0x3e ... Timer Use Expiration flags clear
    通常は 0x3e のまま、変更の必要はありません。
0x70 ... Initial countdown value, lsbyte(100ms/count)
0x17 ... Initial countdown value, msbyte
    Initial countdown value で、カウントダウン時間を設定します。
    BMC のウォッチドッグタイマ機能は、1 count は 100 ミリ秒単位と
    なっているため、カウントダウン時間を 600 秒に設定する場合、
    600 x 10 = 6000 (10 進数) = 0x1770 (16 進数)
    lsbyte, msbyte の順に引数に指定するので 0x70 0x17 の順となる。

```



コマンド詳細については、IPMI 仕様(Appendix 参照)の”Set Watchdog Timer Command”の章を参照してください。

ヒント

(3) プログラムを /usr/sbin 配下にコピー

```
# install -p -m 755 ResetWDT-NMI /usr/sbin
```

- (4) 作成した更新用プログラムを、OS 起動時に自動起動させるスクリプトを作成

この例ではファイル名を「watchdog-nmi」とします。

```
#!/bin/bash
#
# chkconfig: - 27 46
# description: A software watchdog
#
# Source function library.
. /etc/rc.d/init.d/functions
prog=/usr/sbin/ResetWDT-NMI
case "$1" in
start)
    echo -n "Starting watchdog-nmi daemon: "
    ${prog} &
    success "Starting watchdog-nmi daemon"
    echo
    ;;
*)
    echo "Usage: watchdog-nmi {start}"
    exit 1
    ;;
esac
```

- (5) 上記起動用スクリプトを /etc/rc.d/init.d 配下にコピー

```
# install -p -m 755 watchdog-nmi /etc/rc.d/init.d
```

- (6) 起動用スクリプトを自動起動できるように設定

```
# chkconfig --add watchdog-nmi
# chkconfig watchdog-nmi on
```

- (7) システムを再起動

OS 起動時から OS ストール監視が有効になります

```
# reboot
```


6. Serial Over LAN 機能を使用する

Serial Over LAN(SOL)とは、システム BIOS またはコンソールレス対応 OS からシリアルポート 2 に出力されるリダイレクションデータを BMC が取得し、LAN 経由で送信することにより、LAN 経由のリモートコンソールを実現する機能です。

SOLを使用することにより、管理対象サーバの BIOS 設定をリモートから行えるほか、ログインすることにより管理対象サーバのコンソールとして利用することができます。



本機能使用時、管理対象サーバに標準実装されているシリアルポート 2 が使用不可となります。

SOL を使用するためには、管理対象サーバについて以下の作業を行う必要があります。

- システム BIOS の設定
- OS の設定ファイル編集
- GRUB の設定ファイル編集
- EXPRESSSCOPE エンジンの設定変更

6.1. システム BIOS 設定

管理対象サーバのシステム BIOS Setup メニューから、Console Redirection の項目を以下のよう
に設定します。

設定項目	設定値
BIOS Redirection Port	Serial Port B
Baud Rate	19.2k
Flow Control	CTS/RTS または Hardware RTS/CTS
Terminal Type	PC ANSI
Continue Redirection after POST または Continue C.R. after POST	Enabled
Remote Console Reset	Enabled



「ACPI Redirection Port」の項目がある場合は **[Disabled]** に設定して下さい。
項目が無い場合は変更の必要はありません。

DOS 上の日本語を表示する場合は、「Terminal Type」を **[PC ANSI]** に設定して
下さい。項目が無い場合は変更の必要はありません。



「Remote Console Reset」を [Enabled]に設定すると、管理 PC 上の端末で「Esc R Esc r Esc R」とキー入力することでシステムのリセットができるようになります。装置によっては本メニューが存在しない場合があります。

6.2. OS の設定ファイル編集

6.2.1. /etc/inittab 編集

設定ファイル /etc/inittab に以下の記述を追加します

```
S0:2345:respawn:/sbin/agetty -h -L 19200 ttyS1 vt100
```



「6.1 システムBIOSの設定」の設定値と同じにする必要があります。
上記の例では、agetty を以下の状態で使用します。

- ・ ハードウェア制御(RTS/CTS)を有効にする。
- ・ キャリア検知の必要がないローカルな直結回線であるとみなす。
- ・ ボーレート 19200bps
- ・ シリアルポート2を使用する。
- ・ 端末タイプを vt100 とする。

6.2.2. /etc/securetty 編集

スーパーユーザ(root)としてログインするためには、管理対象サーバのシリアルポート 2 のセキュリティを解除する必要があります。十分に注意して運用してください。

シリアルポート 2 のセキュリティを解除するためには、/etc/securetty ファイルに「ttyS1」を追加してください。

6.3. GRUB の設定ファイル編集

管理対象サーバの /boot/grub/grub.conf を編集します。

- (1) splashimage で始まる行をコメントアウト(行頭に#を追加)します。

```
#splashimage=....
```

- (2) kernel で始まる行に以下のカーネルパラメータの設定を追加します。

```
console=tty0 console=ttyS1,19200n8r
```

grub.conf 設定例

```
#boot=/dev/sda
default=0
timeout=5
#splashimage=(hd0,0)/boot/grub/splash.xpm.gz      ..... (1) コメントアウト
hiddenmenu
title CentOS (2.6.18-92.el5)
    root (hd0,0)
    kernel /boot/vmlinuz-2.6.18-92.el5 ro root=LABEL=/ rhgb console=tty0
console=ttyS1,19200n8r      ..... (2) console 以降を追加
    initrd /boot/initrd-2.6.18-92.el5.img
```

6.4. EXPRESSSCOPE エンジンの設定

EXPRESSSCOPE エンジンの設定変更も必要です。

- ・IPMITool を利用する場合(EXPRESSSCOPE エンジン 1/2/3 共通)

以下のコマンドを入力します。

```
# ipmitool -I lanplus -H <IP Address> -U <username> -P <password> sol set enabled
true 2
```

※ IP Address/username/password は、管理対象サーバの設定(3.1 章参照)で登録するものを使用します。



IPMIToolを利用しない場合、以下の操作で同様の設定が可能です。

・EXPRESSSCOPEエンジン1/2の場合

ESMPRO/ServerAgent ExtensionをOS上にインストールし、以下の設定を有効化します。

【コンフィグレーション】→【コンフィグレーション情報設定】
→【共通】→【リダイレクション(LAN)】

・EXPRESSSCOPEエンジン3の場合

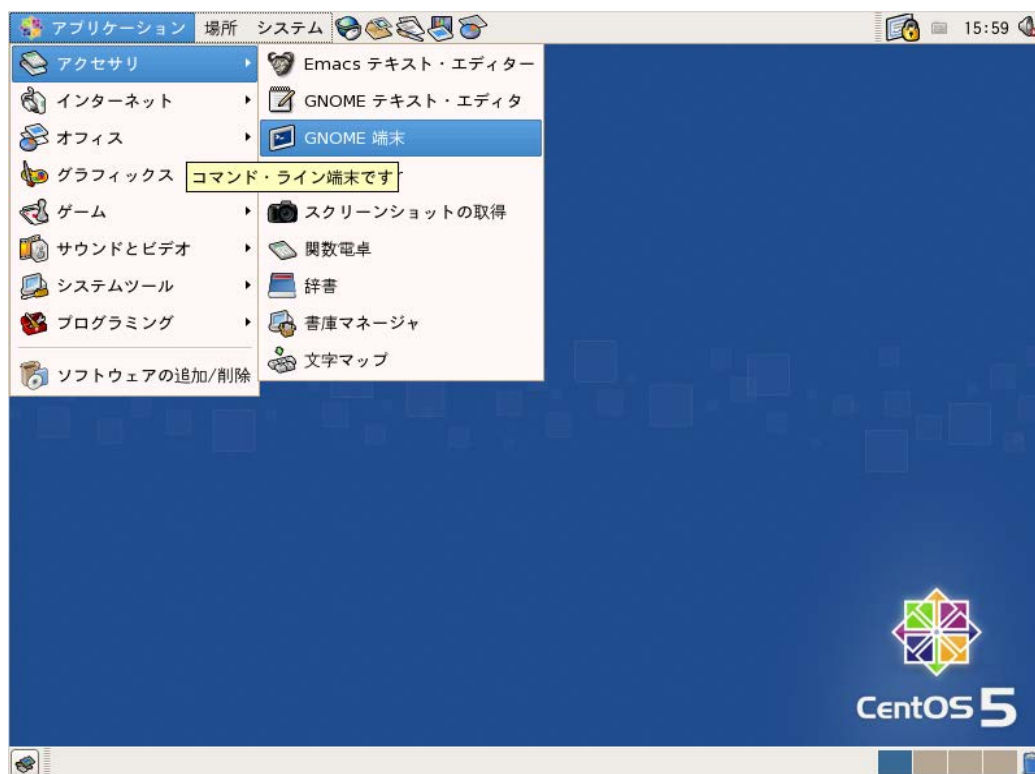
EXPRESSSCOPEエンジンWebConsole画面、オンライン/オフライン BMC Configuration Toolで以下の設定を有効化します。

【設定タブ】→【管理ソフトウェア設定】→【リダイレクション(LAN)】

6.5. SOL を使用する

管理 PC の端末からコマンドを実行します。今回の確認では GNOME 端末を使用しています。

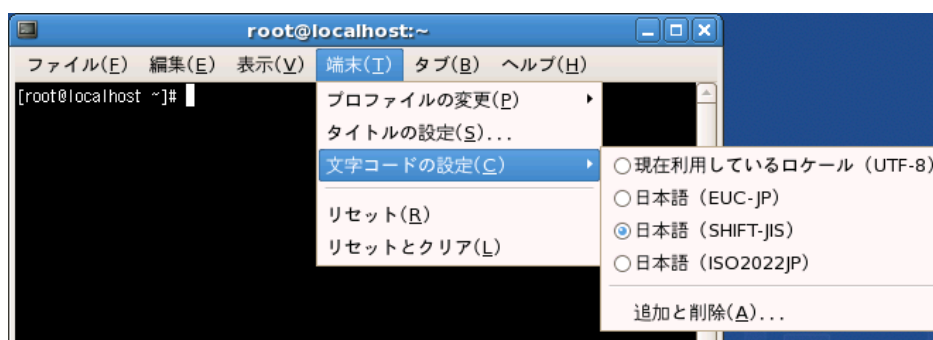
GNOME 端末を起動



端末上で日本語を表示させる場合、管理対象サーバの環境(locale)と端末の文字コードを合わせる必要があります。

管理対象サーバ上の DOS ユーティリティの日本語を表示させる場合は、端末の文字コードを Shift_JIS に設定します。

GNOME 端末の文字コードを Shift_JIS に設定する場合の例

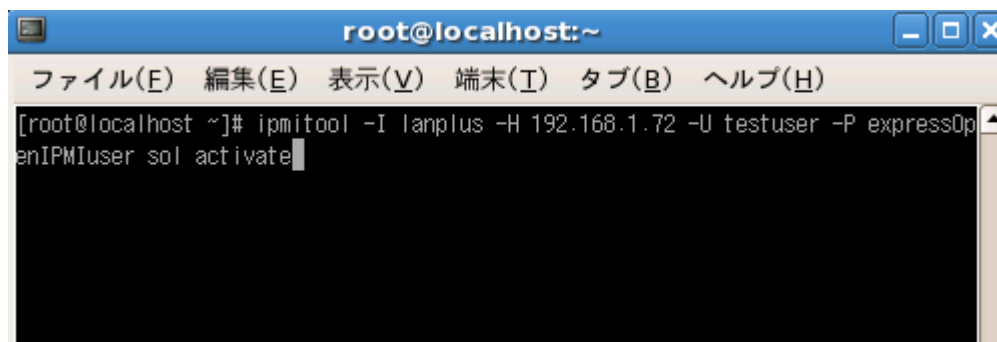


以下のコマンドを実行します。

```
# ipmitool -I lanplus -H <IP Address> -U <username> -P <password> sol activate
```

※ IP Address/username/password は、管理対象サーバの設定(3.1 章参照)で登録するものを使用します。

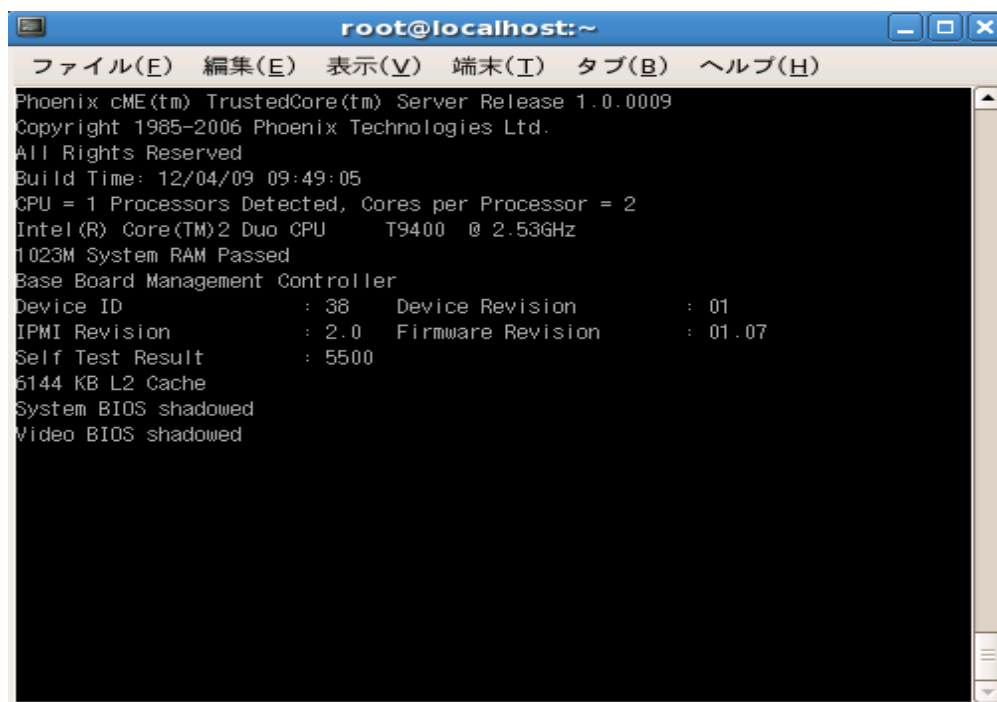
コマンド入力例



```
root@localhost:~  
ファイル(E) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)  
[root@localhost ~]# ipmitool -I lanplus -H 192.168.1.72 -U testuser -P expressOp  
enIPMIuser sol activate
```

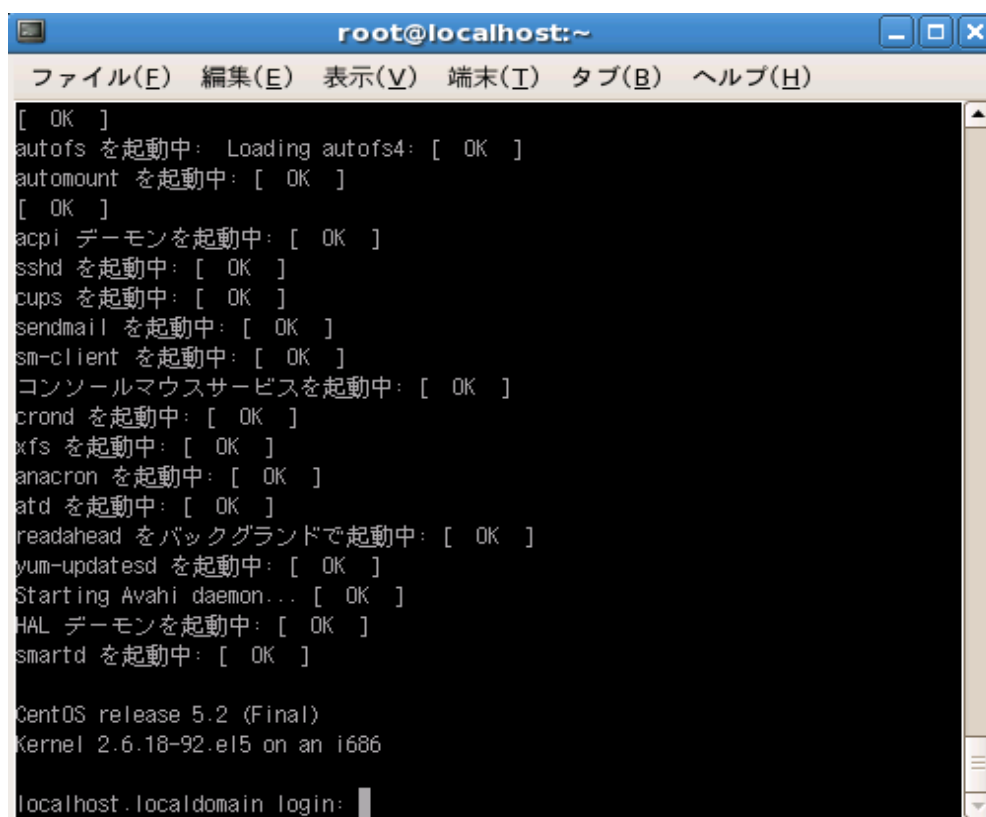
管理対象サーバの再起動を行います。SOL が正常に機能していれば、端末上に管理対象サーバの POST 画面や GRUB のメニュー画面が出力され、ログインプロンプトが表示されます。

POST 画面例



```
root@localhost:~  
ファイル(E) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)  
Phoenix cME(tm) TrustedCore(tm) Server Release 1.0.0009  
Copyright 1985-2006 Phoenix Technologies Ltd.  
All Rights Reserved  
Build Time: 12/04/09 09:49:05  
CPU = 1 Processors Detected, Cores per Processor = 2  
Intel(R) Core(TM)2 Duo CPU T9400 @ 2.53GHz  
1023M System RAM Passed  
Base Board Management Controller  
Device ID : 38 Device Revision : 01  
IPMI Revision : 2.0 Firmware Revision : 01.07  
Self Test Result : 5500  
6144 KB L2 Cache  
System BIOS shadowed  
Video BIOS shadowed
```

ログインプロンプト表示例（端末の文字コードは UTF-8 に設定しています）



A terminal window titled 'root@localhost:~' with a menu bar containing 'ファイル(E)', '編集(E)', '表示(V)', '端末(T)', 'タブ(B)', and 'ヘルプ(H)'. The terminal output shows the following sequence of messages:

```
[ OK ]
autofs を起動中: Loading autofs4: [ OK ]
automount を起動中: [ OK ]
[ OK ]
acpi デーモンを起動中: [ OK ]
sshd を起動中: [ OK ]
cups を起動中: [ OK ]
sendmail を起動中: [ OK ]
sm-client を起動中: [ OK ]
コンソールマウスサービスを起動中: [ OK ]
crond を起動中: [ OK ]
xfs を起動中: [ OK ]
anacron を起動中: [ OK ]
atd を起動中: [ OK ]
readahead をバックグラウンドで起動中: [ OK ]
yum-updatesd を起動中: [ OK ]
Starting Avahi daemon... [ OK ]
HAL デーモンを起動中: [ OK ]
smartd を起動中: [ OK ]

CentOS release 5.2 (Final)
Kernel 2.6.18-92.el5 on an i686

localhost.localdomain login: |
```

6.6. SOL を終了する

SOL を使用している端末上で「~.」とキー入力します。

■ Appendix

参考文献

- ◆ Express5800 シリーズポータルサイト
<http://www.nec.co.jp/products/express/>
- ◆ IPMI 仕様
<http://www.intel.com/design/servers/ipmi/>
- ◆ IPMITool
<http://sourceforge.net/projects/ipmitool>
- ◆ OpenIPMI
<http://sourceforge.net/projects/openipmi/>
- ◆ OpenIPMI ドライバを使用したウォッチドッグタイマ制御
<http://sourceforge.net/projects/watchdog>