

FutureNet NXR,WXR シリーズ

設定例集

運用管理編

Ver 1.4.0

センチュリー・システムズ株式会社



目次

目次.....	2
はじめに	4
改版履歴	5
1. ファームウェアの更新	6
1-1. CLI からのファームウェア更新	7
1-2. 起動時のファームウェア指定	17
2. 設定の保存・復帰	19
2-1. 本体への設定保存 1	20
2-2. 本体への設定保存 2(拡張フラッシュの利用).....	21
2-3. 設定のバックアップ 1.....	23
2-4. 設定のバックアップ 2(拡張フラッシュの利用).....	27
2-5. 設定の復帰 1	32
2-6. 設定の復帰 2(拡張フラッシュの利用).....	34
2-7. 起動時の設定ファイル指定.....	37
2-8. 本体内の設定の削除.....	38
2-9. 設定の初期化.....	39
3. パケットダンプ	40
3-1. パケットダンプを pcap 形式で取得する	41
3-2. パケットダンプの外部転送.....	42
3-3. パケットダンプの削除.....	44
4. スケジュール設定.....	45
4-1. PPP/PPPoE を定期的に接続/切断する	46
4-2. モバイル通信端末を定期的にリセットする	47
4-3. 機器本体を定期的に再起動する.....	48
4-4. シスログを定期的にローテートする	49
4-5. ファームウェアの更新日時を指定する.....	50
4-6. ファームウェアの更新を自動化する	51
4-7. 指定した時間に Wake ON LAN(WOL)で端末を起動する	52
4-8. システムのスリープ/レジュームの日時を指定する.....	53
4-9. モニタログを定期的にローテートする.....	54
4-10. 日時指定例	55
5. シスログ設定.....	57

5-1. シスログの取得.....	58
5-2. シスログサーバへの転送.....	59
5-3. シスログのローテート 1(外部ストレージの利用).....	60
5-4. シスログのローテート 2(内蔵フラッシュの利用).....	61
5-5. メモリ上のシスログの外部転送.....	62
5-6. 内蔵フラッシュ上のシスログの外部転送.....	64
5-7. 機器再起動時のシスログバックアップ.....	66
6. SNMP 設定.....	67
6-1. SNMP エージェント設定.....	68
7. 時刻同期設定.....	72
7-1. NTP 設定.....	73
7-2. WIN 通信モジュールを利用した時刻同期設定.....	77
8. メール送信設定.....	79
8-1. PPP 接続/切断時のメール送信設定.....	80
8-2. ログメール送信設定.....	84
9. Wake ON LAN 設定.....	88
9-1. Wake ON LAN による端末起動.....	89
9-2. VPN 経由での Wake ON LAN.....	91
10. 省電力設定.....	103
10-1. 電源管理モード設定.....	104
10-2. スリープモード設定.....	105
11. テクニカルサポート情報・シスログの取得.....	106
11-1. テクニカルサポート情報の表示とファイル転送.....	107
11-2. シスログ情報の表示とファイル転送.....	110
12. モニタログ設定.....	113
12-1. モニタログ(監視結果の保存).....	114
12-2. モニタログファイルの表示とファイル転送.....	116
付録.....	119
設定例 show config 形式サンプル.....	120
サポートデスクへのお問い合わせ.....	129
サポートデスクへのお問い合わせに関して.....	130
サポートデスクのご利用に関して.....	132

はじめに

- FutureNet はセンチュリー・システムズ株式会社の登録商標です。
- 本書に記載されている会社名,製品名は、各社の商標および登録商標です。
- 本ガイドは、以下の FutureNet NXR,WXR 製品に対応しております。
NXR-120/C,NXR-125/CX,NXR-130/C,NXR-155/C シリーズ,
NXR-230/C,NXR-350/C,NXR-1200,NXR-G100 シリーズ,WXR-250
- 本書の内容の一部または全部を無断で転載することを禁止しています。
- 本書の内容については、将来予告なしに変更することがあります。
- 本書の内容については万全を期しておりますが、ご不審な点や誤り、記載漏れ等お気づきの点がありましたらお手数ですが、ご一報下さいますようお願い致します。
- 本書は FutureNet NXR-120/C, NXR-155/C-XW,NXR-230/C,NXR-G100 シリーズの以下のバージョンをベースに作成しております。

第 1 章～第 8 章 FutureNet NXR-120/C Ver5.24.1G

※1-1 の一部 FutureNet NXR-G100 Ver6.1.2,6.1.3

2-2,2-4,2-6,2-7,2-8 は FutureNet NXR-230/C Ver5.25.2

4-6 は FutureNet NXR-G100 Ver6.1.2,6.1.3

4-7,8 は FutureNet NXR-G100/KL Ver6.4.3

4-9 は FutureNet NXR-G100 Ver6.6.5

5-3～5-7 は FutureNet NXR-G100 Ver6.7.4

7-2 は FutureNet NXR-155/C-XW Ver5.22.5D

8-1 は FutureNet NXR-120/C Ver5.24.1J

8-2 は FutureNet NXR-G100 Ver6.1.3

第 9 章～第 10 章 FutureNet NXR-G100/KL Ver6.4.3

第 11 章～第 12 章 FutureNet NXR-G100 Ver6.6.5

各種機能において、ご使用されている製品およびファームウェアのバージョンによっては一部機能、コマンドおよび設定画面が異なっている場合もありますので、その場合は各製品のユーザーズガイドを参考に適宜読みかえてご参照および設定を行って下さい。

- 本バージョンでは IPv4 のみを対象とし、IPv6 設定については本バージョンでは記載しておりません。
- 設定した内容の復帰(流し込み)を行う場合は、CLI では「copy」コマンド、GUI では設定の復帰を行う必要があります。
- モバイルデータ通信端末をご利用頂く場合で契約内容が従量制またはそれに準ずる場合、大量のデータ通信を行うと利用料が高額になりますので、ご注意ください。
- 本書を利用し運用した結果発生した問題に関しましては、責任を負いかねますのでご了承下さい。

改版履歴

Version	更新内容
1.0.0	初版
1.1.0	NXR-G100 シリーズに対応 ファームウェア更新の自動化を追加(NXR-G100 シリーズのみ) メール送信設定を追加
1.2.0	Wake ON LAN 設定(スケジュールを含む)を追加 省電力設定(スケジュールを含む)を追加
1.3.0	テクニカルサポート情報・シスログの取得を追加 モニタログ設定(スケジュールを含む)を追加
1.4.0	シスログ設定の更新・追加(ローテート,外部転送,機器再起動時のバックアップ)

1. ファームウェアの更新

1-1. CLI からのファームウェア更新

1-2. 起動時のファームウェア指定

1-1. CLI からのファームウェア更新

この設定例では、CLI(コマンドラインインタフェース)を利用して、FTP サーバや SSH サーバといったリモートサーバや USB メモリなどの外部ストレージ内のファームウェアを転送し、ファームウェアを更新する方法について説明します。GUI でのファームウェア更新方法は、ユーザーズガイド(GUI 編)をご参照ください。GUI でのファームウェア更新方法はユーザーズガイド(GUI 編)をご参照ください。

この更新方法を利用する場合は、あらかじめリモートサーバや外部ストレージ上に更新したいファームウェアを置いておく必要があります。

なお、ファームウェアは弊社ホームページよりダウンロードできます。

<http://www.centurysys.co.jp/downloads/router/index.html>

ファームウェアの更新時間は機種により異なりますが、数分程度かかります。なお、ファームウェアの更新中に機器の電源断が発生した場合、故障の原因となりますのでご注意ください。

【 実行例 】

〔FTP サーバの利用〕

1. <ファームウェアのダウンロード>

```
#firmware update ftp://192.168.10.100/nxr120-v5241G.bin  
[-----] 11% DOWNLOAD
```

FTP サーバ 192.168.10.100 上にあるファームウェア(この実行例では nxr120-v5241G.bin)をダウンロードします。なお、FTP サーバを利用した更新を行う場合、anonymous による接続方法のみ対応していません。よって、ユーザ名やパスワードを指定することはできません。コマンド実行後、ファームウェアのダウンロードを開始します。

(☞) サーバ内でファームウェアをディレクトリ毎に管理している場合などは、下記のように IP アドレスの後に「ディレクトリ名/ファームウェア名」を指定します。

```
#firmware update ftp://192.168.10.100/nxr120/nxr120-v5241G.bin
```

(☞) 送信元 IP アドレスを指定することも可能です。通常は出力インタフェースの IP アドレスを送信元 IP アドレスとしますが、冗長化している場合など送信元 IP アドレスを固定したい場合に有効です。

```
#firmware update ftp://192.168.10.100/nxr120-v5241G.bin source 192.168.20.1
```

(☞) NXR-230/C,NXR-350/C,NXR-G100 シリーズ,WXR-250 では、ファームウェアを 2 つ持つことが可能です。通常ファームウェアの更新を行った場合、更新したファームウェアで起動しますが、no-boot オプションを指定することにより更新したファームウェアで起動しないよう設定することが可能です。

```
#firmware update ftp://192.168.10.100/nxr230-v5252.bin no-boot
```

2. <ファームウェアの更新>

ファームウェアのダウンロード完了後、アップデートの実行について問われますが、ここで指定可能な動作

は機種により異なります。

<NXR-120/C の場合>

```
[=====] 100% DOWNLOAD
Proceed with update? [y/n]:
```

指定	動作
y	ルータのサービスを停止してファームウェアの更新を開始し、更新後、機器の再起動を行います
n	ファームウェアの更新を中止します

<NXR-125/CX,NXR-155 シリーズの場合>

```
[=====] 100% DOWNLOAD
Proceed with update? [(y)es/(b)ackground/(r)eserve/(n)o]:
```

指定	動作
y	ルータのサービスを停止してファームウェアの更新を開始し、更新後、機器の再起動を行います
b	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動を行います
r	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動は行いません (☞) 更新したファームウェアを利用するためには、機器の再起動が必要です
n	ファームウェアの更新を中止します

<NXR-130/C の場合>

```
[=====] 100% DOWNLOAD
Proceed with update? [(y)es/(b)ackground/(n)o]:
```

指定	動作
y	ルータのサービスを停止してファームウェアの更新を開始し、更新後、機器の再起動を行います
b	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動を行います
n	ファームウェアの更新を中止します

<NXR-230/C,NXR-350/C,NXR-G100 シリーズ,WXR-250 の場合>

```
[=====] 100% DOWNLOAD
Proceed with update? [(y)es/(r)eserve/(n)o]:
```

指定	動作
y	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動を行います (☞) サービスを停止してファームウェアの更新を行うことができません
r	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動は行いません (☞) 更新したファームウェアを利用するためには、機器の再起動が必要です
n	ファームウェアの更新を中止します

<NXR-1200 の場合>

```
[=====] DOWNLOAD is being executed
Proceed with update? [(y)es/(b)ackground/(r)eserve/(n)o]:
```

指定	動作
y	ルータのサービスを停止してファームウェアの更新を開始し、更新後、機器の再起動を行います
b	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動を行います
r	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動は行いません (☞) 更新したファームウェアを利用するためには、機器の再起動が必要です
n	ファームウェアの更新を中止します

<補足>

```
Unsaved configuration changes exist. Save Flash? [y/n]:
```

ファームウェア更新時に設定を保存していない場合は、保存するかどうか問い合わせるメッセージが表示されます。保存する場合は「y」、保存しない場合は「n」を指定して実行します。

```
[=====] 100% DOWNLOAD
Proceed with update? [y/n]: y
% Timed out.
```

ファームウェアダウンロード後、3分以内にファームウェアのアップデートを実行しなかった場合（「Proceed with update?」が表示されてから3分以内に n 以外を選択しなかった場合）、転送したファームウェアを破棄します。この場合、再度 firmware update コマンドを実行してファイル転送から行う必要があります。

また、ルータのサービスを継続した状態でファームウェアの更新を行った場合、下記操作はできません。

- ・ CLI/GUI からの設定初期化
- ・ CLI/GUI/CMS からの機器再起動
- ・ CLI/GUI/CMS からのファームウェア更新
- ・ GUI/CMS からの設定復帰
- ・ CLI からの設定の復帰/保存
- ・ GUI からの設定(GUIでの設定時には、必ず本体のフラッシュメモリに設定の保存が行われるため)

【SSH サーバの利用】

1. <ファームウェアのダウンロード>

```
#firmware update ssh://test@192.168.10.100/nxr120-v5241G.bin
test@192.168.10.100's password:[パスワード]
[=====] 11% DOWNLOAD
```

SSH サーバ 192.168.10.100 上にあるファームウェア(この実行例では nxr120-v5241G.bin)をダウンロードします。この時 SSH サーバの IP アドレスとともにログイン用のユーザ名を合わせて指定します。この実行例では、ユーザ名を test とします。コマンド実行後、SSH サーバへのログイン用パスワードの入力を求められますので、パスワードを入力します。ログイン後、ファームウェアのダウンロードを開始します。

- (☞) サーバ内でファームウェアをディレクトリ毎に管理している場合などは、下記のように IP アドレスの後に「ディレクトリ名/ファームウェア名」を指定します。

```
#firmware update ssh://test@192.168.10.100/nxr120/nxr120-v5241G.bin
```

- (☞) 送信元 IP アドレスを指定することも可能です。通常は出力インタフェースの IP アドレスを送信元 IP アドレスとしますが、冗長化している場合など送信元 IP アドレスを固定したい場合に有効です。

```
#firmware update ssh://test@192.168.10.100/nxr120-v5241G.bin source 192.168.20.1
```

- (☞) NXR-230/C,NXR-350/C,NXR-G100 シリーズ,WXR-250 では、ファームウェアを 2 つ持つことが可能です。通常ファームウェアの更新を行った場合、更新したファームウェアで起動しますが、no-boot オプションを指定することにより更新したファームウェアで起動しないよう設定することが可能です。

```
#firmware update ssh://test@192.168.10.100/nxr230-v5252.bin no-boot
```

2. <ファームウェアの更新>

ファームウェアのダウンロード完了後、アップデートの実行について問われますが、ここで指定可能な動作は機種により異なります。

<NXR-120/C の場合>

```
[=====] 100% DOWNLOAD  
Proceed with update? [y/n]:
```

指定	動作
y	ルータのサービスを停止してファームウェアの更新を開始し、更新後、機器の再起動を行います
n	ファームウェアの更新を中止します

<NXR-125/CX,NXR-155 シリーズの場合>

```
[=====] 100% DOWNLOAD  
Proceed with update? [(y)es/(b)ackground/(r)eserve/(n)o]:
```

指定	動作
y	ルータのサービスを停止してファームウェアの更新を開始し、更新後、機器の再起動を行います
b	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動を行います
r	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動は行いません (☞) 更新したファームウェアを利用するためには、機器の再起動が必要です
n	ファームウェアの更新を中止します

<NXR-130/C の場合>

```
[=====] 100% DOWNLOAD
Proceed with update? [(y)es/(b)ackground/(n)o]:
```

指定	動作
y	ルータのサービスを停止してファームウェアの更新を開始し、更新後、機器の再起動を行います
b	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動を行います
n	ファームウェアの更新を中止します

<NXR-230/C,NXR-350/C,NXR-G100 シリーズ,WXR-250 の場合>

```
[=====] 100% DOWNLOAD
Proceed with update? [(y)es/(r)eserve/(n)o]:
```

指定	動作
y	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動を行います (☒) サービスを停止してファームウェアの更新を行うことができません
r	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動は行いません (☒) 更新したファームウェアを利用するためには、機器の再起動が必要です
n	ファームウェアの更新を中止します

<NXR-1200 の場合>

```
[===== DOWNLOAD is being executed
Proceed with update? [(y)es/(b)ackground/(r)eserve/(n)o]:
```

指定	動作
y	ルータのサービスを停止してファームウェアの更新を開始し、更新後、機器の再起動を行います
b	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動を行います
r	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動は行いません (☒) 更新したファームウェアを利用するためには、機器の再起動が必要です
n	ファームウェアの更新を中止します

<補足>

```
Unsaved configuration changes exist. Save Flash? [y/n]:
```

ファームウェア更新時に設定を保存していない場合は、保存するかどうか問い合わせるメッセージが表示されます。保存する場合は「y」、保存しない場合は「n」を指定して実行します。

```
[=====] 100% DOWNLOAD
Proceed with update? [y/n]: y
% Timed out.
```

ファームウェアダウンロード後、3分以内にファームウェアのアップデートを実行しなかった場合（「Proceed with update?」が表示されてから3分以内にn以外を選択しなかった場合）、転送したファームウェアを破棄します。この場合、再度 firmware update コマンドを実行してファイル転送から行う必要

があります。

また、ルータのサービスを継続した状態でファームウェアの更新を行った場合、下記操作はできません。

- CLI/GUI からの設定初期化
- CLI/GUI/CMS からの機器再起動
- CLI/GUI/CMS からのファームウェア更新
- GUI/CMS からの設定復帰
- CLI からの設定の復帰/保存
- GUI からの設定(GUIでの設定時には、必ず本体のフラッシュメモリに設定の保存が行われるため)

【外部ストレージの利用】

1. <ファームウェアの確認>

```
#dir
/usb:
total 30212
-rwxr-xr-x 1 root root 30933312 Mar  5 01:24 nxr120-v5241G.bin
```

USB メモリなどの外部ストレージに更新したいファームウェアが入っているかどうか確認します。

- (☞) USB ポートが 1 ポートの機種で USB に接続した場合、外部ストレージを disk0 と認識します。また USB ポートが 2 ポートの機種で USB0 に接続した場合、外部ストレージを disk0、USB1 に接続した場合、外部ストレージを disk1 と認識します。

```
#dir disk0
/usb0:
total 30212
-rwxr-xr-x 1 root root 30933312 Jan 31 08:28 nxr125-v5252.bin
```

```
#dir disk1
/usb1:
total 30212
-rwxr-xr-x 1 root root 30933312 Jan 31 08:28 nxr125-v5252.bin
```

2. <ファームウェアのダウンロード>

```
#firmware update disk0:nxr120-v5241G.bin
[=====] 11% DECODE
```

disk0 と認識されている外部ストレージ上にあるファームウェア(この実行例では nxr120-v5241G.bin)を転送します。コマンド実行後、ファームウェアのデコードを開始します。

- (☞) ストレージ内でファームウェアをディレクトリ毎に管理している場合などは、下記のように「ディレクトリ名/ファームウェア名」を指定します。

```
#firmware update disk0:nxr120/nxr120-v5241G.bin
```

- (☞) NXR-230/C,NXR-350/C,NXR-G100 シリーズ,WXR-250 では、ファームウェアを 2 つ持つことが可能です。通常ファームウェアの更新を行った場合、更新したファームウェアで起動しますが、no-boot オプションを指定することにより更新したファームウェアで起動しないよう設定することが可能です。

```
#firmware update disk0:nxr230-v5252.bin no-boot
```

3. <ファームウェアの更新>

ファームウェアのダウンロード完了後、アップデートの実行について問われますが、ここで指定可能な動作は機種により異なります。

<NXR-120/C の場合>

```
[=====] 100% DECODE
Proceed with update? [y/n]:
```

指定	動作
y	ルータのサービスを停止してファームウェアの更新を開始し、更新後、機器の再起動を行います
n	ファームウェアの更新を中止します

<NXR-125/CX,NXR-155 シリーズの場合>

```
[=====] 100% DECODE
Proceed with update? [(y)es/(b)ackground/(r)eserve/(n)o]:
```

指定	動作
y	ルータのサービスを停止してファームウェアの更新を開始し、更新後、機器の再起動を行います
b	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動を行います
r	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動は行いません (☞) 更新したファームウェアを利用するためには、機器の再起動が必要です
n	ファームウェアの更新を中止します

<NXR-130/C の場合>

```
[=====] 100% DECODE
Proceed with update? [(y)es/(b)ackground/(n)o]:
```

指定	動作
y	ルータのサービスを停止してファームウェアの更新を開始し、更新後、機器の再起動を行います
b	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動を行います
n	ファームウェアの更新を中止します

<NXR-230/C,NXR-350/C,NXR-G100 シリーズ,WXR-250 の場合>

```
[=====] 100% DECODE
Proceed with update? [(y)es/(r)eserve/(n)o]:
```

指定	動作
y	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動を行います (☑) サービスを停止してファームウェアの更新を行うことができません
r	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動は行いません (☑) 更新したファームウェアを利用するためには、機器の再起動が必要です
n	ファームウェアの更新を中止します

<NXR-1200 の場合>

```
[===== DECODE is being executed]
Proceed with update? [(y)es/(b)ackground/(r)eserve/(n)o]:
```

指定	動作
y	ルータのサービスを停止してファームウェアの更新を開始し、更新後、機器の再起動を行います
b	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動を行います
r	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動は行いません (☑) 更新したファームウェアを利用するためには、機器の再起動が必要です
n	ファームウェアの更新を中止します

<補足>

```
Unsaved configuration changes exist. Save Flash? [y/n]:
```

ファームウェア更新時に設定を保存していない場合は、保存するかどうか問い合わせるメッセージが表示されます。保存する場合は「y」、保存しない場合は「n」を指定して実行します。

```
[=====] 100% DOWNLOAD
Proceed with update? [y/n]: y
% Timed out.
```

ファームウェア転送後、3分以内にファームウェアのアップデートを実行しなかった場合(「Proceed with update?」が表示されてから3分以内にn以外を選択しなかった場合)、転送したファームウェアを破棄します。この場合、再度 firmware update コマンドを実行してファイル転送から行う必要があります。また、ルータのサービスを継続した状態でファームウェアの更新を行った場合、下記操作はできません。

- CLI/GUI からの設定初期化
- CLI/GUI/CMS からの機器再起動
- CLI/GUI/CMS からのファームウェア更新
- GUI/CMS からの設定復帰
- CLI からの設定の復帰/保存
- GUI からの設定(GUIでの設定時には、必ず本体のフラッシュメモリに設定の保存が行われるため)

【オフィシャルの利用】

NXR-G100 シリーズでは、弊社ホームページから最新のファームウェア情報を取得し、更新することが可能です。

1. <最新のファームウェア情報の確認>

```
#firmware check official
Update Available :
Century Systems NXR-G100 ver 6.1.3 (build 4/14:05 09 06 2014)
```

弊社ホームページに最新のファームウェアがあるか確認します。なお、フラッシュ上のファームウェアとバージョンが異なる場合、更新情報を表示します(ファームウェアが2面ある場合は、どちらとも異なる場合表示します)。

(☞) 更新情報がない場合は、以下のように表示されます。

```
#firmware check official
% No Update Available.
```

2. <最新ファームウェアのダウンロード>

```
#firmware update official
[=====] 11% DOWNLOAD
```

弊社ホームページにある最新のファームウェアをダウンロードします。コマンド実行後、ファームウェアのダウンロードを開始します。

(☞) NXR-G100 シリーズでは、ファームウェアを2つ持つことが可能です。通常ファームウェアの更新を行った場合、更新したファームウェアで起動しますが、no-boot オプションを指定することにより更新したファームウェアで起動しないよう設定することが可能です。

```
#firmware update official no-boot
```

3. <ファームウェアの更新>

ファームウェアのダウンロード完了後、アップデートの実行について問われます。

```
[=====] 100% DOWNLOAD
Update to :
Century Systems NXR-G100 ver 6.1.3 (build 4/14:05 09 06 2014)
Proceed with update? [(y)es/(r)eserve/(n)o]:
```

指定	動作
y	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動を行います (☞) サービスを停止してファームウェアの更新を行うことができません
r	ルータのサービスを継続した状態でファームウェアの更新を開始し、更新後、機器の再起動は行いません (☞) 更新したファームウェアを利用するためには、機器の再起動が必要です
n	ファームウェアの更新を中止します

<補足>

```
Unsaved configuration changes exist. Save Flash? [y/n]:
```

ファームウェア更新時に設定を保存していない場合は、保存するかどうか問い合わせるメッセージが表示されます。保存する場合は「y」、保存しない場合は「n」を指定して実行します。

```
[=====] 100% DOWNLOAD  
Update to :  
  Century Systems NXR-G100 ver 6.1.3 (build 4/14:05 09 06 2014)  
  
Proceed with update? [(y)es/(r)eserve/(n)o]: y  
% Timed out.
```

ファームウェアダウンロード後、3分以内にファームウェアのアップデートを実行しなかった場合（「Proceed with update?」が表示されてから3分以内にn以外を選択しなかった場合）、転送したファームウェアを破棄します。この場合、再度 firmware update コマンドを実行してファイル転送から行う必要があります。

また、ルータのサービスを継続した状態でファームウェアの更新を行った場合、下記操作はできません。

- CLI/GUI からの設定初期化
- CLI/GUI/CMS からの機器再起動
- CLI/GUI/CMS からのファームウェア更新
- GUI/CMS からの設定復帰
- CLI からの設定の復帰/保存
- GUI からの設定(GUIでの設定時には、必ず本体のフラッシュメモリに設定の保存が行われるため)

1-2. 起動時のファームウェア指定

通常起動時は本体内のフラッシュメモリに保存されている 1 つのファームウェアを使用して起動しますが、一部の機種(※)では、本体内のフラッシュメモリに 2 つのファームウェアを保持することが可能です。これにより運用状況に応じて、すぐに以前の状態に戻して運用を継続することができます。

ここでは CLI(コマンドラインインタフェース)を利用して、この機能に対応している機種での起動時のファームウェア指定方法について説明します。

(※)NXR-230/C,NXR-350/C,NXR-G100 シリーズ,WXR-250 で対応

【 実行例 】

<動作中のファームウェアバージョンの確認>

```
#show version  
Century Systems NXR-230 ver 5.25.1 (build 3/19:27 05 12 2013)
```

現在動作中のファームウェアバージョンを表示します。

<保存されているファームウェアの確認>

```
#show version flash  
[Flash:1]  
Century Systems NXR-230 ver 5.25.1 (build 3/19:27 05 12 2013)  
  
[Flash:2]  
Century Systems NXR-230 ver 5.25.2 (build 1/19:25 19 01 2014)
```

本体内のフラッシュメモリに保存されているファームウェアのバージョンを表示します。

<ファームウェアの起動設定の確認>

```
#show system boot  
Boot firmware is on flash 1  
* Current firmware is on flash 1
```

次回起動時のファームウェアが保存されている面、および現在動作中のファームウェアが保存されている面を表示します。この実行例では、次回起動時のファームウェアおよび動作中のファームウェアともに flash1 に保存されているファームウェアとなっています。

<次回起動時のファームウェア指定>

```
(config)#system boot flash 2
```

次回起動時、flash2 に保存されているファームウェアで起動するように設定します。

<補足>

```
#firmware update ftp://192.168.10.100/nxr230-v5252.bin no-boot
```

ファームウェアの更新時にも、no-boot オプションを指定することで更新したファームウェアでは起動しないよう設定することが可能です。

CLI からのファームウェアの更新については [1-1. CLI からのファームウェア更新](#)ご参照ください。

2. 設定の保存・復帰

2-1. 本体への設定保存 1

2-2. 本体への設定保存 2(拡張フラッシュの利用)

2-3. 設定のバックアップ 1

2-4. 設定のバックアップ 2(拡張フラッシュの利用)

2-5. 設定の復帰 1

2-6. 設定の復帰 2(拡張フラッシュの利用)

2-7. 起動時の設定ファイル指定

2-8. 本体内の設定の削除

2-9. 設定の初期化

2-1. 本体への設定保存 1

ここでは CLI(コマンドラインインタフェース)を利用して、機器本体に設定を保存する方法について説明します。NXR,WXR シリーズでは、running-config(現在動作している設定)と startup-config(本体内のフラッシュメモリに保存され起動時に使用する設定)が存在します。ユーザが設定の保存を実行した場合に限り本体内のフラッシュメモリに保存されます。

【 実行例 】

<本体への設定保存>

```
#save config
```

動作中の設定(running-config)を、本体内のフラッシュメモリ(startup-config)に保存します。

2-2. 本体への設定保存 2(拡張フラッシュの利用)

一部の機種(※)では、本体内のフラッシュメモリに複数の設定ファイルを保存しておくことができます。これにより設定変更の際など万一不具合があっても、すぐに以前の状態に戻して運用を継続することができます。

ここでは CLI(コマンドラインインタフェース)を利用して、この機能に対応している機種での設定の保存方法について説明します。

(※)NXR-230/C,NXR-350/C,NXR-G100 シリーズ,WXR-250 で対応

なお、この機能に対応している機種では管理上様々な設定ファイル形式が存在します。

形式は以下のとおりです。

形式	説明
running-config(config)	現在動作している設定です。save config コマンド実行時に保存されます。
initial-config	ファームウェア内部の初期設定です。ethernet0 インタフェースの IP アドレスとして 192.168.0.254/24 が設定されています。
startup-config	起動時に使用する設定です。これは本体のフラッシュメモリ上に存在します。
bootup-config	起動時に使用する設定ファイルの保存場所が記載されています。これは本体のフラッシュメモリ上に存在します。
system-config	起動時に使用する設定です。起動直後であれば、system-config と running-config は同じになります。

【 実行例 】

<本体内のフラッシュメモリ上にある設定ファイルの確認>

```
#dir flash
/flash:
total 4
-rw-rw-rw- 1 root root 1574 Feb 21 17:52 startup-config
```

本体内のフラッシュメモリ上の設定ファイルを表示します。この実行例では、startup-config のみ保存されている状態となります。

(☞) 設定の保存を一度も行っていない場合、ファイルはありません。

<本体内のフラッシュメモリへの設定保存>

```
#save config
```

running-config を本体内のフラッシュメモリ上に startup-config として保存します。

(☞) copy コマンドでも save config コマンドと同様の保存をすることが可能です。

```
#copy config startup-config all
```

```
#copy config flash:config.xml
succeed.
```

running-config を本体内のフラッシュメモリ上にファイル名を指定して保存します。この実行例では、設定ファイル名 config.xml として保存します。

2-3. 設定のバックアップ 1

ここでは CLI(コマンドラインインタフェース)を利用して、FTP サーバや SSH サーバといったリモートサーバや USB メモリなどの外部ストレージに設定をバックアップする方法について説明します。NXR,WXR シリーズでは、running-config(現在動作している設定)と startup-config(本体内のフラッシュメモリに保存され起動時に使用する設定)が存在します。リモートサーバへは running-config および startup-config を転送できます。

【 実行例 】

〔FTP サーバの利用〕

<設定のバックアップ(running-config)>

```
#copy config ftp://192.168.10.100/config.xml  
succeed.
```

running-config を FTP サーバ 192.168.10.100 上にバックアップします。なお、この実行例ではファイル名を config.xml とします。また FTP サーバを利用した設定のバックアップを行う場合、FTP サーバに対して anonymous による接続方法のみ対応しています。よって、ユーザ名やパスワードを指定することはできません。

コマンド実行後、設定ファイルのバックアップを開始します。

(☞) サーバ内で設定ファイルをディレクトリ毎に管理している場合などは、下記のように IP アドレスの後にディレクトリ名/設定ファイル名を指定します。

```
#copy config ftp://192.168.10.100/nxr120/config.xml
```

(☞) IPsec 用の証明書や SSH の公開鍵も合わせて保存する場合は、コマンドのオプションに all を指定します。なお、ファイルは tar+gzip 形式で保存します。この実行例では、設定ファイル名 config.tar.gz として保存します。

```
#copy config ftp://192.168.10.100/config.tar.gz all  
succeed.
```

<設定のバックアップ(startup-config)>

```
#copy startup-config ftp://192.168.10.100/startupconfig.xml  
succeed.
```

startup-config を FTP サーバ 192.168.10.100 上にバックアップします。なお、この実行例ではファイル名を startupconfig.xml とします。

(☞) NXR-130/C は対応していません。(2014/4 現在)

<設定のバックアップ(show-config 形式)>

```
#copy show-config ftp://192.168.10.100/showconfig.txt  
succeed.
```

running-config を show config 形式(CLI コマンド形式)で、FTP サーバ 192.168.10.100 上にバックアッ

ブします。なお、この実行例ではファイル名を showconfig.txt とします。

- (☞) show config 形式は XML 形式に比べ、現在の設定状況の把握が容易です。ですが show config 形式で保存した設定は、CLI または GUI からファイルを読み込んで設定の復帰を行うことができないため、機器の起動後に CLI でログインし、ターミナルソフトなどからコピー&ペーストで設定の反映を行う必要があります。

<補足>

```
#copy config ftp://192.168.10.100/config.xml source 192.168.20.1
```

送信元 IP アドレスを指定することも可能です。通常は出力インタフェースの IP アドレスを送信元 IP アドレスとしますが、冗長化している場合など送信元 IP アドレスを固定したい場合に有効です。

【SSH サーバの利用】

<設定のバックアップ(running-config)>

```
#copy config ssh://test@192.168.10.100/config.xml  
test@192.168.10.100's password:[パスワード]  
succeed.
```

running-config を SSH サーバ 192.168.10.100 上にバックアップします。なお、この実行例ではファイル名を config.xml とします。この時 SSH サーバのアドレスとともにログイン用のユーザ名を合わせて指定します。この設定例では、ユーザ名を test とします。

コマンド実行後、SSH サーバへのログイン用パスワードの入力を求められますので、パスワードを入力します。ログイン後、設定ファイルのバックアップを開始します。

- (☞) サーバ内で設定ファイルをディレクトリ毎に管理している場合などは、下記のように IP アドレスの後にディレクトリ名/設定ファイル名を指定します。

```
# copy config ssh://test@192.168.10.100/nxr120/config.xml
```

- (☞) IPsec 用の証明書や SSH の公開鍵も合わせて保存する場合は、コマンドのオプションに all を指定します。なお、ファイルは tar+gzip 形式で保存します。この実行例では、設定ファイル名 config.tar.gz として保存します。

```
#copy config ssh://test@192.168.10.100/config.tar.gz all  
test@192.168.10.100's password:[パスワード]  
succeed.
```

<設定のバックアップ(startup-config)>

```
#copy startup-config ssh://test@192.168.10.100/startupconfig.xml  
test@192.168.10.100's password:[パスワード]  
succeed.
```

startup-config を SSH サーバ 192.168.10.100 上にバックアップします。なお、この実行例ではファイル名を startupconfig.xml とします。

- (☞) NXR-130/C は対応していません。(2014/7 現在)

<設定のバックアップ(show-config 形式)>

```
#copy show-config ssh://test@192.168.10.100/showconfig.txt  
test@192.168.10.100's password:[パスワード]  
succeed.
```

running-config を show config 形式(CLI コマンド形式)で、SSH サーバ 192.168.10.100 上にバックアップします。なお、この実行例ではファイル名を showconfig.txt とします。

- (☞) show config 形式は XML 形式に比べ、現在の設定状況の把握が容易です。ですが、show config 形式で保存した設定は、CLI または GUI からファイルを読み込んで設定の復帰を行うことができないため、機器の起動後に CLI でログインし、ターミナルソフトなどからコピー&ペーストで設定の復帰を行う必要があります。

<補足>

```
#copy config ssh://test@192.168.10.100/config.xml source 192.168.20.1
```

送信元 IP アドレスを指定することも可能です。通常は出力インタフェースの IP アドレスを送信元 IP アドレスとしますが、冗長化している場合など送信元 IP アドレスを固定したい場合に有効です。

【外部ストレージの利用】

<設定のバックアップ(running-config)>

```
#copy config disk0:config.xml  
succeed.
```

running-config を disk0 と認識された外部ストレージにバックアップします。なお、この実行例ではファイル名を config.xml とします。

- (☞) 外部ストレージで設定ファイルをディレクトリ毎に管理している場合などは、下記のようにディレクトリ名/設定ファイル名を指定します。

```
#copy config disk0:nxr120/config.xml
```

- (☞) IPsec 用の証明書や SSH の公開鍵も合わせて保存する場合は、コマンドのオプションに all を指定します。なお、ファイルは tar+gzip 形式で保存します。この実行例では、設定ファイル名 config.tar.gz として保存します。

```
#copy config disk0:config.tar.gz all  
succeed.
```

<設定のバックアップ(startup-config)>

```
#copy startup-config disk0:startupconfig.xml  
succeed.
```

startup-config を disk0 と認識された外部ストレージにバックアップします。なお、この実行例ではファイル名を startupconfig.xml とします。

- (☞) NXR-130/C は対応していません。(2014/7 現在)

<設定のバックアップ(show-config 形式)>

```
#copy show-config disk0:showconfig.txt  
succeed.
```

running-config を show config 形式(CLI コマンド形式)で、disk0 と認識された外部ストレージにバックアップします。なお、この実行例ではファイル名を showconfig.txt とします。

- (☞) show config 形式は XML 形式に比べ、現在の設定状況の把握が容易です。ですが、show config 形式で保存した設定は、CLI または GUI からファイルを読み込んで設定の復帰を行うことができないため、機器の起動後に CLI でログインし、ターミナルソフトなどからコピー&ペーストで設定の復帰を行う必要があります。

2-4. 設定のバックアップ 2(拡張フラッシュの利用)

一部の機種(※)では、本体内のフラッシュメモリに複数の設定ファイルを保存しておくことができます。これにより設定変更の際、万一不具合があってもすぐに以前の状態に戻して運用を継続することができます。ここでは CLI(コマンドラインインタフェース)を利用して、この機能に対応している機種での設定のバックアップ方法について説明します。

(※)NXR-230/C,NXR-350/C,NXR-G100 シリーズ,WXR-250 で対応

なお、この機能に対応している機種では管理上様々な設定ファイル形式が存在します。

形式は以下のとおりです。

形式	説明
running-config(config)	現在動作している設定です。save config 実行時に保存されます。
initial-config	ファームウェア内部の初期設定です。ethernet0 インタフェースの IP アドレスとして 192.168.0.254/24 が設定されています。
startup-config	起動時に使用する設定です。これは本体のフラッシュメモリ上に存在します。
bootup-config	起動時に使用する設定ファイルの保存場所が記載されています。これは本体のフラッシュメモリ上に存在します。
system-config	起動時に使用する設定です。起動直後であれば、system-config と running-config は同じになります。

【 実行例 】

[FTP サーバの利用]

<設定のバックアップ(running-config)>

```
#copy config ftp://192.168.10.100/config.xml
succeed.
```

running-config を FTP サーバ 192.168.10.100 上にバックアップします。なお、この実行例ではファイル名を config.xml とします。また FTP サーバを利用した設定のバックアップを行う場合、FTP サーバに対して anonymous による接続方法のみ対応しています。よって、ユーザ名やパスワードを指定することはできません。

コマンド実行後、設定ファイルのバックアップを開始します。

(☞) サーバ内で設定ファイルをディレクトリ毎に管理している場合などは、下記のように IP アドレスの後にディレクトリ名/設定ファイル名を指定します。

```
#copy config ftp://192.168.10.100/nxr230/config.xml
```

(☞) IPsec 用の証明書や SSH の公開鍵も合わせて保存する場合は、コマンドのオプションに all を指定します。なお、ファイルは tar+gzip 形式で保存します。この実行例では、設定ファイル名 config.tar.gz として保存します。

```
#copy config ftp://192.168.10.100/config.tar.gz all
succeed.
```

<設定のバックアップ(startup-config)>

```
#copy startup-config ftp://192.168.10.100/startupconfig.xml
succeed.
```

startup-config を FTP サーバ 192.168.10.100 上にバックアップします。なお、この実行例ではファイル名を startupconfig.xml とします。

<設定のバックアップ(show-config 形式)>

```
#copy show-config ftp://192.168.10.100/showconfig.txt
succeed.
```

running-config を show config 形式(CLI コマンド形式)で、FTP サーバ 192.168.10.100 上にバックアップします。なお、この実行例ではファイル名を showconfig.txt とします。

(☞) show config 形式は XML 形式に比べ、現在の設定状況の把握が容易です。ですが、show config 形式で保存した設定は、CLI または GUI からファイルを読み込んで設定の復帰を行うことができないため、機器の起動後に CLI でログインし、ターミナルソフトなどからコピー&ペーストで設定の復帰を行う必要があります。

<設定のバックアップ(本体内のフラッシュメモリ上にある特定ファイル)>

```
#copy flash:config.xml ftp://192.168.10.100/config.xml
succeed.
```

本体内のフラッシュメモリ上にある設定ファイル名 config.xml を、FTP サーバ 192.168.10.100 上にバックアップします。なお、この実行例ではファイル名を config.xml とします。

(☞) tar+gzip 形式で保存している設定も同様に転送することができます。

```
#copy flash:config.tar.gz ftp://192.168.10.100/config.tar.gz
succeed.
```

<補足>

```
#copy config ftp://192.168.10.100/config.xml source 192.168.20.1
```

送信元 IP アドレスを指定することも可能です。通常は出力インタフェースの IP アドレスを送信元 IP アドレスとしますが、冗長化している場合など送信元 IP アドレスを固定したい場合に有効です。

[SSH サーバの利用]**<設定のバックアップ(running-config)>**

```
#copy config ssh://test@192.168.10.100/config.xml
test@192.168.10.100's password:[パスワード]
succeed.
```

running-config を SSH サーバ 192.168.10.100 上にバックアップします。なお、この実行例ではファイル名を config.xml とします。この時 SSH サーバのアドレスとともにログイン用のユーザ名を合わせて指定します。この設定例では、ユーザ名を test とします。

コマンド実行後、SSH サーバへのログイン用パスワードの入力を求められますので、パスワードを入力します。ログイン後、設定ファイルのバックアップを開始します。

- (☞) サーバ内で設定ファイルをディレクトリ毎に管理している場合などは、下記のように IP アドレスの後にディレクトリ名/設定ファイル名を指定します。

```
# copy config ssh://test@192.168.10.100/nxr230/config.xml
```

- (☞) IPsec 用の証明書や SSH の公開鍵も合わせて保存する場合は、コマンドのオプションに all を指定します。なお、ファイルは tar+gzip 形式で保存します。この実行例では、設定ファイル名 config.tar.gz として保存します。

```
#copy config ssh://test@192.168.10.100/config.tar.gz all
test@192.168.10.100's password:[パスワード]
succeed.
```

<設定のバックアップ(startup-config)>

```
#copy startup-config ssh://test@192.168.10.100/startupconfig.xml
test@192.168.10.100's password:[パスワード]
succeed.
```

startup-config を SSH サーバ 192.168.10.100 上にバックアップします。なお、この実行例ではファイル名を startupconfig.xml とします。

<設定のバックアップ(show-config 形式)>

```
#copy show-config ssh://test@192.168.10.100/showconfig.txt
test@192.168.10.100's password:[パスワード]
succeed.
```

running-config を show config 形式(CLI コマンド形式)で、SSH サーバ 192.168.10.100 上にバックアップします。なお、この実行例ではファイル名を showconfig.txt とします。

- (☞) show config 形式は XML 形式に比べ、現在の設定状況の把握が容易です。ですが、show config 形式で保存した設定は、CLI または GUI からファイルを読み込んで設定の復帰を行うことができないため、機器の起動後に CLI でログインし、ターミナルソフトなどからコピー&ペーストで設定の復帰を行う必要があります。

<設定のバックアップ(本体内のフラッシュメモリ上にある特定ファイル)>

```
#copy flash:config.xml ssh://test@192.168.10.100/config.xml
test@192.168.10.100's password:[パスワード]
succeed.
```

本体内のフラッシュメモリ上にある設定ファイル名 config.xml を、SSH サーバ 192.168.10.100 上にバックアップします。なお、この設定例ではファイル名を config.xml とします。

- (☞) tar+gzip 形式で保存している設定も同様に転送することができます。

```
#copy flash:config.tar.gz ssh://test@192.168.10.100/config.tar.gz
test@192.168.10.100's password:[パスワード]
succeed.
```

<補足>

```
#copy config ssh://test@192.168.10.100/config.xml source 192.168.20.1
```

送信元 IP アドレスを指定することも可能です。通常は出力インタフェースの IP アドレスを送信元 IP アド

レスとしますが、冗長化している場合など送信元 IP アドレスを固定したい場合に有効です。

〔外部ストレージの利用〕

<設定のバックアップ(running-config)>

```
#copy config disk0:config.xml
succeed.
```

running-config を disk0 と認識された外部ストレージにバックアップします。なお、この実行例ではファイル名を config.xml とします。

(☞) 外部ストレージで設定ファイルをディレクトリ毎に管理している場合などは、下記のようにディレクトリ名/設定ファイル名を指定します。

```
# copy config disk0:nxr230/config.xml
```

(☞) IPsec 用の証明書や SSH の公開鍵も合わせて保存する場合は、コマンドのオプションに all を指定します。なお、ファイルは tar+gzip 形式で保存します。この実行例では、設定ファイル名 config.tar.gz として保存します。

```
#copy config disk0:config.tar.gz all
succeed.
```

<設定のバックアップ(startup-config)>

```
#copy startup-config disk0:startupconfig.xml
succeed.
```

startup-config を disk0 と認識された外部ストレージにバックアップします。なお、この実行例ではファイル名を startupconfig.xml とします。

<設定のバックアップ(show-config 形式)>

```
#copy show-config disk0:showconfig.txt
succeed.
```

running-config を show config 形式(CLI コマンド形式)で、disk0 と認識された外部ストレージにバックアップします。なお、この実行例ではファイル名を showconfig.txt とします。

(☞) show config 形式は XML 形式に比べ、現在の設定状況の把握が容易です。ですが、show config 形式で保存した設定は、CLI または GUI からファイルを読み込んで設定の復帰を行うことができないため、機器の起動後に CLI でログインし、ターミナルソフトなどからコピー&ペーストで設定の復帰を行う必要があります。

<設定のバックアップ(本体内のフラッシュメモリ上にある特定ファイル)>

```
#copy flash:config.xml disk0:config.xml
succeed.
```

本体内のフラッシュメモリ上にある設定ファイル名 config.xml を、disk0 と認識された外部ストレージにバックアップします。なお、この設定例ではファイル名を config.xml とします。

(☞) tar+gzip 形式で保存している設定も同様に転送することができます。

```
#copy flash:config.tar.gz disk0:config.tar.gz  
succeed.
```

2-5. 設定の復帰 1

ここでは CLI(コマンドラインインタフェース)を利用して、FTP サーバや SSH サーバといったリモートサーバや USB メモリなどの外部ストレージに保存している設定を復帰する方法について説明します。復帰した設定は startup-config(本体内のフラッシュメモリに保存され起動時に使用する設定)に保存されます。よって、running-config(現在動作している設定)に上書きすることはできません。なお、設定復帰可能なファイル形式は XML 形式、または tar.gz 形式となります。

【 実行例 】

〔FTP サーバの利用〕

<設定の復帰>

```
#copy ftp://192.168.10.100/config.xml startup-config  
succeed.
```

FTP サーバ 192.168.10.100 上にある設定ファイルを復帰します。なお、この実行例ではファイル名を config.xml とします。また FTP サーバを利用した設定のバックアップを行う場合、FTP サーバに対して anonymous による接続方法のみ対応しています。よって、ユーザ名やパスワードを指定することはできません。コマンド実行後、設定ファイルの復帰を開始します。復帰した設定は startup-config に保存されますので、次回起動時に復帰した設定が有効となります。

(☞) サーバ内で設定ファイルをディレクトリ毎に管理している場合などは、下記のように IP アドレスの後にディレクトリ名/設定ファイル名を指定します。

```
#copy ftp://192.168.10.100/nxr120/config.xml startup-config
```

(☞) 送信元 IP アドレスを指定することも可能です。通常は出力インタフェースの IP アドレスを送信元 IP アドレスとしますが、冗長化している場合など送信元 IP アドレスを固定したい場合に有効です。

```
#copy ftp://192.168.10.100/config.xml startup-config source 192.168.20.1
```

(☞) NXR-130/C では startup-config の代わりに flash-config を指定する必要があります。(2014/4 現在)

```
#copy ftp://192.168.10.100/config.xml flash-config
```

〔SSH サーバの利用〕

<設定の復帰>

```
#copy ssh://test@192.168.10.100/config.xml startup-config  
test@192.168.10.100's password:[パスワード]  
succeed.
```

SSH サーバ 192.168.10.100 上にある設定ファイルを復帰します。なお、この実行例ではファイル名を config.xml とします。この時 SSH サーバのアドレスとともにログイン用のユーザ名を合わせて指定します。この設定例では、ユーザ名を test とします。

コマンド実行後、SSH サーバへのログイン用パスワードの入力を求められますので、パスワードを入力します。コマンド実行後、設定ファイルの復帰を開始します。復帰した設定は startup-config に保存されます

ので、次回起動時に復帰した設定が有効となります。

- (☞) サーバ内で設定ファイルをディレクトリ毎に管理している場合などは、下記のように IP アドレスの後にディレクトリ名/設定ファイル名を指定します。

```
#copy ssh://test@192.168.10.100/nxr120/config.xml startup-config
```

- (☞) 送信元 IP アドレスを指定することも可能です。通常は出力インタフェースの IP アドレスを送信元 IP アドレスとしますが、冗長化している場合など送信元 IP アドレスを固定したい場合に有効です。

```
#copy ssh://test@192.168.10.100/config.xml startup-config source 192.168.20.1
```

- (☞) NXR-130/C では startup-config の代わりに flash-config を指定する必要があります。(2014/4 現在)

```
#copy ssh://test@192.168.10.100/config.xml flash-config
```

【外部ストレージの利用】

<設定の復帰>

```
#copy disk0:config.xml startup-config  
succeed.
```

disk0 と認識された外部ストレージ上にある設定ファイルを復帰します。なお、この実行例ではファイル名を config.xml とします。コマンド実行後、設定ファイルの復帰を開始します。復帰した設定は startup-config に保存されますので、次回起動時に復帰した設定が有効となります。

- (☞) 外部ストレージで設定ファイルをディレクトリ毎に管理している場合などは、下記のようにディレクトリ名/設定ファイル名を指定します。

```
#copy disk0:nxr120/config.xml startup-config
```

- (☞) NXR-130/C では startup-config の代わりに flash-config を指定する必要があります。(2014/4 現在)

```
#copy disk0:config.xml flash-config
```

2-6. 設定の復帰 2(拡張フラッシュの利用)

一部の機種(※)では、本体内のフラッシュメモリに複数の設定ファイルを保存しておくことができます。これにより設定変更の際など万一不具合があっても、すぐに以前の状態に戻して運用を継続することができます。

ここでは CLI(コマンドラインインタフェース)を利用して、この機能に対応している機種での外部から設定を復帰する方法について説明します。

(※)NXR-230/C,NXR-350/C,NXR-G100 シリーズ,WXR-250 で対応

なお、この機能に対応している機種では管理上様々な設定ファイル形式が存在します。

形式は以下のとおりです。

形式	説明
running-config(config)	現在動作している設定です。save config 実行時に保存されます。
initial-config	ファームウェア内部の初期設定です。ethernet0 インタフェースの IP アドレスとして 192.168.0.254/24 が設定されています。
startup-config	起動時に使用する設定です。これは本体のフラッシュメモリ上に存在します。
bootup-config	起動時に使用する設定ファイルの保存場所が記載されています。これは本体のフラッシュメモリ上に存在します。
system-config	起動時に使用する設定です。起動直後であれば、system-config と running-config は同じになります。

【 実行例 】

〔FTP サーバの利用〕

<設定の復帰(startup-config に保存)>

```
#copy ftp://192.168.10.100/config.xml startup-config
succeed.
```

FTP サーバ 192.168.10.100 上にある設定ファイルを復帰します。なお、この実行例ではファイル名を config.xml とします。また FTP サーバを利用した設定のバックアップを行う場合、FTP サーバに対して anonymous による接続方法のみ対応しています。よって、ユーザ名やパスワードを指定することはできません。コマンド実行後、設定ファイルの復帰を開始します。復帰した設定は startup-config に保存されますので、次回起動時に復帰した設定が有効となります。

(☞) サーバ内で設定ファイルをディレクトリ毎に管理している場合などは、下記のように IP アドレスの後にディレクトリ名/設定ファイル名を指定します。

```
#copy ftp://192.168.10.100/nxr230/config.xml startup-config
```

<設定の復帰(ファイル名を指定して保存)>

```
#copy ftp://192.168.10.100/config.xml flash:config.xml
succeed.
```

FTP サーバ 192.168.10.100 上にあるファイル名 config.xml を本体内のフラッシュメモリ上にファイル名

config.xml として保存します。

- (☞) この方法では本体内のフラッシュメモリ上に保存しただけで、次回起動時に保存した設定が利用されるわけではありません。次回起動時、保存した設定を利用するためには起動時の設定ファイル指定が必要になります。その方法については [2-7. 起動時の設定ファイル指定](#) をご参照ください。

<補足>

```
#copy ftp://192.168.10.100/config.xml startup-config source 192.168.20.1
```

送信元 IP アドレスを指定することも可能です。通常は出力インタフェースの IP アドレスを送信元 IP アドレスとしますが、冗長化している場合など送信元 IP アドレスを固定したい場合に有効です。

[SSH サーバの利用]

<設定の復帰>

```
#copy ssh://test@192.168.10.100/config.xml startup-config  
test@192.168.10.100's password:[パスワード]  
succeed.
```

SSH サーバ 192.168.10.100 上にある設定ファイルを復帰します。なお、この実行例ではファイル名を config.xml とします。この時 SSH サーバのアドレスとともにログイン用のユーザ名を合わせて指定します。この設定例では、ユーザ名を test とします。

コマンド実行後、SSH サーバへのログイン用パスワードの入力を求められますので、パスワードを入力します。コマンド実行後、設定ファイルの復帰を開始します。復帰した設定は startup-config に保存されますので、次回起動時に復帰した設定が有効となります。

- (☞) サーバ内で設定ファイルをディレクトリ毎に管理している場合などは、下記のように IP アドレスの後にディレクトリ名/設定ファイル名を指定します。

```
#copy ssh://test@192.168.10.100/nxr230/config.xml startup-config
```

<設定の復帰(ファイル名を指定して保存)>

```
#copy ssh://test@192.168.10.100/config.xml flash:config.xml  
test@192.168.10.100's password:[パスワード]  
succeed.
```

SSH サーバ 192.168.10.100 上にあるファイル名 config.xml を本体内のフラッシュメモリ上にファイル名 config.xml として保存します。

- (☞) この方法では本体内のフラッシュメモリ上に保存しただけで、次回起動時に保存した設定が利用されるわけではありません。次回起動時、保存した設定を利用するためには起動時の設定ファイル指定が必要になります。その方法については [2-7. 起動時の設定ファイル指定](#) をご参照ください。

<補足>

```
#copy ssh://test@192.168.10.100/config.xml startup-config source 192.168.20.1
```

送信元 IP アドレスを指定することも可能です。通常は出力インタフェースの IP アドレスを送信元 IP アドレスとしますが、冗長化している場合など送信元 IP アドレスを固定したい場合に有効です。

〔外部ストレージの利用〕

<設定の復帰>

```
#copy disk0:config.xml startup-config  
succeed.
```

disk0 と認識された外部ストレージ上にある設定ファイルを復帰します。なお、この実行例ではファイル名を config.xml とします。コマンド実行後、設定ファイルの復帰を開始します。復帰した設定は startup-config に保存されますので、次回起動時に復帰した設定が有効となります。

(☞) 外部ストレージで設定ファイルをディレクトリ毎に管理している場合などは、下記のようにディレクトリ名/設定ファイル名を指定します。

```
#copy disk0:nxr230/config.xml startup-config
```

<設定の復帰(ファイル名を指定して保存)>

```
#copy disk0:config.xml flash:config.xml  
succeed.
```

disk0 と認識された外部ストレージ上にあるファイル名 config.xml を本体内のフラッシュメモリ上にファイル名 config.xml として保存します。

(☞) この方法では本体内のフラッシュメモリ上に保存しただけで、次回起動時に保存した設定が利用されるわけではありません。次回起動時、保存した設定を利用するためには起動時の設定ファイル指定が必要になります。その方法については [2-7. 起動時の設定ファイル指定](#) をご参照ください。

2-7. 起動時の設定ファイル指定

通常起動時の設定は本体内のフラッシュメモリ上に保存されている startup-config を使用しますが、一部の機種(※1)では、本体内に複数保存している設定ファイルの中から指定した設定ファイルを起動時に利用することが可能です。また一部の機種(※2)では、USB メモリなどの外部ストレージに保存している設定ファイルを指定することが可能です。ここでは CLI(コマンドラインインタフェース)を利用して、この機能に対応している機種での起動時の設定ファイル選択方法について説明します。

(※1)NXR-230/C,NXR-350/C,NXR-G100 シリーズ,WXR-250 で対応

(※2)NXR,WXR シリーズ(NXR-130/C,NXR-1200 を除く)で対応

【 実行例 】

<本体内のフラッシュメモリ内の特定ファイルでの起動設定>

```
(config)#system config flash:config.xml
```

本体内のフラッシュメモリ上に保存されている設定ファイル config.xml を、次回起動時に使用するように設定します。

(☞) NXR-230/C,NXR-350/C,WXR-250 で対応

<外部ストレージ内の特定ファイルでの起動設定>

```
(config)#system config disk0:config.xml
```

disk0 と認識された外部ストレージに保存されている設定ファイル config.xml を、次回起動時に使用するよう設定します。

(☞) NXR,WXR シリーズ(NXR-130/C,NXR-1200 を除く)で対応

(☞) 対応機種の中で NXR-230/C,NXR-350/C,WXR-250 を除く機種では、設定後、save config コマンドを実行する必要があります。

<次回起動時の設定ファイル名の確認>

```
#show system config  
flash:config.xml
```

次回起動時の設定ファイル名を確認できます。この実行例では本体内のフラッシュメモリ上に保存されている設定ファイル config.xml で、次回起動します。

(☞) 次回起動時の設定ファイルを指定していない場合は、startup-config で起動します。

<次回起動時の設定ファイル指定の削除>

```
(config)#no system config
```

次回起動時の設定ファイル指定を削除することができます。(bootup-config の初期化)

(☞) 下記コマンドでも同様に削除することができます。

```
#delete bootup-config
```

2-8. 本体内の設定の削除

ここでは CLI(コマンドラインインタフェース)を利用して本体内に保存されている設定ファイルを削除する方法について説明します。

【 実行例 】

<本体内のフラッシュメモリ上にある特定ファイルを削除する>

```
#delete flash config.xml
```

本体内のフラッシュメモリ上に保存されている設定ファイルを削除します。この実行例では設定ファイル名 config.xml を削除します。

```
#delete flash config.tar.gz
```

tar+gzip 形式の設定ファイルも同様に削除できます。この実行例では、設定ファイル名 config.tar.gz を削除します。

<補足>

delete file コマンドでも同様に削除することができます。

```
#delete file flash:config.xml
```

```
#delete file flash:config.tar.gz
```

2-9. 設定の初期化

ここでは CLI(コマンドラインインタフェース)を利用して設定を初期化する方法について説明します。

【 実行例 】

<本体内のフラッシュメモリ内の設定を全て削除する>

```
#erase flash
```

本体内のフラッシュメモリ上に保存されている設定を全て削除します。これにより次回起動時、設定は工場出荷時の状態で起動します。

(注) NXR-130/C のみ以下のコマンドとなります。(2014/4 現在)

```
#erase flash-config
```

<補足>

パスワードがわからないなどの理由によりログインができなくなった場合、erase flash コマンドを実行することができません。このような場合は、INIT ボタンを押下した状態で機器を起動することで一時的に工場出荷時の状態で起動することができます。そして起動後、ログインして erase flash コマンドを実行することで設定を初期化することができます。

(注) INIT ボタンによる起動方法については、ユーザズガイドをご参照ください。

(注) INIT ボタンによる起動を行った場合は、save config コマンドなどの本体内のフラッシュメモリに対する操作(firmware update,erase flash を除く)が禁止されます。また、GUI による設定も禁止されま

3. パケットダンプ

3-1. パケットダンプを pcap 形式で取得する

3-2. パケットダンプの外部転送

3-3. パケットダンプの削除

3-1. パケットダンプを pcap 形式で取得する

パケットダンプ機能は、機器で送受信したパケットを表示・取得する機能です。パケットダンプ機能を利用することにより障害時の切り分けなどに活用することができます。

※パケットダンプ機能利用時の注意点

パケットダンプ機能を利用する場合は、ファストフォワーディング機能を無効にする必要があります。有効のままパケットダンプ機能で情報を取得しても、正しく取得することができません。

ファストフォワーディング機能は、以下の設定で無効にすることができます。

```
(config)#no fast-forwarding enable
```

また L2TPv3 ファストフォワーディング機能も合わせて有効にしている場合は、ともに無効にします。

```
(config)#no l2tpv3 fast-forwarding enable
(config)#no fast-forwarding enable
```

【 実行例 】

<pcap 形式での取得>

```
#dump interface ethernet 0 pcap count 1000 size 1518
```

ethernet0 インタフェースを通過するパケットを対象に、取得するデータ長を 1518 バイトとし、1000 個のパケット取得します。なお取得後、自動的にコマンド実行を終了します。

(☞) Ctrl+C を押すことで、それまで取得した結果を保存して強制終了することができます。

<pcap 形式での取得時に一部通信をフィルタする>

```
#dump interface ethernet 0 pcap count 1000 size 1518 filter telnet
```

ethernet0 インタフェースを通過するパケットを対象に、取得するデータ長を 1518 バイトとし、1000 個のパケット取得します。ただし、telnet ポートでの通信は除きます。

(☞) TELNET アクセスしてパケットダンプを取得するような場合など自身のアクセスを含む TELNET 通信のパケットを取得したくない場合に有効です。

```
#dump interface ethernet 0 pcap count 1000 size 1518 filter telnet ssh
```

ethernet0 インタフェースを通過するパケットを対象に、取得するデータ長を 1518 バイトとし、1000 個のパケット取得します。ただし、telnet ポート,ssh ポートでの通信は除きます。

<補足>

```
#dump interface ethernet 0
```

pcap 形式で取得するためのオプションを指定せずに実行することも可能です。

この場合は CLI 上に実行結果が表示されます。なお、Ctrl+C を押すことで強制終了することができます。

この方法は、Ping による疎通確認など簡単な通信確認時の利用に適しています。

3-2. パケットダンプの外部転送

pcap 形式で取得したパケットダンプはリモートサーバや外部ストレージに転送することができます。ここでは CLI(コマンドラインインタフェース)を利用して、FTP サーバや SSH サーバといったリモートサーバや USB メモリなどの外部ストレージにパケットダンプを転送する方法について説明します。

【 実行例 】

〔FTP サーバの利用〕

<パケットダンプの転送>

```
#copy dump ftp://192.168.10.100/dump.pcap  
succeed.
```

dump コマンドで取得した pcap 形式の実行結果を FTP サーバ 192.168.10.100 上に転送します。なお、この実行例ではファイル名を dump.pcap とします。また FTP サーバを利用して転送する場合、FTP サーバに対して anonymous による接続方法のみ対応しています。よって、ユーザ名やパスワードを指定することはできません。コマンド実行後、パケットダンプを転送します。

(☞) サーバ内でパケットダンプをディレクトリ毎に管理している場合などは、下記のように IP アドレスの後にディレクトリ名/ファイル名を指定します。

```
#copy dump ftp://192.168.10.100/nxr120dump/dump.pcap
```

〔SSH サーバの利用〕

<パケットダンプの転送>

```
#copy dump ssh://test@192.168.10.100/dump.pcap  
test@192.168.10.100's password:[パスワード]  
succeed.
```

dump コマンドで取得した pcap 形式の実行結果を SSH サーバ 192.168.10.100 上に転送します。なお、この実行例ではファイル名を dump.pcap とします。この時 SSH サーバのアドレスとともにログイン用のユーザ名を合わせて指定します。この設定例では、ユーザ名を test とします。

コマンド実行後、SSH サーバへのログイン用パスワードの入力を求められますので、パスワードを入力します。ログイン後、パケットダンプ結果を転送します。

(☞) サーバ内でパケットダンプ結果をディレクトリ毎に管理している場合などは、下記のように IP アドレスの後にディレクトリ名/ファイル名を指定します。

```
#copy dump ssh://test@192.168.10.100/nxr120dump/dump.pcap
```

〔外部ストレージの利用〕

<パケットダンプの転送>

```
#copy dump disk0:dump.pcap  
succeed.
```

dump コマンドで取得した pcap 形式の実行結果を disk0 と認識された外部ストレージに転送します。

なお、この実行例ではファイル名を dump.pcap とします。

(☞) 外部ストレージでパケットダンプ結果をディレクトリ毎に管理している場合などは、下記のようにディレクトリ名/ファイル名を指定します。

```
#copy dump disk0:nxr120dump/dump.pcap
```

3-3. パケットダンプの削除

ここでは CLI(コマンドラインインタフェース)を利用して、取得したパケットダンプを削除する方法について説明します。パケットダンプはコマンドによる削除を行わない限り、本体の RAM 上に保持され続けますので、不要な場合は削除してください。

【 実行例 】

<取得したパケットダンプの削除>

```
#delete dump
```

取得したパケットダンプを削除します。

4. スケジュール設定

- 4-1. PPP/PPPoE を定期的に接続/切断する
- 4-2. モバイル通信端末を定期的にリセットする
- 4-3. 機器本体を定期的に再起動する
- 4-4. シスログを定期的にローテートする
- 4-5. ファームウェアの更新日時を指定する
- 4-6. ファームウェアの更新を自動化する
- 4-7. 指定した時間に Wake ON LAN(WOL)で端末を起動する
- 4-8. システムのスリープ/レジュームの日時を指定する
- 4-9. モニタログを定期的にローテートする
- 4-10. 日時指定例

4-1. PPP/PPPoE を定期的に接続/切断する

スケジュール機能を利用することで指定日時に PPP/PPPoE の接続/切断/再接続を行うことができます。
これにより 3G などのモバイル接続を業務時間中のみ接続するなどの利用を自動化することができます。

【 設定例 】

<指定した時刻に PPP/PPPoE 回線を接続する>

```
(config)#schedule 1 08:00 ** interface ppp 0 connect
```

毎日 8:00 に ppp0 インタフェースの PPP/PPPoE 回線を接続します。

<指定した時刻に PPP/PPPoE 回線を切断する>

```
(config)#schedule 2 17:00 ** interface ppp 0 disconnect
```

毎日 17:00 に ppp0 インタフェースの PPP/PPPoE 回線を切断します。

<指定した時刻に PPP/PPPoE 回線を再接続する>

```
(config)#schedule 3 12:00 sunday interface ppp 0 reconnect
```

毎週日曜日 12:00 に ppp0 インタフェースの PPP/PPPoE 回線を再接続します。

- (☞) PPP/PPPoE の設定例は、弊社ホームページにあります FutureNet NXR,WXR 設定例集 WAN インタフェース編をご参照ください。

4-2. モバイル通信端末を定期的のリセットする

スケジュール機能を利用することで指定した日時にモバイル通信端末をリセットすることができます。

【 設定例 】

<指定した時刻にモバイル通信端末をリセットする>

```
(config)#schedule 1 01:00 ** mobile 0 clear
```

毎日 1:00 にモバイル通信端末(mobile0)のリセットを行います。

- (注) モバイル通信端末を利用した設定例は、弊社ホームページにあります FutureNet NXR,WXR 設定例集 WAN インタフェース編をご参照ください。

4-3. 機器本体を定期的に再起動する

スケジュール機能を利用することで指定した日時に機器本体を再起動することができます。

【 設定例 】

<指定した時刻に機器本体を再起動する>

```
(config)#schedule 1 1:00 ** system restart
```

毎日 1:00 に機器本体の再起動を行います。

4-4. シスログを定期的にローテートする

スケジュール機能を利用することで指定した日時にシスログをローテートすることができます。

【 設定例 】

<指定した時刻にシスログをローテートする>

```
(config)#schedule 1 04:00 ** syslog rotate
```

毎日 4:00 にシスログのローテートを行います。

(注) 指定時刻にローテートが行われるかどうかは、シスログのローテート設定に依存します。シスログのローテート設定については [5-3. シスログのローテート](#) も合わせてご参照ください。

4-5. ファームウェアの更新日時を指定する

スケジュール機能を利用することで、指定した日時にファームウェアを更新することができます。

【 設定例 】

<指定した時刻にファームウェアを更新する>

```
(config)#schedule 1 01:20 10 1 firmware update ftp://192.168.10.100/nxr120-v5241G.bin
```

1/10 の 1:20 にファームウェアの更新を行います。ファームウェアは FTP サーバ 192.168.10.100 上にあるファームウェア(この設定例では nxr120-v5241G.bin)を使用します。

```
(config)#schedule 1 01:20 10 1 firmware update ssh://test@192.168.10.100/nxr120-v5241G.bin password [パスワード]
```

1/10 の 1:20 にファームウェアの更新を行います。ファームウェアは SSH サーバ 192.168.10.100 上にあるファームウェア(この設定例では nxr120-v5241G.bin)を使用します。

```
(config)#schedule 1 01:20 10 1 firmware update disk0:nxr120-v5241G.bin
```

1/10 の 1:20 にファームウェアの更新を行います。ファームウェアは disk0 と認識された外部ストレージ上にあるファームウェア(この設定例では nxr120-v5241G.bin)を使用します。

- (☞) 機種によっては hold オプションを設定することができます。hold オプションはバックグラウンドでファームウェアの更新を行います。hold を指定した場合、ファームウェア更新後の自動再起動を保留し、再起動するまでは既存のファームウェアで動作します。

```
(config)#schedule 1 01:20 10 1 firmware update ftp://192.168.10.100/nxr125-v5252.bin hold
```

- (☞) スケジュールによるファームウェア更新時、ファームウェアのバージョンチェックを行います。そして同じバージョンの場合、ファームウェアの更新は行いません。これによりスケジュール設定が残っている場合でも、不要なファームウェアの更新を回避することができます。なお、拡張フラッシュに対応した機種では起動面と非起動面の両方のファームウェアバージョンを比較対象とします。そして、いずれかの面と同じだった場合、ファームウェアの更新は実行されません。
- (☞) その他ファームウェアの更新については、[1-1. CLI からのファームウェア更新](#)が参考になりますので、合わせてご参照ください。

4-6. ファームウェアの更新を自動化する

NXR-G100 シリーズでは、スケジュール機能を利用することで弊社ホームページにある最新ファームウェアの更新状況のチェック、および最新ファームウェアの更新を自動化することができます。

【 設定例 】

<指定した時刻に最新のファームウェア情報を確認する>

```
(config)#schedule 1 1:10 monday firmware check official
```

毎週月曜日の 1:10 に弊社ホームページに最新のファームウェアがあるか確認します。

(☞) 更新情報がある場合は、show version 実行時、またはシスログに出力します。

```
#show version
Century Systems NXR-G100 Series ver 6.1.2 (build 4/18:25 22 05 2014)
Update Available :
  Century Systems NXR-G100 ver 6.1.3 (build 4/14:05 09 06 2014)
#show syslog message
nxrg100 scheduler: schedule 1 executes firmware update check
nxrg100 scheduler: Update Available !!
nxrg100 scheduler: Century Systems NXR-G100 ver 6.1.3 (build 4/14:05 09 06 2014)
```

<指定した時刻に最新のファームウェアに更新する>

```
(config)#schedule 1 1:10 saturday firmware update official
```

毎週土曜日の 1:10 に弊社ホームページにある最新のファームウェアをダウンロードし、完了後、ファームウェアを更新します。

(☞) バックグラウンドでファームウェアの更新を行う hold オプションを設定することができます。hold を指定した場合、ファームウェア更新後の自動再起動を保留し、再起動するまでは既存のファームウェアで動作します。

```
(config)#schedule 1 1:10 saturday firmware update official hold
```

(☞) スケジュールによるファームウェア更新時、ファームウェアのバージョンチェックを行います。チェック時は、起動面と非起動面の両方のファームウェアバージョンを比較対象とします。そして、いずれかの面と同じだった場合、ファームウェアの更新は実行されません。これによりスケジュール設定が残っている場合でも、不要なファームウェアの更新を回避することができます。

4-7. 指定した時間に Wake ON LAN(WOL)で端末を起動する

NXR-G100 シリーズでは、スケジュール機能を利用することでスリープ状態にある端末にマジックパケットを送信し、端末の電源を ON にすることができます。

【 設定例 】

<指定した時刻に WOL を利用して端末を起動する>

```
(config)#schedule 1 9:00 monday friday wol send name PC
```

月曜日から金曜日の 9:00 に PC という名前で定義された端末にマジックパケットを送信します。

(注) 指定端末の定義は別途 wol コマンドで設定する必要があります。wol コマンドによる設定については、[9-1. Wake ON LAN による端末起動](#)をご参照ください。

4-8. システムのスリープ/レジュームの日時を指定する

NXR-G100 シリーズの一部機種では、スケジュール機能を利用することで指定した日時にシステムをスリープ/レジュームすることができます。

【 設定例 】

<指定した時刻にシステムをスリープ状態にする>

```
(config)#schedule 1 18:00 ** system sleep
```

毎日 18:00 にスリープ状態に遷移します。

(注) オプションで timer を設定しなかった場合、365 日間スリープ状態にする設定となります。この場合、レジュームさせる方法として INIT ボタン押下などがあります。

<指定した時刻にシステムをスリープ状態にし、その後レジュームする>

```
(config)#schedule 1 8:00 ** system resume  
(config)#schedule 2 17:00 ** system sleep schedule 1
```

毎日 17:00 にスリープ状態に遷移し、8:00 にレジュームします。

4-9. モニタログを定期的にローテートする

スケジュール機能を利用することで指定した日時にモニタログをローテートすることができます。

【 設定例 】

<指定した時刻にモニタログ(Ping/Ping6 監視)をローテートする>

```
(config)#schedule 1 04:00 ** monitor-log reachability rotate
```

毎日 4:00 にモニタログ(Ping/Ping6 監視)のローテートを行います。

(☞) モニタログ設定については [12-1. モニタログ\(監視結果の保存\)](#) も合わせてご参照ください。

<指定した時刻にモニタログ(システムリソース監視)をローテートする>

```
(config)#schedule 1 05:00 ** monitor-log resource rotate
```

毎日 5:00 にモニタログ(システムリソース監視)のローテートを行います。

(☞) モニタログ設定については [12-1. モニタログ\(監視結果の保存\)](#) も合わせてご参照ください。

4-10. 日時指定例

ここではスケジュール設定で指定する日時指定の例を紹介します。

なお、この設定例は日時指定に特化したものですので、設定を有効にするためにはスケジュール設定で指定可能なアクション項目を合わせて設定する必要があります。

【 設定例 】

<毎時 0 分に実行する>

```
schedule 1 *:00 **
```

<毎日 1:20 に実行する>

```
schedule 1 1:20 **
```

<毎月 10 日の 1:20 に実行する>

```
schedule 1 1:20 10 *
```

<毎月 10 日の毎時 20 分に実行する>

```
schedule 1 *:20 10 *
```

<1/10 の毎時 20 分に実行する>

```
schedule 1 *:20 10 1
```

<1/10 の 10:20 に実行する>

```
schedule 1 10:20 10 1
```

<1 月の毎日 10:20 に実行する>

```
schedule 1 10:20 * 1
```

<毎週月曜日の毎時 10 分に実行する>

```
schedule 1 *:10 monday
```

<毎週日曜日の 1:10 に実行する>

schedule 1 1:10 Sunday

<月曜日から金曜日の 4:10 に実行する>

schedule 1 4:10 monday friday

5. シスログ設定

5-1. シスログの取得

5-2. シスログサーバへの転送

5-3. シスログのローテート

5-1. シスログの取得

シスログは本体内(ローカル)、外部ストレージ、シスログサーバに出力することができます。ここでは CLI(コマンドラインインタフェース)を利用して、ローカル出力および外部ストレージ出力の方法について説明します。

【 設定例 】

<ローカルに出力する>

```
(config)#syslog  
(config-syslog)#local enable
```

シスログをローカル出力します。

<外部ストレージに出力する>

```
(config)#syslog  
(config-syslog)#local file disk0:syslog.txt
```

disk0 と認識された外部ストレージにシスログを出力します。なお、この設定例ではファイル名を syslog.txt とします。

5-2. シスログサーバへの転送

シスログは本体内(ローカル)、外部ストレージ、シスログサーバに出力することができます。ここでは CLI(コマンドラインインタフェース)を利用して、シスログをシスログサーバに転送する方法について説明します。

【 設定例 】

<指定したシスログサーバにログを出力する>

```
(config)#syslog  
(config-syslog)#server 192.168.10.100
```

シスログをシスログサーバ 192.168.10.100 に転送します。

(☞) シスログサーバ送信時の送信元アドレスも設定することができます。この実行例では送信元 IP アドレスとして 192.168.10.1 を設定しています。

```
(config-syslog)#server 192.168.10.100 source 192.168.10.1
```

<補足>

シスログサーバへの転送が失敗した場合は、再送キューに蓄積し、60 秒後に再送します。以降 60 秒毎に再送を試行し、シスログサーバへの転送が成功すると、再送キューをクリアします。この場合のタイムスタンプは、サーバへの送信時刻ではなく、最初にシスログを送信しようとした際の時刻になります。なお、蓄積可能なメッセージキューの長さは 1000 件となります。

5-3. シスログのローテート 1(外部ストレージの利用)

シスログ機能では、シスログメッセージの容量が最大許容量の 80%を超えると、後方の 4000 行を残して削除します。また、これはシステム起動時より 10 分周期で自動的にチェックします。

それとは別にローテート設定をすることができます。ローテート設定を行うことで指定した間隔で定期的に外部ストレージにシスログのバックアップを取得することができます。その際ログファイルのサイズ指定やバックアップファイル数を指定することができます。

【 設定例 】

<指定日時にシスログをローテートする>

```
(config)#schedule 1 4:00 sunday syslog rotate  
(config)#syslog  
(config-syslog)#rotate disk0 threshold files 5
```

毎週日曜日 4:00 にローテートを実行し、disk0 と認識された外部ストレージにログファイルをバックアップします。なお、この設定例ではバックアップファイル数を 5 とします。

5-4. シスログのローテート 2(内蔵フラッシュの利用)

この設定例では定期的にシスログのバックアップを取得し、内蔵フラッシュに保存します。その際ログファイルのサイズやバックアップファイル数を指定することができます。

【対象機種】 NXR-G100 シリーズ

【 設定例 】

<指定日時にシスログをローテートする>

```
(config)#schedule 1 4:00 sunday syslog rotate  
(config)#syslog  
(config-syslog)#rotate flash1 threshold files 5
```

毎週日曜日 4:00 にローテートを実行し、内蔵フラッシュにログファイルをバックアップします。なお、この設定例ではバックアップファイル数を 5 とします。

5-5. メモリ上のシスログの外部転送

シスログサーバへの転送とは別にシスログをファイルとして外部に転送することもできます。

【 実行例 】

〔FTP サーバの利用〕

```
#copy syslog ftp://192.168.10.100/log.gz  
succeed.
```

シスログファイルを FTP サーバ 192.168.10.100 上に転送します。なおこの実行例では、ファイル名を log.gz とします。また FTP サーバを利用して転送する場合、anonymous による接続方法のみ対応しています。よってユーザ名やパスワードを指定することはできません。コマンド実行後、シスログファイルを転送します。

(☞) サーバ内でシスログファイルをディレクトリ毎に管理している場合は、下記のように IP アドレスの後に「ディレクトリ名/ファイル名」を指定します。

```
#copy syslog ftp://192.168.10.100/syslog/log.gz
```

〔SSH サーバの利用〕

```
#copy syslog ssh://test@192.168.10.100/log.gz  
test@192.168.10.100's password:[パスワード]  
succeed.
```

シスログファイルを SSH サーバ 192.168.10.100 上に転送します。なおこの実行例では転送先のファイル名を log.gz とします。この時 SSH サーバの IP アドレスとともにログイン用のユーザ名を合わせて指定します。この実行例では、ユーザ名を test とします。コマンド実行後、SSH サーバへのログイン用パスワードの入力を求められますので、パスワードを入力します。ログイン後、シスログファイルを転送します。

(☞) サーバ内でシスログファイルをディレクトリ毎に管理している場合は、下記のように IP アドレスの後に「ディレクトリ名/ファイル名」を指定します。

```
#copy syslog ssh://test@192.168.10.100/syslog/log.gz
```

〔外部ストレージの利用〕

```
#copy syslog disk0:log.gz  
succeed.
```

シスログファイルを disk0 と認識された外部ストレージに転送します。なおこの実行例ではファイル名を log.gz とします。

(☞) 外部ストレージでシスログファイルをディレクトリ毎に管理している場合は、下記のように「ディレクトリ名/ファイル名」を指定します。

```
#copy syslog disk0:syslog/log.gz
```

【 補足 】

<外部ストレージ内のシスログを表示する>

```
#show syslog file disk0:[ファイル名]
```

disk0 と認識された外部ストレージに保存したシスログファイルを表示します。

5-6. 内蔵フラッシュ上のシスログの外部転送

NXR-G100 シリーズでは、内蔵フラッシュに保存したシスログを外部に転送することができます。

【対象機種】 NXR-G100 シリーズ

【 実行例 】

〔FTP サーバの利用〕

```
#copy flash1:[ファイル名] ftp://192.168.10.100/log.gz  
succeed.
```

内蔵フラッシュ内のシスログファイルを FTP サーバ 192.168.10.100 上に転送します。なおこの実行例では、ファイル名を log.gz とします。また FTP サーバを利用して転送する場合、anonymous による接続方法のみ対応しています。よってユーザ名やパスワードを指定することはできません。コマンド実行後、シスログ情報ファイルを転送します。

(☞) サーバ内でシスログファイルをディレクトリ毎に管理している場合は、下記のように IP アドレスの後に「ディレクトリ名/ファイル名」を指定します。

```
#copy flash1:[ファイル名] ftp://192.168.10.100/syslog/log.gz
```

〔SSH サーバの利用〕

```
#copy flash1:[ファイル名] ssh://test@192.168.10.100/log.gz  
test@192.168.10.100's password:[パスワード]  
succeed.
```

内蔵フラッシュ内のシスログファイルを SSH サーバ 192.168.10.100 上に転送します。なおこの実行例では転送先のファイル名を log.gz とします。この時 SSH サーバの IP アドレスとともにログイン用のユーザ名を合わせて指定します。この実行例では、ユーザ名を test とします。コマンド実行後、SSH サーバへのログイン用パスワードの入力を求められますので、パスワードを入力します。ログイン後、シスログファイルを転送します。

(☞) サーバ内でシスログファイルをディレクトリ毎に管理している場合は、下記のように IP アドレスの後に「ディレクトリ名/ファイル名」を指定します。

```
#copy flash1:[ファイル名] ssh://test@192.168.10.100/syslog/log.gz
```

〔外部ストレージの利用〕

```
#copy flash1:[ファイル名] disk0:log.gz  
succeed.
```

内蔵フラッシュ内のシスログファイルを disk0 と認識された外部ストレージに転送します。なおこの実行例ではファイル名を log.gz とします。

(☞) 外部ストレージでシスログファイルをディレクトリ毎に管理している場合は、下記のように「ディレクトリ名/ファイル名」を指定します。

```
#copy flash1:[ファイル名] disk0:syslog/log.gz
```


【 補足 】**<内蔵フラッシュ内のシスログを表示する>**

```
#show syslog file flash1:[ファイル名]
```

内蔵フラッシュに保存したシスログファイルを表示します。

5-7. 機器再起動時のシスログバックアップ

シスログは通常ファームウェアの更新や restart コマンドによる機器再起動などを行うと消去されますが、再起動時にシスログを外部ストレージや内蔵フラッシュ(※)に保存することができます。これにより再起動後も再起動前のログを確認することができます。

ただし、以下のような場合にはバックアップの取得はできません。

- 電源ケーブルの抜き差しによる電源の OFF/ON
- 停電による電源の OFF/ON
- 予期しないシステムのハングアップ等に伴う機器再起動時

※NXR-G100 シリーズのみ対応

【 設定例 】

<機器再起動時シスログを外部ストレージに保存する>

```
(config)#syslog
(config-syslog)#auto-rotate enable
(config-syslog)#rotate disk0 threshold files 5
```

機器再起動時に disk0 と認識された外部ストレージにシスログをバックアップします。なお、この設定例ではバックアップファイル数を 5 とします。

(☞) シスログはサイズに関係なくバックアップされます。よって threshold logsize の設定値は無視されません。

<機器再起動時シスログを内蔵フラッシュに保存する>

【対象機種】 NXR-G100 シリーズ

```
(config)#syslog
(config-syslog)#auto-rotate enable
(config-syslog)#rotate flash1 threshold files 5
```

機器再起動時に内蔵フラッシュにシスログをバックアップします。なお、この設定例ではバックアップファイル数を 5 とします。

(☞) シスログはサイズに関係なくバックアップされます。よって threshold logsize の設定値は無視されません。

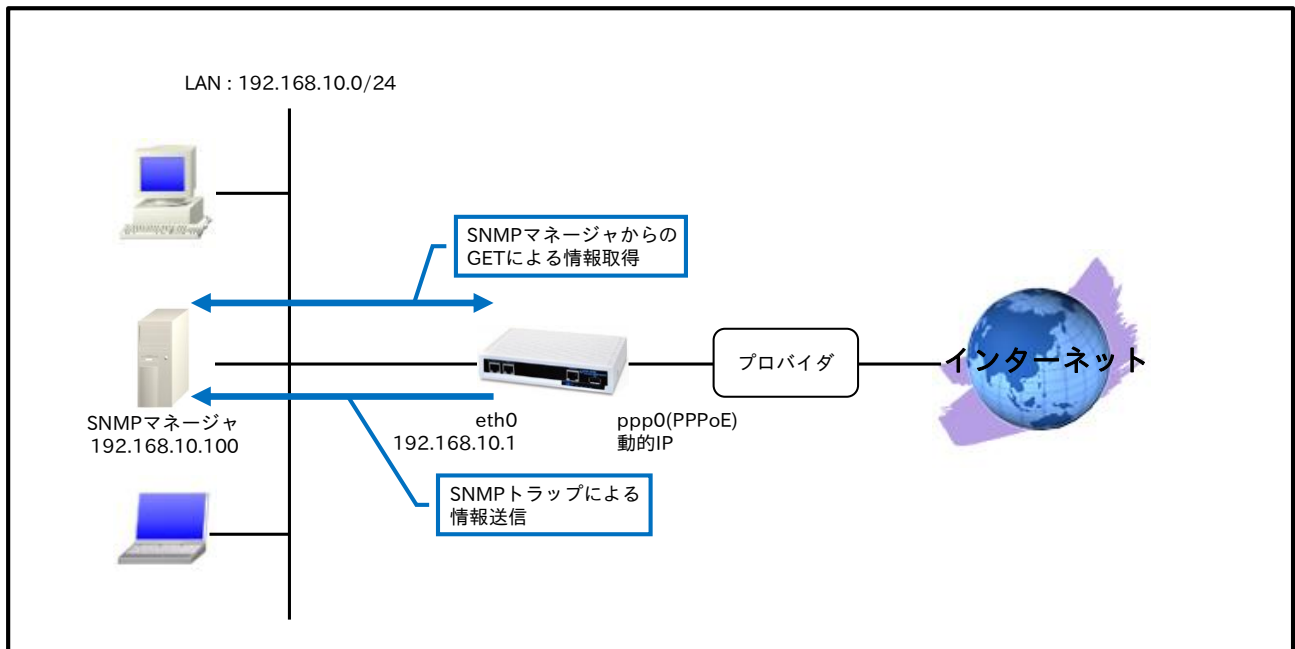
6. SNMP 設定

6-1. SNMP エージェント設定

6-1. SNMP エージェント設定

SNMP 機能では、システム情報の取得、およびシステム上で状態変化が発生した際にトラップを送信することが可能です。ここでは SNMP マネージャでの情報取得とトラップの送信設定について説明します。

【 構成図 】



【 設定パラメータ 】

設定項目		設定内容	
LAN 側インタフェース	ethernet0 の IP アドレス	192.168.10.1/24	
WAN 側インタフェース	PPPoE クライアント(ethernet1)	ppp0	
	ppp0 の IP アドレス	動的 IP アドレス	
	IP マスカレード	有効	
	SPI	有効	
	MSS 自動調整	オート	
	IP リダイレクト	無効	
	ISP 接続用ユーザ ID	test1@example.jp	
	ISP 接続用パスワード	test1pass	
スタティックルート	宛先 IP アドレス	0.0.0.0/0	
	ゲートウェイ(インタフェース)	ppp0	
SNMP	SNMP マネージャの IP アドレス	192.168.10.100/32	
	コミュニティ名	public	
	sysContact	admin	
	sysLocation	Honsya	
	sysName	Honsya-NXR	
	トラップ宛先 IP アドレス	192.168.10.100	
	トラップコミュニティ名	public	
	inform リクエスト	再送間隔	10 秒
		再送回数	3 回
	送信元 IP アドレス	192.168.10.1	
LED	aux1	ppp0 アップ時点灯	
DNS	サービス	有効	
FastForwarding		有効	

【 設定例 】

```
nrx120#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
nrx120(config)#interface ethernet 0
nrx120(config-if)#ip address 192.168.10.1/24
nrx120(config-if)#exit
nrx120(config)#ip route 0.0.0.0/0 ppp 0
nrx120(config)#interface ppp 0
nrx120(config-ppp)#ip address negotiated
nrx120(config-ppp)#ip masquerade
nrx120(config-ppp)#ip spi-filter
nrx120(config-ppp)#ip tcp adjust-mss auto
nrx120(config-ppp)#no ip redirects
nrx120(config-ppp)#ppp username test1@example.jp password test1pass
nrx120(config-ppp)#exit
nrx120(config)#interface ethernet 1
nrx120(config-if)#no ip address
nrx120(config-if)#pppoe-client ppp 0
nrx120(config-if)#exit
nrx120(config)#system led aux 1 interface ppp 0
nrx120(config)#dns
nrx120(config-dns)#service enable
nrx120(config-dns)#exit
nrx120(config)#snmp
nrx120(config-snmp)#security 192.168.10.100/32 public
nrx120(config-snmp)#syscontact admin
nrx120(config-snmp)#syslocation Honsya
nrx120(config-snmp)#sysname Honsya-NXR
nrx120(config-snmp)#trap manager 192.168.10.100 public inform interval 10 retry 3
nrx120(config-snmp)#bind address 192.168.10.1
nrx120(config-snmp)#exit
nrx120(config)#fast-forwarding enable
nrx120(config)#exit
nrx120#save config
```

【 設定例解説 】

1. <LAN 側(ethernet0)インタフェース設定>

```
nrx120(config)#interface ethernet 0
nrx120(config-if)#ip address 192.168.10.1/24
```

LAN 側(ethernet0)インタフェースの IP アドレスを設定します。

2. <スタティックルート設定>

```
nrx120(config)#ip route 0.0.0.0/0 ppp 0
```

デフォルトルートを設定します。

3. <WAN 側(ppp0)インタフェース設定>

```
nrx120(config)#interface ppp 0
nrx120(config-ppp)#ip address negotiated
```

WAN 側(ppp0)インタフェースを設定します。IP アドレスとして negotiated を設定します。

```
nrx120(config-ppp)#ip masquerade  
nrx120(config-ppp)#ip spi-filter  
nrx120(config-ppp)#ip tcp adjust-mss auto  
nrx120(config-ppp)#no ip redirects
```

IP マスカレードを有効、ステートフルパケットインスペクションを有効に設定します。
また、TCP MSS の調整機能をオート、ICMP リダイレクト機能を無効に設定します。

```
nrx120(config-ppp)#ppp username test1@example.jp password test1pass
```

ISP との接続で使用するユーザ ID とパスワードを設定します。

4. <ethernet1 インタフェース設定>

```
nrx120(config)#interface ethernet 1  
nrx120(config-if)#no ip address  
nrx120(config-if)#pppoe-client ppp 0
```

PPPoE クライアントとして ppp0 インタフェースを使用できるように設定します。

5. <システム LED 設定>

```
nrx120(config)#system led aux 1 interface ppp 0
```

ppp0 インタフェースのアップ/ダウンを aux1 LED で表示するように設定します。

6. <DNS 設定>

```
nrx120(config)#dns  
nrx120(config-dns)#service enable
```

DNS 設定で DNS サービスを有効にします。

7. <SNMP 設定>

```
nrx120(config)#snmp
```

SNMP を設定します。

```
nrx120(config-snmp)#security 192.168.10.100/32 public
```

SNMP マネージャの IP アドレス、コミュニティ名を設定します。

(☞) SNMP マネージャが設置されているネットワーク範囲(ex. 192.168.10.0/24)を設定することも可能です。

```
nrx120(config-snmp)#syscontact admin
```

sysContact を設定します。

(☞) sysContact には、一般的に管理者名やその連絡先を設定します。

```
nrx120(config-snmp)#syslocation Honsya
```

sysLocation を設定します。

(☞) sysLocation には、一般的に設置場所を設定します。

```
nxr120(config-snmp)#sysname Honsya-NXR
```

sysName を設定します。

(注) sysName には、一般的に管理上の機器名称を設定します。初期値では機種名が設定されています。

```
nxr120(config-snmp)#trap manager 192.168.10.100 public inform interval 10 retry 3
```

SNMP トラップの宛先 IP アドレス、SNMP トラップのコミュニティ名を設定します。また、Inform リクエストを送信できるように設定します。これは宛先からの応答がない時に再送する機能です。

(注) SNMP Inform は SNMPv2c で対応しています。なお、Inform リクエストを設定しなくても SNMP トラップを送信することは可能です。

```
nxr120(config-snmp)#bind address 192.168.10.1
```

SNMP トラップ送信時の送信元 IP アドレスを設定します。

(注) 指定したアドレスに対応したインタフェースがリンクダウンの状態の場合でも、当該アドレスを送信元 IP アドレスとして使用することができます。

8. <ファストフォワーディングの有効化>

```
nxr120(config)#fast-forwarding enable
```

ファストフォワーディングを有効にします。ファストフォワーディングを設定することによりパケット転送の高速化を行うことができます。

(注) ファストフォワーディングの詳細および利用時の制約については、NXR,WXR シリーズのユーザーズガイド(CLI 版)に記載されているファストフォワーディングの解説をご参照ください。

【 端末の設定例 】

	端末	SNMP マネージャ
IP アドレス	192.168.10.200	192.168.10.100
サブネットマスク	255.255.255.0	
デフォルトゲートウェイ	192.168.10.1	
DNS サーバ		

7. 時刻同期設定

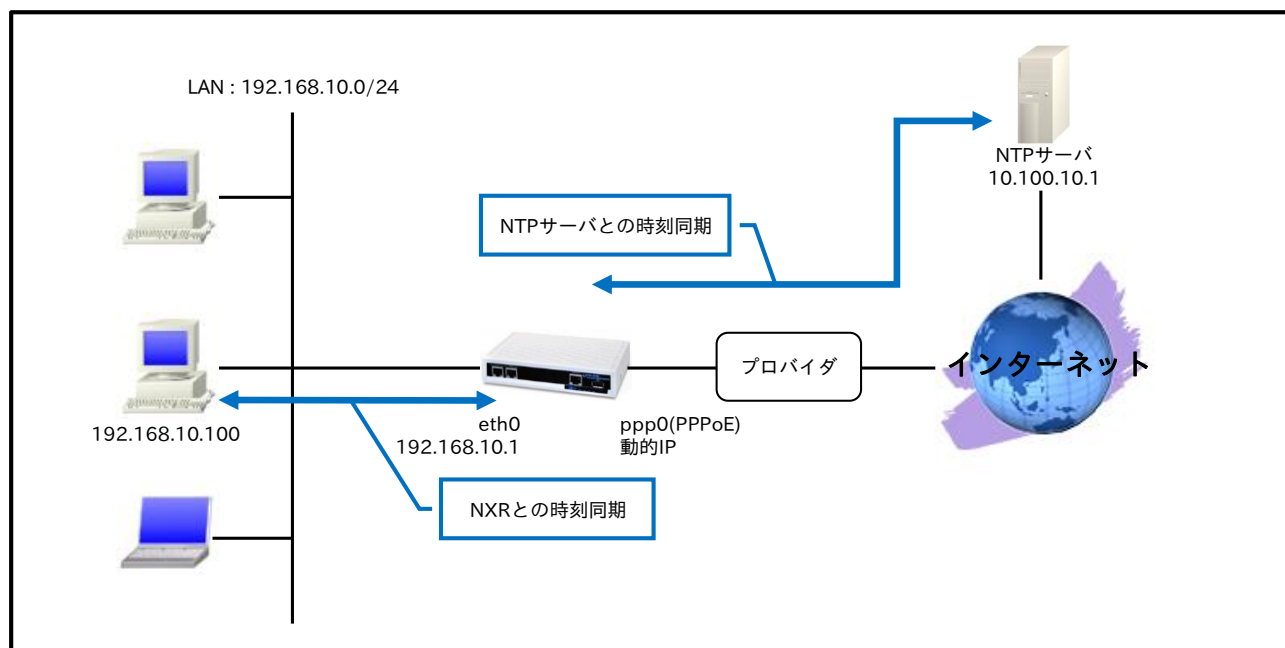
7-1. NTP 設定

7-2. WIN 通信モジュールを利用した時刻同期設定

7-1. NTP 設定

NTP 機能は、時刻同期を行う機能です。ここでは NTP クライアント/NTP サーバとして動作するための設定について説明します。

【 構成図 】



- ・ NTP サーバを指定して時刻同期を行うのと同時に、ルータを NTP サーバとして利用することでルータ配下の端末からの時刻同期要求に応答するようにします。

【 設定パラメータ 】

設定項目		設定内容	
LAN 側インタフェース	ethernet0 の IP アドレス	192.168.10.1/24	
WAN 側インタフェース	PPPoE クライアント(ethernet1)	ppp0	
	ppp0 の IP アドレス	動的 IP アドレス	
	IP マスカレード	有効	
	SPI	有効	
	MSS 自動調整	オート	
	IP リダイレクト	無効	
	ISP 接続用ユーザ ID	test1@example.jp	
スタティックルート	宛先 IP アドレス	0.0.0.0/0	
	ゲートウェイ(インタフェース)	ppp0	
NTP	NTP サーバの IP アドレス		10.100.10.1
	ポーリング間隔	最小	4
		最大	5
タイムアウト	30 秒		
LED	aux1	ppp0 アップ時点灯	
DNS	サービス	有効	
FastForwarding		有効	

【 設定例 】

```
nrx120#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
nrx120(config)#interface ethernet 0
nrx120(config-if)#ip address 192.168.10.1/24
nrx120(config-if)#exit
nrx120(config)#ip route 0.0.0.0/0 ppp 0
nrx120(config)#interface ppp 0
nrx120(config-ppp)#ip address negotiated
nrx120(config-ppp)#ip masquerade
nrx120(config-ppp)#ip spi-filter
nrx120(config-ppp)#ip tcp adjust-mss auto
nrx120(config-ppp)#no ip redirects
nrx120(config-ppp)#ppp username test1@example.jp password test1pass
nrx120(config-ppp)#exit
nrx120(config)#interface ethernet 1
nrx120(config-if)#no ip address
nrx120(config-if)#pppoe-client ppp 0
nrx120(config-if)#exit
nrx120(config)#system led aux 1 interface ppp 0
nrx120(config)#dns
nrx120(config-dns)#service enable
nrx120(config-dns)#exit
nrx120(config)#ntp
nrx120(config-ntp)#service enable
nrx120(config-ntp)#server 10.100.10.1 polling 4 5
nrx120(config-ntp)#timeout 30
nrx120(config-ntp)#exit
nrx120(config)#fast-forwarding enable
nrx120(config)#exit
nrx120#save config
```

【 設定例解説 】

1. <LAN 側(ethernet0)インタフェース設定>

```
nrx120(config)#interface ethernet 0
nrx120(config-if)#ip address 192.168.10.1/24
```

LAN 側(ethernet0)インタフェースの IP アドレスを設定します。

2. <スタティックルート設定>

```
nrx120(config)#ip route 0.0.0.0/0 ppp 0
```

デフォルトルートを設定します。

3. <WAN 側(ppp0)インタフェース設定>

```
nrx120(config)#interface ppp 0
nrx120(config-ppp)#ip address negotiated
```

WAN 側(ppp0)インタフェースを設定します。IP アドレスとして negotiated を設定します。

```
nrx120(config-ppp)#ip masquerade
nrx120(config-ppp)#ip spi-filter
nrx120(config-ppp)#ip tcp adjust-mss auto
nrx120(config-ppp)#no ip redirects
```

IP マスカレード、ステートフルパケットインスペクションを有効に設定します。

また、TCP MSS の調整機能をオート、ICMP リダイレクト機能を無効に設定します。

```
nxr120(config-ppp)#ppp username test1@example.jp password test1pass
```

ISP との接続で使用するユーザ ID とパスワードを設定します。

4. <ethernet1 インタフェース設定>

```
nxr120(config)#interface ethernet 1  
nxr120(config-if)#no ip address  
nxr120(config-if)#pppoe-client ppp 0
```

PPPoE クライアントとして ppp0 インタフェースを使用できるように設定します。

5. <システム LED 設定>

```
nxr120(config)#system led aux 1 interface ppp 0
```

ppp0 インタフェースのアップ/ダウンを AUX1 LED で表示するように設定します。

6. <DNS 設定>

```
nxr120(config)#dns  
nxr120(config-dns)#service enable
```

DNS 設定で DNS サービスを有効にします。

7. <NTP 設定>

```
nxr120(config)#ntp
```

NTP を設定します。

```
nxr120(config-ntp)#service enable
```

NTP サービスを有効にします。

```
nxr120(config-ntp)#server 10.100.10.1 polling 4 5
```

NTP サーバアドレスとポーリング間隔を設定します。

(☞) ポーリング間隔で設定した値の意味は、4 は 2 の 4 乗=16 秒を、5 は 2 の 5 乗=32 秒を表しています。よって、NTP の通信量を減らすためにポーリング間隔を大きくしたり、精度を上げるためにポーリング間隔を小さくしたりすることができます。

```
nxr120(config-ntp)#timeout 30
```

NTP サーバからの応答タイムアウト時間を設定します。

(☞) NTP サーバからのサンプリングを 4 回行うため、実際のタイムアウト時間は timeout × 4 秒です。

8. <ファストフォワーディングの有効化>

```
nxr120(config)#fast-forwarding enable
```

ファストフォワーディングを有効にします。ファストフォワーディングを設定することによりパケット転送の高速化を行うことができます。

- (注) ファストフォワーディングの詳細および利用時の制約については、NXR,WXR シリーズのユーザーズガイド(CLI 版)に記載されているファストフォワーディングの解説をご参照ください。

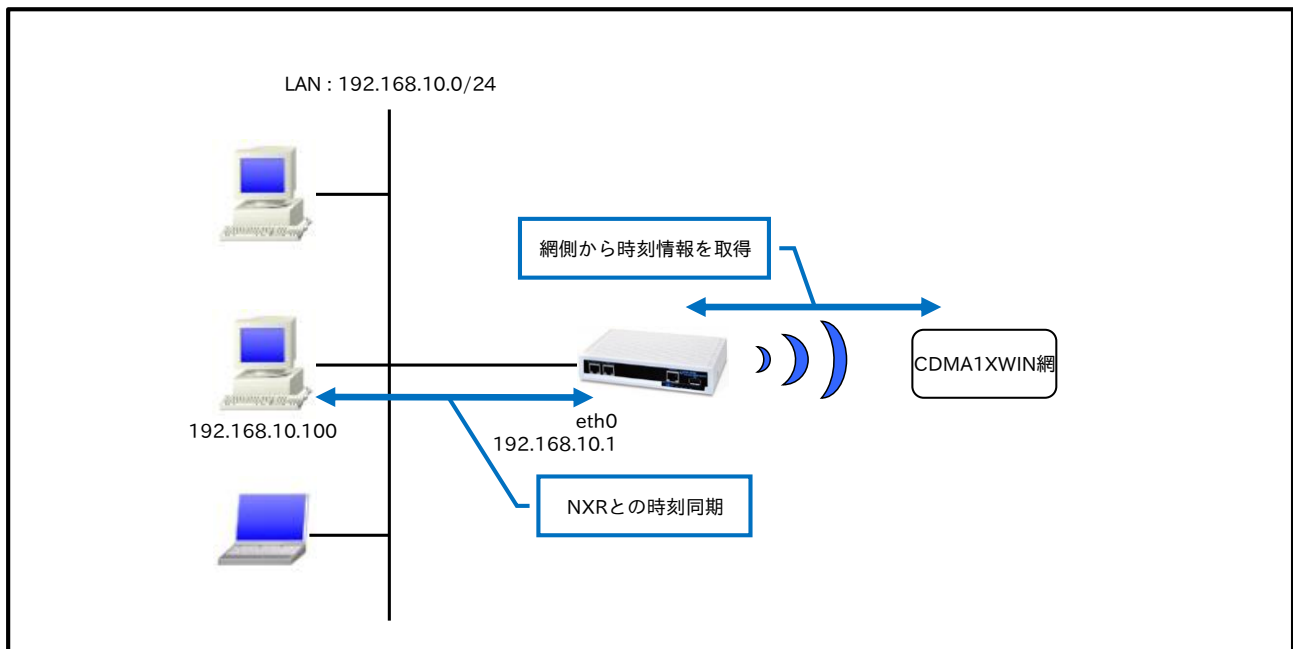
【 端末の設定例 】

IP アドレス	192.168.10.100
サブネットマスク	255.255.255.0
デフォルトゲートウェイ	192.168.10.1
DNS サーバ	
NTP サーバ	

7-2. WIN 通信モジュールを利用した時刻同期設定

NXR-155/C-XW に内蔵されている WIN 通信モジュールを利用することにより、CDMA 網から取得した時刻情報を、システム時刻に反映することができます。ここでは CDMA 網から取得した時刻情報をシステム時刻に反映し、かつ NTP クライアントからのリクエストに応答する設定について説明します。

【 構成図 】



- ・ NXR-155/C-XW に内蔵されている WIN 通信モジュールを使用した PPP 接続設定例は、弊社ホームページにあります FutureNet NXR,WXR 設定例集 WAN インタフェース編をご参照ください。

【 設定パラメータ 】

設定項目		設定内容
LAN 側インタフェース	ethernet0 の IP アドレス	192.168.10.1/24
自動時刻同期		mobile2
NTP		有効

【 設定例 】

```

nxr155#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
nxr155(config)#interface ethernet 0
nxr155(config-if)#ip address 192.168.10.1/24
nxr155(config-if)#exit
nxr155(config)#system clock mobile 2
nxr155(config)#ntp
nxr155(config-ntp)#service enable
nxr155(config-ntp)#exit
nxr155(config)#exit
nxr155#save config

```

【 設定例解説 】

1. <LAN 側(ethernet0)インタフェース設定>

```
nxr155(config)#interface ethernet 0
nxr155(config-if)#ip address 192.168.10.1/24
```

LAN 側(ethernet0)インタフェースの IP アドレスとして 192.168.10.1/24 を設定します。

2. <自動時刻同期設定>

```
nxr155(config)#system clock mobile 2
```

通信モジュールが CDMA 網側から取得した時刻情報を、システム時刻に反映するよう設定します。

3. <NTP 設定>

```
nxr155(config)#ntp
```

NTP を設定します。

```
nxr155(config-ntp)#service enable
```

NTP サービスを有効にします。

(注) NTP 機能と自動時刻同期機能を有効にすると、NTP クライアントからのリクエストに応答することができます。

【 端末の設定例 】

IP アドレス	192.168.10.100
サブネットマスク	255.255.255.0
NTP サーバ	192.168.10.1

8. メール送信設定

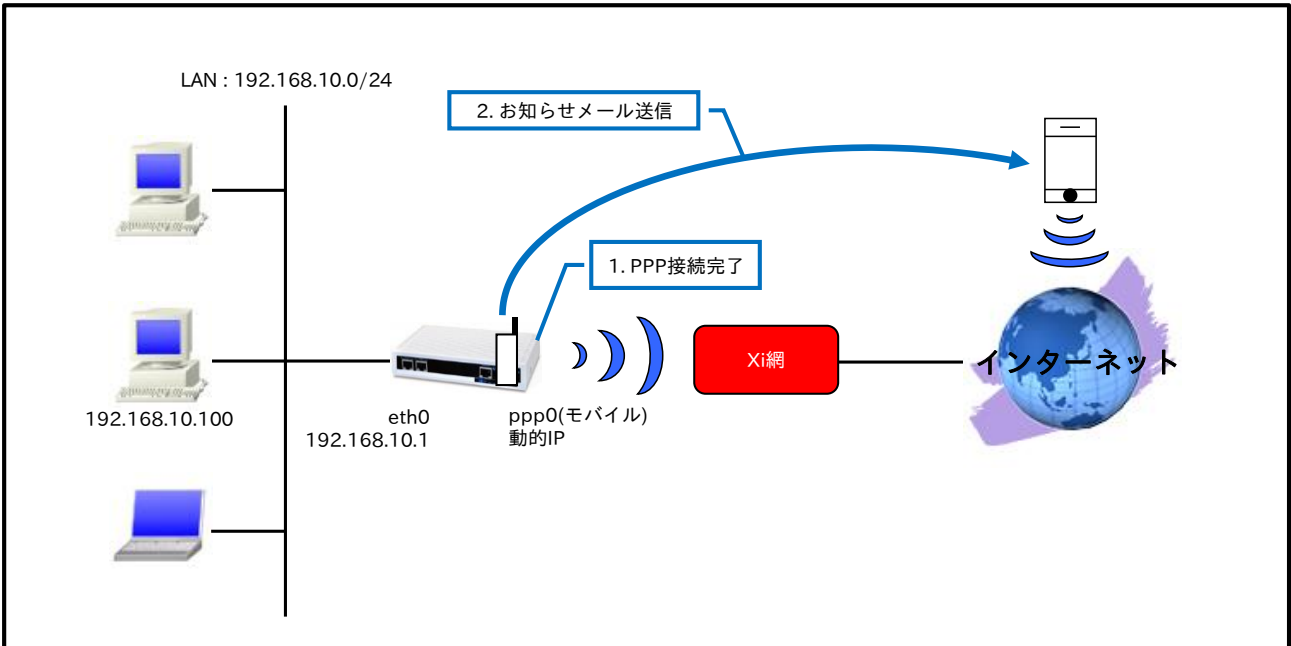
8-1. PPP 接続/切断時のメール送信設定

8-2. ログメール送信設定

8-1. PPP 接続/切断時のメール送信設定

メール送信機能は、イベント発生時に指定した宛先に対してメールを送信する機能です。
この設定例では、PPP 回線の接続時にメールを送信します。

【 構成図 】



- ・ ppp0 インタフェース接続時に指定した宛先にメールアドレスを送信します。なお、メール本文には IP アドレスが記載されます。
 - (☞) PPP 切断時、他の経路がない等の理由でメールを送信できない場合、メール送信に失敗し切断時のメールは届きません。
- ・ ppp0 インタフェースは常時接続とします。
 - (☞) PPP オンデマンド接続が有効な場合、メール送信機能は無効となり動作しません。

【 設定パラメータ 】

	設定項目	設定内容
LAN 側インタフェース	ethernet0 の IP アドレス	192.168.10.1/24
WAN 側インタフェース	mobile0	ppp0
	ppp0 の IP アドレス	動的 IP アドレス
	IP マスカレード	有効
	SPI	有効
	MSS 自動調整	オート
	IP リダイレクト	無効
	PPP 接続用ユーザ ID	lte
	PPP 接続用パスワード	lte
	APN	mopera.net
	PDP タイプ	IP
	電話番号	*99**1#
	ダイヤルタイムアウト	30 秒

スタティックルート	宛先 IP アドレス	0.0.0.0/0
	ゲートウェイ(インタフェース)	ppp0
メール送信	認証方式	smtp-auth-login
	SMTP サーバ	smtp.example.jp
	SMTP ポート番号	587
	ユーザ名	nxr
	パスワード	testpass
	送信元メールアドレス	nxr@example.jp
	宛先メールアドレス	admin@example.jp
LED	ext0	mobile0 の電波状態表示
DNS	サービス	有効

【 設定例 】

```

nxr120#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
nxr120(config)#interface ethernet 0
nxr120(config-if)#ip address 192.168.10.1/24
nxr120(config-if)#exit
nxr120(config)#mail server 1
nxr120(config-mail-server)#server authentication smtp-auth-login
nxr120(config-mail-server)#server smtp address smtp.example.jp
nxr120(config-mail-server)#server smtp port 587
nxr120(config-mail-server)#username nxr password testpass
nxr120(config-mail-server)#exit
nxr120(config)#ip route 0.0.0.0/0 ppp 0
nxr120(config)#interface ppp 0
nxr120(config-ppp)#ip address negotiated
nxr120(config-ppp)#ip masquerade
nxr120(config-ppp)#ip spi-filter
nxr120(config-ppp)#ip tcp adjust-mss auto
nxr120(config-ppp)#no ip redirects
nxr120(config-ppp)#ppp username lte password lte
nxr120(config-ppp)#mobile apn mopera.net cid 1 pdp-type ip
nxr120(config-ppp)#dial-up string *99**1#
nxr120(config-ppp)#dial-up timeout 30
nxr120(config-ppp)#mail send server 1
nxr120(config-ppp)#mail send to admin@example.jp
nxr120(config-ppp)#mail send from nxr@example.jp
nxr120(config-ppp)#exit
nxr120(config)#mobile error-recovery-reset
nxr120(config)#mobile 0 ppp 0
nxr120(config)#system led ext 0 signal-level mobile 0
nxr120(config)#dns
nxr120(config-dns)#service enable
nxr120(config-dns)#exit
nxr120(config)#exit
nxr120#save config

```

【 設定例解説 】

1. <LAN 側(ethernet0)インタフェース設定>

```

nxr120(config)#interface ethernet 0
nxr120(config-if)#ip address 192.168.10.1/24

```

LAN 側(ethernet0)インタフェースの IP アドレスを設定します。

2. <メールサーバ設定>

```
nxr120(config)#mail server 1
```

メールサーバ 1 を設定します。

```
nxr120(config-mail-server)#server authentication smtp-auth-login
```

SMTP サーバの認証方式を設定します。

```
nxr120(config-mail-server)#server smtp address smtp.example.jp
```

SMTP サーバを設定します。

```
nxr120(config-mail-server)#server smtp port 587
```

SMTP のポート番号を設定します。

(☞) サブミッションポートの利用を想定し、587 番ポートを設定します。

```
nxr120(config-mail-server)#username nxr password testpass
```

SMTP サーバの認証用 ID とパスワードを設定します。

3. <スタティックルート設定>

```
nxr120(config)#ip route 0.0.0.0/0 ppp 0
```

デフォルトルートを設定します。

4. <WAN 側(ppp0)インタフェース設定>

```
nxr120(config)#interface ppp 0  
nxr120(config-ppp)#ip address negotiated
```

WAN 側(ppp0)インタフェースを設定します。IP アドレスとして negotiated を設定します。

```
nxr120(config-ppp)#ip masquerade  
nxr120(config-ppp)#ip spi-filter  
nxr120(config-ppp)#ip tcp adjust-mss auto  
nxr120(config-ppp)#no ip redirects
```

IP マスカレード、ステートフルパケットインスペクションを有効に設定します。

また、TCP MSS の調整機能をオート、ICMP リダイレクト機能を無効に設定します。

```
nxr120(config-ppp)#ppp username lte password lte
```

PPP 接続で使用するユーザ ID とパスワードを設定します。

(☞) Xi データプランでは通常ユーザ ID,パスワードは任意となります。

```
nxr120(config-ppp)#mobile apn mopera.net cid 1 pdp-type ip  
nxr120(config-ppp)#dial-up string *99**1#  
nxr120(config-ppp)#dial-up timeout 30
```

PPP 接続に使用する APN,CID,pdp-type、電話番号、ダイアルタイムアウトを設定します。

```
nxr120(config-ppp)#mail send server 1
```

メールサーバとして先ほど設定したメールサーバ 1 を設定します。

```
nxr120(config-ppp)#mail send to admin@example.jp
```

送信先メールアドレスを設定します。

```
nxr120(config-ppp)#mail send from nxr@example.jp
```

送信元メールアドレスを設定します。

5. <モバイルエラーリカバリー設定>

```
nxr120(config)#mobile error-recovery-reset
```

モバイルデータ通信端末との通信に重大な問題が発生する可能性が高いと判断した場合、モバイルデータ通信端末のリセットを行うように設定します。

6. <モバイル割り当て設定>

```
nxr120(config)#mobile 0 ppp 0
```

mobile0 と認識されているモバイルデータ通信端末と ppp0 インタフェースの関連づけを行います。

(☞) mobile0 に割り当てられているモバイルデータ通信端末の情報は show mobile 0 コマンドで確認することができます。

7. <システム LED 設定>

```
nxr120(config)#system led ext 0 signal-level mobile 0
```

モバイルデータ通信端末の電波状態を AUX LED1,2 の点灯/消灯で表示するように設定します。

8. <DNS 設定>

```
nxr120(config)#dns
nxr120(dns-config)#service enable
```

DNS 設定で DNS サービスを有効にします。

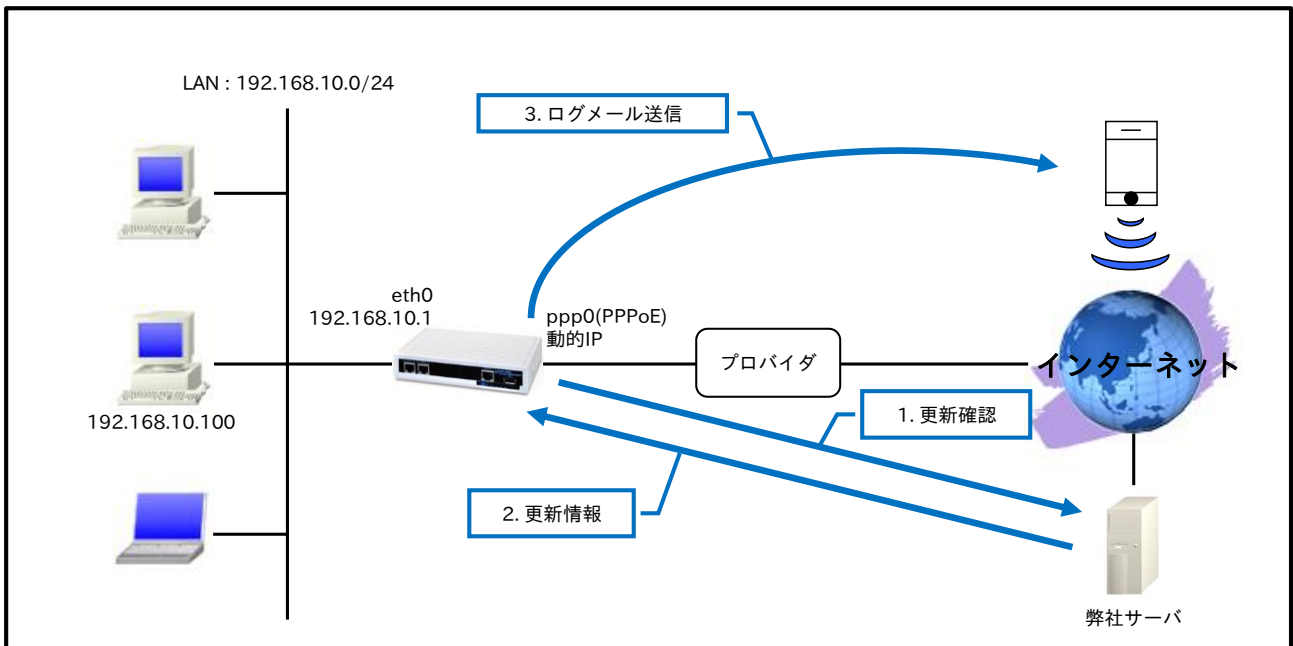
【 端末の設定例 】

IP アドレス	192.168.10.100
サブネットマスク	255.255.255.0
デフォルトゲートウェイ	192.168.10.1
DNS サーバ	

8-2. ログメール送信設定

ログメール送信機能は、指定した文字列が含まれるシスログが出力された場合、そのログを指定した宛先にメールで送信することができます。

【 構成図 】



- ・ 「scheduler:」をキーワードとし、NXR-G100 シリーズで対応している最新ファームウェアの更新状況のチェック結果をメールで送信するように設定します。
- (☑) ファームウェアの更新状況のみメールで送信しますので、更新も合わせて行う場合は別途スケジュール設定を追加する必要があります。
- (☑) ログメール機能は、NXR-G100 シリーズ以外の NXR シリーズでも利用することができます。

【 設定パラメータ 】

	設定項目	設定内容
LAN 側インタフェース	ethernet0 の IP アドレス	192.168.10.1/24
WAN 側インタフェース	PPPoE クライアント(ethernet1)	ppp0
	ppp0 の IP アドレス	動的 IP アドレス
	IP マスカレード	有効
	SPI	有効
	MSS 自動調整	オート
	IP リダイレクト	無効
	ISP 接続用ユーザ ID	test1@example.jp
	ISP 接続用パスワード	test1pass
スタティックルート	宛先 IP アドレス	0.0.0.0/0
	ゲートウェイ(インタフェース)	ppp0
シスログ	シスログサービス	有効
	メール送信機能	有効
メール送信	認証方式	smtp-auth-login

	SMTP サーバ	smtp.example.jp
	SMTP ポート番号	587
	ユーザ名	nxr
	パスワード	testpass
	送信元メールアドレス	nxr@example.jp
	宛先メールアドレス	admin@example.jp
	件名	Firmware Check
	指定文字列	scheduler:
スケジュール		毎週月曜日 9:00 にファームウェアの更新情報を確認
LED	status2	ppp0 アップ時点灯
DNS	サービス	有効

【 設定例 】

```

nxrg100#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
nxrg100(config)#interface ethernet 0
nxrg100(config-if)#ip address 192.168.10.1/24
nxrg100(config-if)#exit
nxrg100(config)#ip route 0.0.0.0/0 ppp 0
nxrg100(config)#interface ppp 0
nxrg100(config-ppp)#ip address negotiated
nxrg100(config-ppp)#ip masquerade
nxrg100(config-ppp)#ip spi-filter
nxrg100(config-ppp)#ip tcp adjust-mss auto
nxrg100(config-ppp)#no ip redirects
nxrg100(config-ppp)#ppp username test1@example.jp password test1pass
nxrg100(config-ppp)#exit
nxrg100(config)#interface ethernet 1
nxrg100(config-if)#no ip address
nxrg100(config-if)#pppoe-client ppp 0
nxrg100(config-if)#exit
nxrg100(config)#system led status 2 interface ppp 0
nxrg100(config)#dns
nxrg100(config-dns)#service enable
nxrg100(config-dns)#exit
nxrg100(config)#syslog
nxrg100(config-syslog)#local enable
nxrg100(config-syslog)#mail send enable
nxrg100(config-syslog)#mail server authentication smtp-auth-login
nxrg100(config-syslog)#mail server smtp address smtp.example.jp
nxrg100(config-syslog)#mail server smtp port 587
nxrg100(config-syslog)#mail server username nxr password testpass
nxrg100(config-syslog)#mail to admin@example.jp
nxrg100(config-syslog)#mail from nxr@example.jp
nxrg100(config-syslog)#mail subject Firmware Check
nxrg100(config-syslog)#mail strings 1 scheduler:
nxrg100(config-syslog)#exit
nxrg100(config)#schedule 1 9:00 monday firmware check official
nxrg100(config)#exit
nxrg100#save config

```

【 設定例解説 】

1. <LAN 側(ethernet0)インタフェース設定>

```

nxrg100(config)#interface ethernet 0
nxrg100(config-if)#ip address 192.168.10.1/24

```

LAN 側(ethernet0)インタフェースの IP アドレスを設定します。

2. <スタティックルート設定>

```
nxrg100(config)#ip route 0.0.0.0/0 ppp 0
```

デフォルトルートを設定します。

3. <WAN 側(ppp0)インタフェース設定>

```
nxrg100(config)#interface ppp 0  
nxrg100(config-ppp)#ip address negotiated
```

WAN 側(ppp0)インタフェースを設定します。IP アドレスとして negotiated を設定します。

```
nxrg100(config-ppp)#ip masquerade  
nxrg100(config-ppp)#ip spi-filter  
nxrg100(config-ppp)#ip tcp adjust-mss auto  
nxrg100(config-ppp)#no ip redirects
```

IP マスカレード、ステートフルパケットインスペクションを有効に設定します。

また、TCP MSS の調整機能をオート、ICMP リダイレクト機能を無効に設定します。

```
nxrg100(config-ppp)#ppp username test1@example.jp password test1pass
```

ISP との接続で使用するユーザ ID とパスワードを設定します。

4. <ethernet1 インタフェース設定>

```
nxrg100(config)#interface ethernet 1  
nxrg100(config-if)#no ip address  
nxrg100(config-if)#pppoe-client ppp 0
```

PPPoE クライアントとして ppp0 インタフェースを使用できるように設定します。

5. <システム LED 設定>

```
nxrg100(config)#system led status 2 interface ppp 0
```

ppp0 インタフェースのアップ/ダウンを STATUS LED2 で表示するように設定します。

6. <DNS 設定>

```
nxrg100(config)#dns  
nxrg100(config-dns)#service enable
```

DNS 設定で DNS サービスを有効にします。

7. <シスログ設定>

```
nxrg100(config)#syslog  
nxrg100(config-syslog)#local enable
```

シスログをローカル出力します。

```
nxrg100(config-syslog)#mail send enable
```

ログメール送信を有効にします。

```
nrxgl00(config-syslog)#mail server authentication smtp-auth-login
```

SMTP サーバの認証方式を設定します。

```
nrxgl00(config-syslog)#mail server smtp address smtp.example.jp
```

SMTP サーバを設定します。

```
nrxgl00(config-syslog)#mail server smtp port 587
```

SMTP のポート番号を設定します。

(☞) サブミッションポートの利用を想定し、587 番ポートを設定します。

```
nrxgl00(config-syslog)#mail server username nxr password testpass
```

SMTP サーバの認証用 ID とパスワードを設定します。

```
nrxgl00(config-syslog)#mail to admin@example.jp
```

送信先メールアドレスを設定します。

```
nrxgl00(config-syslog)#mail from nxr@example.jp
```

送信元メールアドレスを設定します。

```
nrxgl00(config-syslog)#mail subject Firmware Check
```

メールの件名を設定します。

```
nrxgl00(config-syslog)#mail strings 1 scheduler:
```

ログメールで送信したい文字列を設定します。

8. <スケジュール設定>

```
nrxgl00(config)#schedule 1 9:00 monday firmware check official
```

毎週月曜日 9:00 に弊社ホームページに最新のファームウェアがあるか確認するように設定します。

(☞) この設定は、ファームウェアの更新状況のみ確認する設定です。更新を行う場合は別途スケジュール設定を追加する必要があります。設定方法は、[4-6. ファームウェアの更新を自動化する](#)をご参照ください。

【 端末の設定例 】

IP アドレス	192.168.10.100
サブネットマスク	255.255.255.0
デフォルトゲートウェイ	192.168.10.1
DNS サーバ	

9. Wake ON LAN 設定

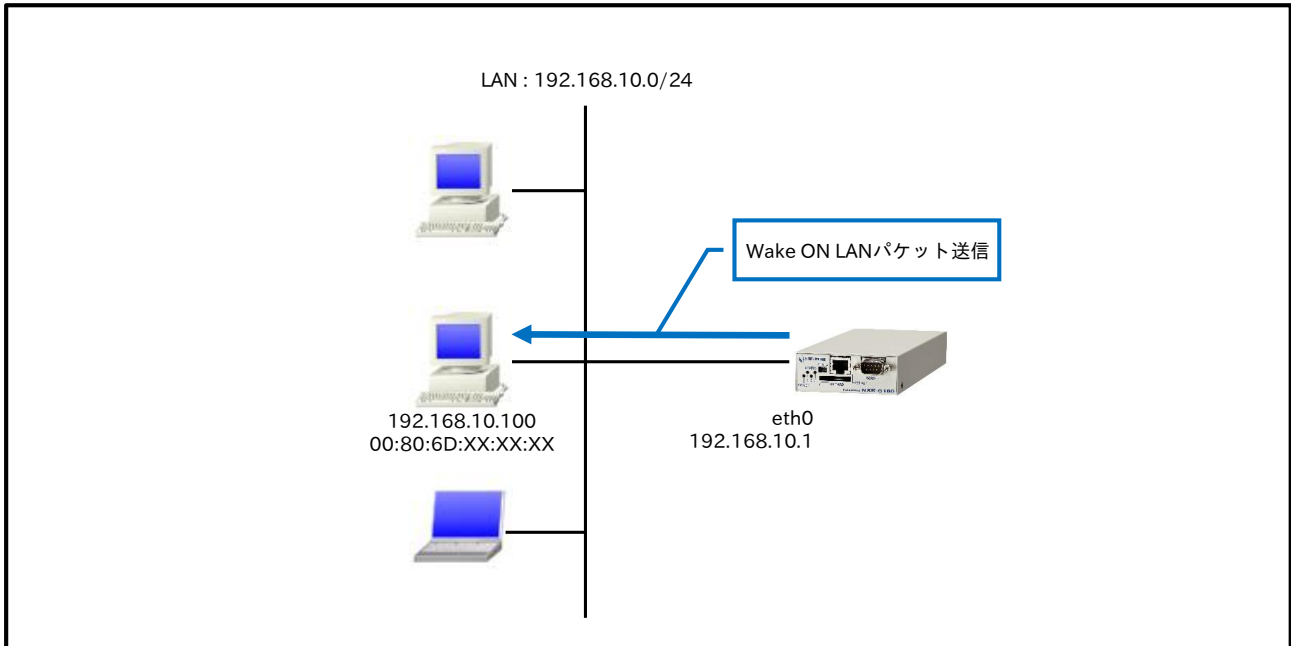
9-1. Wake ON LAN による端末起動

9-2. VPN 経由での Wake ON LAN

9-1. Wake ON LAN による端末起動

NXR-G100 シリーズでは、スリープ状態にある端末にマジックパケットを送信し、端末の電源を ON にすることができます。

【 構成図 】



- ・ Wake ON LAN(以下 WOL)で電源を ON にする端末は、下記要件を満たしている必要があります。
 - WOL 機能に対応している
 - WOL 設定が有効になっている

【 設定データ 】

設定項目		設定内容
LAN 側インタフェース	ethernet0 の IP アドレス	192.168.10.1/24
Wake ON LAN(WOL)	定義名	PC
	フレームタイプ	Ethernet
	インタフェース	ethernet0
	端末の MAC アドレス	00:80:6D:XX:XX:XX
	宛先 MAC アドレス	ブロードキャスト

【 設定例 】

```
nxrg100#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
nxrg100(config)#interface ethernet 0
nxrg100(config-if)#ip address 192.168.10.1/24
nxrg100(config-if)#exit
nxrg100(config)#wol name PC interface ethernet 0 00:80:6D:XX:XX:XX ethernet broadcast
nxrg100(config)#exit
nxrg100#save config
```

【 設定例解説 】

1. <ethernet0 インタフェース設定>

```
nxrg100(config)#interface ethernet 0  
nxrg100(config-if)#ip address 192.168.10.1/24
```

ethernet0 インタフェースの IP アドレスを設定します。

2. <Wake ON LAN 設定>

```
nxrg100(config)#wol name PC interface ethernet 0 00:80:6D:XX:XX:XX ethernet broadcast
```

定義名を PC とし、MAC アドレス 00:80:6D:XX:XX:XX の端末を起動するよう設定します。

なお、マジックパケットは ethernet0 インタフェースから宛先 MAC アドレス FF:FF:FF:FF:FF:FF(ブロードキャストアドレス)で送信します。

【 端末の設定例 】

IP アドレス	192.168.10.100
サブネットマスク	255.255.255.0

【 補足 】

wol name コマンドで定義しただけでは、マジックパケットは送信されません。定義名を指定して手動、またはスケジュールによる送信を行う必要があります。

<マジックパケットの送信>

```
nxrg100#wol send name PC
```

Wake ON LAN 設定で PC という名前で定義した端末にマジックパケットを送信します。

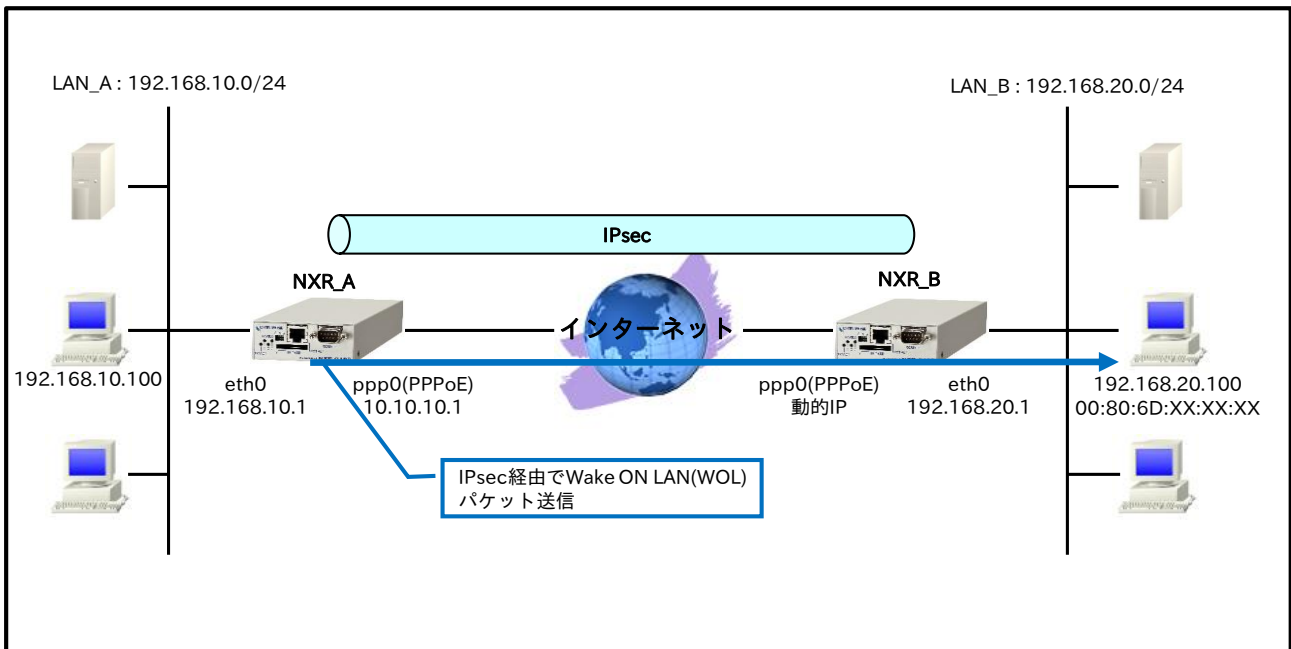
(☞) wol name コマンドであらかじめ定義しておくことでスケジュール機能と組み合わせて利用することもできますが、wol name コマンドで定義せずにマジックパケットを送信することも可能です。

```
nxrg100#wol send interface ethernet 0 00:80:6D:XX:XX:XX ethernet broadcast
```

9-2. VPN 経由での Wake ON LAN

NXR-G100 シリーズでは、マジックパケットを UDP パケットで送信することもできます。これを利用することで遠隔地にある端末の電源を VPN 経由で ON にすることができます。

【 構成図 】



- Wake ON LAN(以下 WOL)で電源を ON にする端末は、下記要件を満たしている必要があります。
 - WOL 機能に対応している
 - WOL 設定が有効になっている
- この設定例では VPN として Route Based IPsec を利用します。

【 設定パラメータ 】

設定項目		設定内容	
ホスト名		NXR_A	
LAN インタフェース	ethernet0 の IP アドレス	192.168.10.1/24	
WAN インタフェース	PPPoE クライアント(ethernet1)	ppp0	
	ppp0 の IP アドレス	10.10.10.1/32	
	IP マスカレード	有効	
	IP アクセスグループ	in	
	SPI フィルタ	有効	
	MSS 自動調整	オート	
	IP リダイレクト	無効	
	ISP 接続用ユーザ ID	test1@example.jp	
	ISP 接続用パスワード	test1pass	
	IPsec ローカルポリシー	1	
スタティックルート	No.1	宛先 IP アドレス	192.168.20.0/24
		ゲートウェイ(インタフェース)	tunnell
		ディスタンス	1
	No.2	宛先 IP アドレス	192.168.20.0/24
		ゲートウェイ(インタフェース)	null
	ディスタンス	254	

	No.3	宛先 IP アドレス	0.0.0.0/0	
		ゲートウェイ(インタフェース)	ppp0	
IP フィルタ	ルール名		ppp0_in	
	ppp0_in	No.1	動作	許可
			送信元 IP アドレス	any
			宛先 IP アドレス	10.10.10.1
			プロトコル	UDP
			送信元ポート	500
			宛先ポート	500
	No.2	動作	許可	
		送信元 IP アドレス	any	
		宛先 IP アドレス	10.10.10.1	
プロトコル		50(ESP)		
IPsec	IPsec アクセスリスト		リスト名 送信元 IP アドレス 宛先 IP アドレス	ipsec_acl any any
	IPsec ローカルポリシー1		address	ip
	IPsecISAKMP ポリシー1	名前		NXR_B
		認証方式		pre-share
		認証鍵		ipseckey
		DPD	再送間隔	30 秒
			リトライ回数	3 回
			動作	clear
		認証アルゴリズム		sha256
		暗号化アルゴリズム		aes128
		DH グループ		グループ 5
		ライフタイム		10800 秒
		ISAKMP モード		アグレッシブモード
		リモートアドレス		any
		リモート ID(fqdn)		nxb
	ローカルポリシー		1	
	IPsec トンネルポリシー1	名前		NXR_B
		ネゴシエーションモード		レスポнда
		認証アルゴリズム		sha256
		暗号化アルゴリズム		aes128
PFS		有効(グループ 5)		
ライフタイム		3600 秒		
ISAKMP ポリシー		1		
IPsec アクセスリスト		ipsec_acl		
トンネル 1 インタフェース	トンネルモード		IPsec(IPv4)	
	トンネルプロテクション		ipsec policy 1	
	MSS 自動調整		オート	
Wake ON LAN(WOL)	定義名		PC	
	パケットタイプ		UDP	
	宛先 IP アドレス		192.168.20.255	
	端末の MAC アドレス		00:80:6D:XX:XX:XX	
DNS	サービス		有効	
FastForwarding			有効	

設定項目		設定内容	
ホスト名		NXR_B	
LAN インタフェース	ethernet0 の IP アドレス	192.168.20.1/24	
	ダイレクトブロードキャスト	有効	
WAN インタフェース	PPPoE クライアント(ethernet1)		ppp0
	ppp0 の IP アドレス		動的 IP アドレス
	IP マスカレード		有効
	IP アクセスグループ	in	ppp0_in
	SPI フィルタ		有効
	MSS 自動調整		オート
	IP リダイレクト		無効
	ISP 接続用ユーザ ID		test2@example.jp

	ISP 接続用パスワード	test2pass				
	IPsec ローカルポリシー	1				
スタティックルート	No.1	宛先 IP アドレス	192.168.10.0/24			
		ゲートウェイ(インタフェース)	tunnell			
		ディスタンス	1			
	No.2	宛先 IP アドレス	192.168.10.0/24			
		ゲートウェイ(インタフェース)	null			
		ディスタンス	254			
No.3	宛先 IP アドレス	0.0.0.0/0				
	ゲートウェイ(インタフェース)	ppp0				
IP フィルタ	ルール名		ppp0_in			
	ppp0_in	No.1	動作	許可		
			送信元 IP アドレス	10.10.10.1		
			宛先 IP アドレス	any		
		プロトコル	UDP			
		送信元ポート	500			
		宛先ポート	500			
	No.2	動作	許可			
		送信元 IP アドレス	10.10.10.1			
		宛先 IP アドレス	any			
プロトコル		50(ESP)				
IPsec	IPsec アクセスリスト		リスト名	ipsec_acl		
	IPsec ローカルポリシー1		送信元 IP アドレス	any		
			宛先 IP アドレス	any		
			address	ip		
			セルフ ID(fqdn)	nrxrb		
	IPsec ISAKMP ポリシー1		名前		NXR_A	
			認証方式		pre-share	
			認証鍵		ipseckey	
			DPD	再送間隔	30 秒	
				リトライ回数	3 回	
				動作	restart	
					認証アルゴリズム	sha256
					暗号化アルゴリズム	aes128
					DH グループ	グループ 5
					ライフタイム	10800 秒
					ISAKMP モード	アグレッシブモード
					リモートアドレス	10.10.10.1
					ローカルポリシー	1
	IPsec トンネルポリシー1		名前		NXR_A	
			ネゴシエーションモード		オート	
			認証アルゴリズム		sha256	
			暗号化アルゴリズム		aes128	
			PFS		有効(グループ 5)	
			ライフタイム		3600 秒	
			ISAKMP ポリシー		1	
	IPsec アクセスリスト		ipsec_acl			
	トンネル 1 インタフェース	トンネルモード		IPsec(IPv4)		
トンネルプロテクション		ipsec policy 1				
MSS 自動調整		オート				
DNS	サービス		有効			
FastForwarding			有効			

【 設定例 】

【 NXR_A の設定 】

```

nrgl00#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
nrgl00(config)#hostname NXR_A

```

```
NXR_A(config)#interface ethernet 0
NXR_A(config-if)#ip address 192.168.10.1/24
NXR_A(config-if)#exit
NXR_A(config)#ip route 192.168.20.0/24 tunnel 1 1
NXR_A(config)#ip route 192.168.20.0/24 null 254
NXR_A(config)#ip route 0.0.0.0/0 ppp 0
NXR_A(config)#ip access-list ppp0_in permit any 10.10.10.1 udp 500 500
NXR_A(config)#ip access-list ppp0_in permit any 10.10.10.1 50
NXR_A(config)#ipsec access-list ipsec_acl ip any any
NXR_A(config)#ipsec local policy 1
NXR_A(config-ipsec-local)#address ip
NXR_A(config-ipsec-local)#exit
NXR_A(config)#ipsec isakmp policy 1
NXR_A(config-ipsec-isakmp)#description NXR_B
NXR_A(config-ipsec-isakmp)#authentication pre-share ipseckey
NXR_A(config-ipsec-isakmp)#hash sha256
NXR_A(config-ipsec-isakmp)#encryption aes128
NXR_A(config-ipsec-isakmp)#group 5
NXR_A(config-ipsec-isakmp)#lifetime 10800
NXR_A(config-ipsec-isakmp)#isakmp-mode aggressive
NXR_A(config-ipsec-isakmp)#remote address ip any
NXR_A(config-ipsec-isakmp)#remote identity fqdn nxrb
NXR_A(config-ipsec-isakmp)#keepalive 30 3 periodic clear
NXR_A(config-ipsec-isakmp)#local policy 1
NXR_A(config-ipsec-isakmp)#exit
NXR_A(config)#ipsec tunnel policy 1
NXR_A(config-ipsec-tunnel)#description NXR_B
NXR_A(config-ipsec-tunnel)#negotiation-mode responder
NXR_A(config-ipsec-tunnel)#set transform esp-aes128 esp-sha256-hmac
NXR_A(config-ipsec-tunnel)#set pfs group5
NXR_A(config-ipsec-tunnel)#set sa lifetime 3600
NXR_A(config-ipsec-tunnel)#set key-exchange isakmp 1
NXR_A(config-ipsec-tunnel)#match address ipsec_acl
NXR_A(config-ipsec-tunnel)#exit
NXR_A(config)#interface tunnel 1
NXR_A(config-tunnel)#tunnel mode ipsec ipv4
NXR_A(config-tunnel)#tunnel protection ipsec policy 1
NXR_A(config-tunnel)#ip tcp adjust-mss auto
NXR_A(config-tunnel)#exit
NXR_A(config)#interface ppp 0
NXR_A(config-ppp)#ip address 10.10.10.1/32
NXR_A(config-ppp)#ip masquerade
NXR_A(config-ppp)#ip access-group in ppp0_in
NXR_A(config-ppp)#ip spi-filter
NXR_A(config-ppp)#ip tcp adjust-mss auto
NXR_A(config-ppp)#no ip redirects
NXR_A(config-ppp)#ppp username test1@example.jp password test1pass
NXR_A(config-ppp)#ipsec policy 1
NXR_A(config-ppp)#exit
NXR_A(config)#interface ethernet 1
NXR_A(config-if)#no ip address
NXR_A(config-if)#pppoe-client ppp 0
NXR_A(config-if)#exit
NXR_A(config)#wol name PC ip 192.168.20.255 00:80:6D:XX:XX:XX
NXR_A(config)#dns
NXR_A(config-dns)#service enable
NXR_A(config-dns)#exit
NXR_A(config)#fast-forwarding enable
NXR_A(config)#exit
NXR_A#save confi
```

[NXR_B の設定]

```
nxrg100#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
nxrg100(config)#hostname NXR_B
NXR_B(config)#interface ethernet 0
NXR_B(config-if)#ip address 192.168.20.1/24
NXR_B(config-if)#ip directed-broadcast
NXR_B(config-if)#exit
NXR_B(config)#ip route 192.168.10.0/24 tunnel 1 1
NXR_B(config)#ip route 192.168.10.0/24 null 254
NXR_B(config)#ip route 0.0.0.0/0 ppp 0
NXR_B(config)#ip access-list ppp0_in permit 10.10.10.1 any udp 500 500
NXR_B(config)#ip access-list ppp0_in permit 10.10.10.1 any 50
NXR_B(config)#ipsec access-list ipsec_acl ip any any
NXR_B(config)#ipsec local policy 1
NXR_B(config-ipsec-local)#address ip
NXR_B(config-ipsec-local)#self-identity fqdn nxrb
NXR_B(config-ipsec-local)#exit
NXR_B(config)#ipsec isakmp policy 1
NXR_B(config-ipsec-isakmp)#description NXR_A
NXR_B(config-ipsec-isakmp)#authentication pre-share ipseckey
NXR_B(config-ipsec-isakmp)#hash sha256
NXR_B(config-ipsec-isakmp)#encryption aes128
NXR_B(config-ipsec-isakmp)#group 5
NXR_B(config-ipsec-isakmp)#lifetime 10800
NXR_B(config-ipsec-isakmp)#isakmp-mode aggressive
NXR_B(config-ipsec-isakmp)#remote address ip 10.10.10.1
NXR_B(config-ipsec-isakmp)#keepalive 30 3 periodic restart
NXR_B(config-ipsec-isakmp)#local policy 1
NXR_B(config-ipsec-isakmp)#exit
NXR_B(config)#ipsec tunnel policy 1
NXR_B(config-ipsec-tunnel)#description NXR_A
NXR_B(config-ipsec-tunnel)#negotiation-mode auto
NXR_B(config-ipsec-tunnel)#set transform esp-aes128 esp-sha256-hmac
NXR_B(config-ipsec-tunnel)#set pfs group5
NXR_B(config-ipsec-tunnel)#set sa lifetime 3600
NXR_B(config-ipsec-tunnel)#set key-exchange isakmp 1
NXR_B(config-ipsec-tunnel)#match address ipsec_acl
NXR_B(config-ipsec-tunnel)#exit
NXR_B(config)#interface tunnel 1
NXR_B(config-tunnel)#tunnel mode ipsec ipv4
NXR_B(config-tunnel)#tunnel protection ipsec policy 1
NXR_B(config-tunnel)#ip tcp adjust-mss auto
NXR_B(config-tunnel)#exit
NXR_B(config)#interface ppp 0
NXR_B(config-ppp)#ip address negotiated
NXR_B(config-ppp)#ip masquerade
NXR_B(config-ppp)#ip access-group in ppp0_in
NXR_B(config-ppp)#ip spi-filter
NXR_B(config-ppp)#ip tcp adjust-mss auto
NXR_B(config-ppp)#no ip redirects
NXR_B(config-ppp)#ppp username test2@example.jp password test2pass
NXR_B(config-ppp)#ipsec policy 1
NXR_B(config-ppp)#exit
NXR_B(config)#interface ethernet 1
NXR_B(config-if)#no ip address
NXR_B(config-if)#pppoe-client ppp 0
NXR_B(config-if)#exit
NXR_B(config)#dns
NXR_B(config-dns)#service enable
NXR_B(config-dns)#exit
NXR_B(config)#fast-forwarding enable
NXR_B(config)#exit
```

```
NXR_B#save config
```

【 設定例解説 】

[NXR_A の設定]

1. <ホスト名の設定>

```
nxrg100(config)#hostname NXR_A
```

ホスト名を設定します。

2. <LAN 側(ethernet0)インタフェース設定>

```
NXR_A(config)#interface ethernet 0  
NXR_A(config-if)#ip address 192.168.10.1/24
```

ethernet0 インタフェースの IP アドレスを設定します。

3. <スタティックルート設定>

```
NXR_A(config)#ip route 192.168.20.0/24 tunnel 1 1  
NXR_A(config)#ip route 192.168.20.0/24 null 254
```

LAN_B 向けのルートを設定します。なお、IPsec SA 確立時はトンネル 1 インタフェースを、未確立時は null インタフェースのルートを利用するように設定します。

- (☞) null インタフェースを出力インタフェースとして設定した場合、パケットが出力されることはありません(ドロップされます)。
- (☞) インタフェース名の後の数字はディスタンス値です。

```
NXR_A(config)#ip route 0.0.0.0/0 ppp 0
```

デフォルトルートを設定します。

4. <IP アクセスリスト設定>

```
NXR_A(config)#ip access-list ppp0_in permit any 10.10.10.1 udp 500 500  
NXR_A(config)#ip access-list ppp0_in permit any 10.10.10.1 50
```

IP アクセスリスト名を ppp0_in とし、NXR_A の WAN 側 IP アドレス 10.10.10.1 宛の IKE パケット (UDP ポート 500 番)、ESP パケット(プロトコル番号 50)を許可します。

なお、この IP アクセスリスト設定は ppp0 インタフェース設定で登録します。

- (☞) IP アクセスリストを設定しただけではフィルタとして有効にはなりません。フィルタリングしたいインタフェースでの登録が必要になります。

5. <IPsec アクセスリスト設定>

```
NXR_A(config)#ipsec access-list ipsec_acl ip any any
```

ipsec_acl という名前の IPsec アクセスリストを設定します。

- (☞) Route Based IPsec で ESP 化するか否かは、IPsec アクセスリストではなくトンネルインタフェース

をゲートウェイとするルート設定の有無で決まります。

6. <IPsec ローカルポリシー設定>

```
NXR_A(config)#ipsec local policy 1  
NXR_A(config-ipsec-local)#address ip
```

IPsec トンネルの送信元 IP アドレスを ip(IPv4)と設定します。

7. <IPsec ISAKMP ポリシー設定>

```
NXR_A(config)#ipsec isakmp policy 1  
NXR_A(config-ipsec-isakmp)#description NXR_B  
NXR_A(config-ipsec-isakmp)#authentication pre-share ipseckey
```

ISAKMP ポリシーの説明として NXR_B、認証方式として pre-share(事前共有鍵)を選択し事前共有鍵 ipseckey を設定します。なお、事前共有鍵は NXR_B と共通の値を設定します。

```
NXR_A(config-ipsec-isakmp)#hash sha256  
NXR_A(config-ipsec-isakmp)#encryption aes128  
NXR_A(config-ipsec-isakmp)#group 5  
NXR_A(config-ipsec-isakmp)#lifetime 10800  
NXR_A(config-ipsec-isakmp)#isakmp-mode aggressive
```

認証アルゴリズムとして sha256、暗号化アルゴリズムとして aes128、Diffie-Hellman(DH)グループとして group 5、ISAKMP SA のライフタイムとして 10800 秒、フェーズ 1 のネゴシエーションモードとして アグレッシブモードを設定します。

```
NXR_A(config-ipsec-isakmp)#remote address ip any  
NXR_A(config-ipsec-isakmp)#remote identity fqdn nxrb  
NXR_A(config-ipsec-isakmp)#keepalive 30 3 periodic clear  
NXR_A(config-ipsec-isakmp)#local policy 1
```

NXR_B の WAN 側 IP アドレスが動的 IP アドレスのため、リモートアドレスを any とします。また、リモート ID を FQDN 方式で nxrb とし、NXR_B のセルフ ID と同じ ID を設定します。そして、IKE KeepAlive(DPD)を監視間隔 30 秒、リトライ回数 3 回とし keepalive 失敗時に SA を削除するよう設定します。さらに、IPsec ローカルポリシー1 と関連づけを行います。

8. <IPsec トンネルポリシー設定>

```
NXR_A(config)#ipsec tunnel policy 1  
NXR_A(config-ipsec-tunnel)#description NXR_B  
NXR_A(config-ipsec-tunnel)#negotiation-mode responder
```

IPsec トンネルポリシーの説明として NXR_B、ネゴシエーションモードとして responder を設定します。

```
NXR_A(config-ipsec-tunnel)#set transform esp-aes128 esp-sha256-hmac  
NXR_A(config-ipsec-tunnel)#set pfs group5  
NXR_A(config-ipsec-tunnel)#set sa lifetime 3600
```

暗号化アルゴリズムとして aes128、認証アルゴリズムとして sha256、PFS を有効とし、かつ DH グループとして group5、IPsec SA のライフタイムとして 3600 秒を設定します。

```
NXR_A(config-ipsec-tunnel)#set key-exchange isakmp 1
```

```
NXR_A(config-ipsec-tunnel)#match address ipsec_acl
```

ISAKMP ポリシー1 と関連づけを行い、IPsec アクセスリスト ipsec_acl を設定します。

9. <トンネルインタフェース設定>

```
NXR_A(config)#interface tunnel 1  
NXR_A(config-tunnel)#tunnel mode ipsec ipv4  
NXR_A(config-tunnel)#tunnel protection ipsec policy 1  
NXR_A(config-tunnel)#ip tcp adjust-mss auto
```

トンネル 1 インタフェースでトンネルモードを ipsec ipv4、使用するトンネルポリシーとして 1 を設定します。また、TCPMSS 値をオートに設定します。

10. <WAN 側(ppp0)インタフェース設定>

```
NXR_A(config)#interface ppp 0  
NXR_A(config-ppp)#ip address 10.10.10.1/32
```

ppp0 インタフェースの IP アドレスを設定します。

```
NXR_A(config-ppp)#ip masquerade  
NXR_A(config-ppp)#ip access-group in ppp0_in  
NXR_A(config-ppp)#ip spi-filter  
NXR_A(config-ppp)#ip tcp adjust-mss auto  
NXR_A(config-ppp)#no ip redirects
```

IP マスカレード、ステートフルパケットインスペクションを有効に設定します。また、IP アクセスリスト ppp0_in を in フィルタに適用します。そして、TCP MSS の調整機能をオート、ICMP リダイレクト機能を無効に設定します。

```
NXR_A(config-ppp)#ppp username test1@example.jp password test1pass  
NXR_A(config-ppp)#ipsec policy 1
```

ISP 接続用ユーザ ID とパスワードを設定します。また、IPsec トンネルのエンドポイントとなるため IPsec ローカルポリシー1 を設定します。

11. <ethernet1 インタフェース設定>

```
NXR_A(config)#interface ethernet 1  
NXR_A(config-if)#no ip address  
NXR_A(config-if)#pppoe-client ppp 0
```

PPPoE クライアントとして ppp0 インタフェースを使用できるように設定します。

12. <Wake ON LAN 設定>

```
NXR_A(config)#wol name PC ip 192.168.20.255 00:80:6D:XX:XX:XX
```

定義名を PC とし、MAC アドレス 00:80:6D:XX:XX:XX の端末を起動するよう設定します。

なお、マジックパケットは宛先 IP アドレス 192.168.20.255(ブロードキャストアドレス)で送信します。

13. <DNS 設定>

```
NXR_A(config)#dns
```

```
NXR_A(config-dns)#service enable
```

DNS サービスを有効にします。

14. <ファストフォワーディングの有効化>

```
NXR_A(config)#fast-forwarding enable
```

ファストフォワーディングを有効にします。ファストフォワーディングを設定することによりパケット転送の高速化を行うことができます。

(☞) ファストフォワーディングの詳細および利用時の制約については、NXR,WXR シリーズのユーザーズガイド(CLI 版)に記載されているファストフォワーディングの解説をご参照ください。

[NXR_B の設定]

1. <ホスト名の設定>

```
nxrg100(config)#hostname NXR_B
```

ホスト名を設定します。

2. <LAN 側(ethernet0)インタフェース設定>

```
NXR_B(config)#interface ethernet 0  
NXR_B(config-if)#ip address 192.168.20.1/24
```

ethernet0 インタフェースの IP アドレスを設定します。

```
NXR_B(config-if)#ip directed-broadcast
```

ダイレクトブロードキャストを有効にします。

(☞) ダイレクトブロードキャストを有効にすることで、ダイレクトブロードキャストを宛先とするパケットを転送することができます。

3. <スタティックルート設定>

```
NXR_B(config)#ip route 192.168.10.0/24 tunnel 1 1  
NXR_B(config)#ip route 192.168.10.0/24 null 254
```

LAN_A 向けのルートを設定します。なお、IPsec SA 確立時はトンネル 1 インタフェースを、未確立時は null インタフェースのルートを利用するように設定します。

(☞) null インタフェースを出力インタフェースとして設定した場合、パケットが出力されることはありません(ドロップされます)。

```
NXR_B(config)#ip route 0.0.0.0/0 ppp 0
```

デフォルトルートを設定します。

4. <IP アクセスリスト設定>

```
NXR_B(config)#ip access-list ppp0_in permit 10.10.10.1 any udp 500 500  
NXR_B(config)#ip access-list ppp0_in permit 10.10.10.1 any 50
```

IP アクセスリスト名を ppp0_in とし、送信元が NXR_A の WAN 側 IP アドレス 10.10.10.1 の IKE パケット(UDP ポート 500 番)、ESP パケット(プロトコル番号 50)を許可します。

なお、この IP アクセスリスト設定は ppp0 インタフェース設定で登録します。

(※) IP アクセスリストを設定しただけではフィルタとして有効にはなりません。フィルタリングしたいインタフェースでの登録が必要になります。

5. <IPsec アクセスリスト設定>

```
NXR_B(config)#ipsec access-list ipsec_acl ip any any
```

ipsec_acl という名前の IPsec アクセスリストを設定します。なお、送信元 IP アドレス、宛先 IP アドレスともに any とします。

6. <IPsec ローカルポリシー設定>

```
NXR_B(config)#ipsec local policy 1  
NXR_B(config-ipsec-local)#address ip  
NXR_B(config-ipsec-local)#self-identity fqdn nxrb
```

IPsec トンネルの送信元 IP アドレスを ip(IPv4)と設定します。また、セルフ ID を FQDN 方式で nxrb とし、NXR_A のリモート ID と同じ ID を設定します。

7. <IPsec ISAKMP ポリシー設定>

```
NXR_B(config)#ipsec isakmp policy 1  
NXR_B(config-ipsec-isakmp)#description NXR_A  
NXR_B(config-ipsec-isakmp)#authentication pre-share ipseckey
```

ISAKMP ポリシーの説明として NXR_A、認証方式として pre-share(事前共有鍵)を選択し事前共有鍵 ipseckey を設定します。なお、事前共有鍵は NXR_A と共通の値を設定します。

```
NXR_B(config-ipsec-isakmp)#hash sha256  
NXR_B(config-ipsec-isakmp)#encryption aes128  
NXR_B(config-ipsec-isakmp)#group 5  
NXR_B(config-ipsec-isakmp)#lifetime 10800  
NXR_B(config-ipsec-isakmp)#isakmp-mode aggressive
```

認証アルゴリズムとして sha256、暗号化アルゴリズムとして aes128、Diffie-Hellman(DH)グループとして group 5、ISAKMP SA のライフタイムとして 10800 秒、フェーズ 1 のネゴシエーションモードとして アグレッシブモードを設定します。

```
NXR_B(config-ipsec-isakmp)#remote address ip 10.10.10.1  
NXR_B(config-ipsec-isakmp)#keepalive 30 3 periodic restart  
NXR_B(config-ipsec-isakmp)#local policy 1
```

リモートアドレスに NXR_A の WAN 側 IP アドレスを設定します。また、IKE KeepAlive(DPD)を監視間隔 30 秒、リトライ回数 3 回とし keepalive 失敗時に SA を削除し IKE のネゴシエーションを開始するように設定します。そして、IPsec ローカルポリシー 1 と関連づけを行います。

8. <IPsec トンネルポリシー設定>

```
NXR_B(config)#ipsec tunnel policy 1  
NXR_B(config-ipsec-tunnel)#description NXR_A  
NXR_B(config-ipsec-tunnel)#negotiation-mode auto
```

IPsec トンネルポリシーの説明として NXR_A、ネゴシエーションモードとして auto を設定します。

```
NXR_B(config-ipsec-tunnel)#set transform esp-aes128 esp-sha256-hmac  
NXR_B(config-ipsec-tunnel)#set pfs group5  
NXR_B(config-ipsec-tunnel)#set sa lifetime 3600
```

暗号化アルゴリズムとして aes128、認証アルゴリズムとして sha256、PFS を有効とし、かつ DH グループとして group5、IPsec SA のライフタイムとして 3600 秒を設定します。

```
NXR_B(config-ipsec-tunnel)#set key-exchange isakmp 1  
NXR_B(config-ipsec-tunnel)#match address ipsec_acl
```

ISAKMP ポリシー1 と関連づけを行い、IPsec アクセスリスト ipsec_acl を設定します。

9. <トンネルインタフェース設定>

```
NXR_B(config)#interface tunnel 1  
NXR_B(config-tunnel)#tunnel mode ipsec ipv4  
NXR_B(config-tunnel)#tunnel protection ipsec policy 1  
NXR_B(config-tunnel)#ip tcp adjust-mss auto
```

トンネル 1 インタフェースでトンネルモードを ipsec ipv4、使用するトンネルポリシーとして 1 を設定します。また、TCP MSS 値をオートに設定します。

10. <WAN 側(ppp0)インタフェース設定>

```
NXR_B(config)#interface ppp 0  
NXR_B(config-ppp)#ip address negotiated
```

ppp0 インタフェースの IP アドレスが動的 IP アドレスの場合は、negotiated を設定します。

```
NXR_B(config-ppp)#ip masquerade  
NXR_B(config-ppp)#ip access-group in ppp0_in  
NXR_B(config-ppp)#ip spi-filter  
NXR_B(config-ppp)#ip tcp adjust-mss auto  
NXR_B(config-ppp)#no ip redirects
```

IP マスカレード、ステートフルパケットインスペクションを有効に設定します。また、IP アクセスリスト ppp0_in を in フィルタに適用します。そして、TCP MSS の調整機能をオート、ICMP リダイレクト機能を無効に設定します。

```
NXR_B(config-ppp)#ppp username test2@example.jp password test2pass  
NXR_B(config-ppp)#ipsec policy 1
```

ISP 接続用ユーザ ID とパスワードを設定します。また、IPsec トンネルのエンドポイントとなるため IPsec ローカルポリシー1 を設定します。

11. <ethernet1 インタフェース設定>

```
NXR_B(config)#interface ethernet 1  
NXR_B(config-if)#no ip address  
NXR_B(config-if)#pppoe-client ppp 0
```

PPPoE クライアントとして ppp0 インタフェースを使用できるように設定します。

12. <DNS 設定>

```
NXR_B(config)#dns  
NXR_B(config-dns)#service enable
```

DNS サービスを有効にします。

13. <ファストフォワーディングの有効化>

```
NXR_B(config)#fast-forwarding enable
```

ファストフォワーディングを有効にします。

【 端末の設定例 】

	LAN_A の端末	LAN_B の端末
IP アドレス	192.168.10.100	192.168.20.100
サブネットマスク	255.255.255.0	
デフォルトゲートウェイ	192.168.10.1	192.168.20.1
DNS サーバ		

【 補足 】

wol name コマンドで定義しただけでは、マジックパケットは送信されません。定義名を指定して手動、またはスケジュールによる送信を行う必要があります。

<マジックパケットの送信>

```
nxrg100#wol send name PC
```

Wake ON LAN 設定で PC という名前で定義した端末にマジックパケットを送信します。

(☞) wol name コマンドであらかじめ定義しておくことでスケジュール機能と組み合わせて利用することもできますが、wol name コマンドで定義せずにマジックパケットを送信することも可能です。

```
nxrg100#wol send ip 192.168.20.255 00:80:6D:XX:XX:XX
```

10. 省電力設定

10-1. 電源管理モード設定

10-2. スリープモード設定

10-1. 電源管理モード設定

NXR-G100 シリーズで実装されている電源管理モードは、利用環境によりバランスモードと M2M モードを選択することができます。

【 設定例 】

<バランスモードを利用する>

```
(config)#system power-management mode balance
```

電源管理モードをバランスモードに設定します。

(注) バランスモードは、デフォルトのモードであり、消費電力と性能のバランスを考慮したモードです。

<M2M モードを利用する>

```
(config)#system power-management mode m2m
```

電源管理モードを M2M モードに設定します。

(注) M2M モードは、消費電力を抑えるために、CPU の動作クロックを低く抑えるとともに、Ethernet のリンクスピードを最大 100Mbps に抑えます。なお、温度プロテクション機能は動作しません。

10-2. スリープモード設定

NXR-G100 シリーズの一部機種では、システムのスリープ/レジュームを設定することができます。なお、スケジュール機能でのシステムのスリープ/レジュームについては、[4-8. システムのスリープ/レジュームの日時を指定する](#)も合わせてご参照ください。

【 設定例 】

<INIT ボタンでシステムをスリープ状態にする>

```
(config)#system power-management sleep init-button
```

ルータ動作中に INIT ボタンを 3 秒間押下した場合、スリープ状態に遷移します。(デフォルト有効)

(☞) オプションで timer を設定しなかった場合、365 日間スリープ状態にする設定となります。timer を設定することでスリープ状態に遷移後、指定時間経過後レジュームさせることができます。この実行例では、スリープ状態に遷移後、3600 秒(=1 時間)後にレジュームさせます。

```
(config)#system power-management sleep init-button timer 3600
```

<コマンドでシステムをスリープ状態にする>

```
nxrg100#sleep system  
Proceed with sleep? [y/n]: y  
Exit command line ? [y/n]: y
```

```
System sleeping ...
```

コマンド実行によりスリープ状態に遷移します。

(☞) オプションで timer を設定しなかった場合、365 日間スリープ状態にする設定となります。timer を設定することでスリープ状態に遷移後、指定時間経過後レジュームさせることができます。また、スケジュール機能でレジュームさせることもできます。この実行例では、スリープ状態に遷移後、3600 秒(=1 時間)後にレジュームさせます。

```
nxrg100#sleep system timer 3600  
Proceed with sleep? [y/n]: y  
Exit command line ? [y/n]: y
```

```
System sleeping ...
```

<PPP アイドルタイムアウトによる切断時にスリープ状態にする>

```
(config)#interface ppp 0  
(config-ppp)#ppp idle-timeout 30 system sleep timer 3600
```

ppp0 インタフェースで、30 秒間データの送受信がなければ、PPP 切断時にスリープ状態に遷移します。なお、スリープ状態に遷移後、3600 秒(=1 時間)後にレジュームさせます。

(☞) PPP/PPPoE の設定例は、弊社ホームページにあります FutureNet NXR,WXR 設定例集 WAN インタフェース編をご参照ください。

11. テクニカルサポート情報・シスログの取得

11-1. テクニカルサポート情報の表示とファイル転送

11-2. シスログ情報の表示とファイル転送

11-1. テクニカルサポート情報の表示とファイル転送

テクニカルサポート情報とはルータの各機能のステータスやシスログ、設定情報(show config 形式)を一括表示・取得する機能です。この情報はトラブル発生時にお客様自身で障害を切り分ける場合や FutureNet サポートデスクへお問い合わせ頂く際に活用することができます。

(☞) テクニカルサポート情報には設定情報が含まれていますので、取得した情報の取り扱いには十分ご注意ください。

(☞) テクニカルサポート情報に含まれるシスログは直近 100 行分のみとなりますので、必要に応じてシスログ情報の表示・取得を行って下さい。なおシスログ設定は運用管理編の [5. シスログ設定](#) を、シスログ情報の表示・取得は [11-2. シスログ情報の表示とファイル転送](#) をご参照下さい。

【 実行例 】

〔CLI 上で表示〕

<テクニカルサポート情報の表示>

```
#show tech-support
```

テクニカルサポート情報を表示します。

(☞) グローバルノードで「terminal length 0」を実行すると画面単位での一時停止を行わなくなります。なお初期値に戻す場合は「terminal no length」を実行して下さい。

〔FTP サーバの利用〕

<テクニカルサポート情報の転送>

```
#copy tech-support ftp://192.168.10.100/tech-support.txt  
Succeed.
```

テクニカルサポート情報をファイルとして FTP サーバ 192.168.10.100 上に転送します。なおこの実行例では、ファイル名を tech-support.txt とします。また FTP サーバを利用して転送する場合、anonymous による接続方法のみ対応しています。よってユーザ名やパスワードを指定することはできません。コマンド実行後、テクニカルサポート情報ファイルを転送します。

(☞) サーバ内でテクニカルサポート情報ファイルをディレクトリ毎に管理している場合は、下記のように IP アドレスの後に「ディレクトリ名/ファイル名」を指定します。

```
#copy tech-support ftp://192.168.10.100/nxrg100techsupport/tech-support.txt
```

〔SSH サーバの利用〕

<テクニカルサポート情報の転送>

```
#copy tech-support ssh://test@192.168.10.100/tech-support.txt  
test@192.168.10.100's password:[パスワード]  
Succeed.
```

テクニカルサポート情報をファイルとして SSH サーバ 192.168.10.100 上に転送します。なおこの実行例

ではファイル名を tech-support.txt とします。この時 SSH サーバの IP アドレスとともにログイン用のユーザ名を合わせて指定します。この実行例では、ユーザ名を test とします。コマンド実行後、SSH サーバへのログイン用パスワードの入力を求められますので、パスワードを入力します。ログイン後、テクニカルサポート情報ファイルを転送します。

(☞) サーバ内でテクニカルサポート情報ファイルをディレクトリ毎に管理している場合は、下記のように IP アドレスの後に「ディレクトリ名/ファイル名」を指定します。

```
#copy tech-support ssh://test@192.168.10.100/nxrg100techsupport/tech-support.txt
```

〔外部ストレージの利用〕

<テクニカルサポート情報の転送>

```
#copy tech-support disk0:tech-support.txt  
Succeed.
```

テクニカルサポート情報をファイルとして disk0 と認識された外部ストレージに転送します。なおこの実行例ではファイル名を tech-support.txt とします。

(☞) 外部ストレージでテクニカルサポート情報ファイルをディレクトリ毎に管理している場合は、下記のように「ディレクトリ名/ファイル名」を指定します。

```
#copy tech-support disk0:nxrg100techsupport/tech-support.txt
```

〔Web 設定画面から取得〕

Web 設定画面からテクニカルサポート情報をファイルとして取得する場合、予め HTTP サービスを起動しておく必要があります。HTTP サービスの起動コマンドは以下のとおりです。

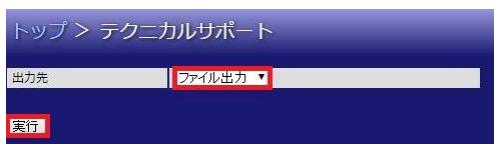
```
(config)#http-server enable
```

<テクニカルサポート情報の転送>

[運用機能]-[システム情報]-[テクニカルサポート]をクリックしてテクニカルサポート取得画面を開きます。



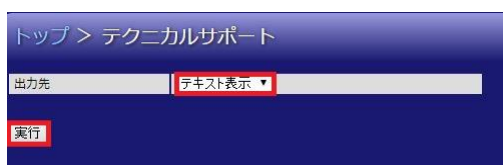
出力先に「ファイル出力」を指定し、「実行」をクリックします。



(※) 出力先として外部ストレージを指定することもできます。

<テクニカルサポート情報の表示>

出力先に「テキスト表示」を指定し、「実行」をクリックします。



11-2. シスログ情報の表示とファイル転送

シスログ情報はトラブル発生時にお客様自身で障害を切り分ける場合や FutureNet サポートデスクへお問い合わせ頂く際に活用することができます。

(☞) シスログ情報を取得するためにはシスログが有効に設定されている必要があります。シスログ設定は運用管理編の [5. シスログ設定](#) をご参照下さい。

【 実行例 】

〔CLI 上で表示〕

<シスログ情報の表示>

```
#show syslog message
```

シスログ情報を表示します。

(☞) グローバルノードで「terminal length 0」を実行すると画面単位での一時停止を行わなくなります。なお初期値に戻す場合は「terminal no length」を実行して下さい。

〔FTP サーバの利用〕

<シスログファイルの転送>

```
#copy syslog ftp://192.168.10.100/syslog.txt  
Succeed.
```

シスログファイルを FTP サーバ 192.168.10.100 上に転送します。なおこの実行例では、ファイル名を syslog.txt とします。また FTP サーバを利用して転送する場合、anonymous による接続方法のみ対応しています。よってユーザ名やパスワードを指定することはできません。コマンド実行後、シスログファイルを転送します。

(☞) サーバ内でシスログファイルをディレクトリ毎に管理している場合は、下記のように IP アドレスの後に「ディレクトリ名/ファイル名」を指定します。

```
#copy syslog ftp://192.168.10.100/nxrg100syslog/syslog.txt
```

〔SSH サーバの利用〕

<シスログファイルの転送>

```
#copy syslog ssh://test@192.168.10.100/syslog.txt  
test@192.168.10.100's password:[パスワード]  
Succeed.
```

シスログファイルを SSH サーバ 192.168.10.100 上に転送します。なおこの実行例ではファイル名を syslog.txt とします。この時 SSH サーバの IP アドレスとともにログイン用のユーザ名を合わせて指定します。この実行例では、ユーザ名を test とします。コマンド実行後、SSH サーバへのログイン用パスワードの入力を求められますので、パスワードを入力します。ログイン後、シスログファイルを転送します。

(☞) サーバ内でシスログファイルをディレクトリ毎に管理している場合などは、下記のように IP アドレス

の後に「ディレクトリ名/ファイル名」を指定します。

```
#copy syslog ssh://test@192.168.10.100/nxrg100syslog/syslog.txt
```

〔外部ストレージの利用〕

<シスログファイルの転送>

```
#copy syslog disk0:syslog.txt
succeed.
```

シスログファイルを disk0 と認識された外部ストレージに転送します。なおこの実行例ではファイル名を syslog.txt とします。

(☞) 外部ストレージでシスログファイルをディレクトリ毎に管理している場合は、下記のように「ディレクトリ名/ファイル名」を指定します。

```
#copy syslog disk0:nxrg100syslog/syslog.txt
```

〔Web 設定画面から取得〕

Web 設定画面からシスログファイルを取得するためには、予め HTTP サービスを起動しておく必要があります。HTTP サービスの起動コマンドは以下のとおりです。

```
(config)#http-server enable
```

<シスログファイルの取得>

[運用機能]-[ログ情報]-[システムログ]をクリックして、シスログ取得画面を開きます。



出力先に「ファイル出力」を指定し、「実行」をクリックします。



(☞) 出力先として外部ストレージを指定することもできます。

<シスログ情報の表示>

出力先に「テキスト表示」を指定し、「実行」をクリックします。



(注) シスログ情報は最大 1000 行表示されます。

12. モニタログ設定

12-1. モニタログ(監視結果の保存)

12-2. モニタログファイルの表示とファイル転送

12-1. モニタログ(監視結果の保存)

モニタログとはネットイベント機能にある Ping/Ping6 監視(拡張機能のみ)の結果やシステムリソース情報をモニタログファイルに保存する機能です。

【 設定例 】

〔Ping 監視結果の保存〕

192.168.10.100 への Ping 監視結果をモニタログ機能で保存します。

(☞) モニタログの出力情報の見方については NXR,WXR シリーズのユーザーズガイド(CLI 版)に記載されているモニタログの解説をご参照ください。

1. <拡張 Ping 監視設定>

```
(config)#track 2048 ip reachability
(config-track)#destination 192.168.10.100
(config-track)#netevent monitor-log-status
```

拡張 Ping 監視設定で宛先 192.168.10.100 を設定します。また監視結果をモニタログ機能で保存するように設定します。

2. <モニタログ設定>

```
(config)#monitor-log
(config-monitor-log)#reachability disk0 threshold logsize 200 files 10
```

Ping/Ping6 監視(拡張機能のみ)の結果をモニタログ機能で保存します。またバックアップは disk0 と認識された外部ストレージに保存します。なおこの設定例ではログファイルのサイズを 200Kbyte、バックアップファイル数を 10 とします。

3. <指定した時刻にモニタログの定期バックアップを実行する>

```
(config)#schedule 1 4:00 sunday monitor-log reachability rotate
```

毎週日曜日 4:00 にモニタログ(Ping/Ping6 監視)のローテートを行います。

(☞) 定期バックアップ以外にも監視ログ出力時に閾値のチェックを行います。チェックの結果、条件に達している場合はバックアップを行います。

〔システムリソース監視結果の保存〕

機器のシステムリソース情報を 3 分毎に監視し、結果をモニタログ機能で保存します。

(☞) モニタログの出力情報の見方については NXR,WXR シリーズのユーザーズガイド(CLI 版)に記載されているモニタログの解説をご参照ください。

1. <モニタログ設定>

```
(config)#monitor-log
(config-monitor-log)#resource disk0 threshold logsize 200 files 10
```


システムリソース情報の結果をモニタログ機能で保存します。またバックアップは disk0 と認識された外部

ストレージに保存します。なおこの設定例ではログファイルのサイズを 200Kbyte、バックアップファイル数を 10 とします。

2. <指定した時刻にモニタログの定期バックアップを実行する>

```
(config)#schedule 1 5:00 sunday monitor-log resource rotate
```

毎週日曜日 5:00 にモニタログ(システムリソース監視)のローテートを行います。

() 定期バックアップ以外にも監視ログ出力時に閾値のチェックを行います。チェックの結果、条件に達している場合はバックアップを行います。

12-2. モニタログファイルの表示とファイル転送

モニタログファイルは show コマンドによる表示、copy コマンドによる外部転送が可能です。

【 実行例 】

【CLI 上で表示】

モニタログの出力情報の見方については NXR,WXR シリーズのユーザーズガイド(CLI 版)に記載されているモニタログの解説をご参照ください。

<Ping/Ping6 監視結果の表示>

```
#show monitor-log reachability
```

Ping 監視結果を表示します。

(🔗) グローバルノードで「terminal length 0」を実行すると画面単位での一時停止を行わなくなります。なお初期値に戻す場合は「terminal no length」を実行して下さい。

<システムリソース監視結果の表示>

```
#show monitor-log resource
```

システムリソース監視結果を表示します。

(🔗) グローバルノードで「terminal length 0」を実行すると画面単位での一時停止を行わなくなります。なお初期値に戻す場合は「terminal no length」を実行して下さい。

【FTP サーバの利用】

<モニタログ(Ping/Ping6 監視)ファイルの転送>

```
#copy reachability-log ftp://192.168.10.100/reachability-log.txt  
succeed.
```

モニタログ(Ping/Ping6 監視)ファイルを FTP サーバ 192.168.10.100 上に転送します。なおこの実行例では、ファイル名を reachability-log.txt とします。また FTP サーバを利用して転送する場合、anonymous による接続方法のみ対応しています。よってユーザ名やパスワードを指定することはできません。コマンド実行後、モニタログファイルを転送します。

(🔗) サーバ内でモニタログファイルをディレクトリ毎に管理している場合は、下記のように IP アドレスの後に「ディレクトリ名/ファイル名」を指定します。

```
#copy reachability-log ftp://192.168.10.100/nxrg100monitor-log/reachability-log.txt
```

<モニタログ(システムリソース監視)ファイルの転送>

```
#copy resource-log ftp://192.168.10.100/resource-log.txt  
succeed.
```

モニタログ(システムリソース監視)ファイルを FTP サーバ 192.168.10.100 上に転送します。なおこの実行例では、ファイル名を resource-log.txt とします。また FTP サーバを利用して転送する場合、anonymous

による接続方法のみ対応しています。よってユーザ名やパスワードを指定することはできません。コマンド実行後、モニタログファイルを転送します。

(☞) サーバ内でモニタログファイルをディレクトリ毎に管理している場合は、下記のように IP アドレスの後に「ディレクトリ名/ファイル名」を指定します。

```
#copy resource-log ftp://192.168.10.100/nxrg100monitor-log/resource-log.txt
```

【SSH サーバの利用】

<モニタログ(Ping/Ping6 監視)ファイルの転送>

```
#copy reachability-log ssh://test@192.168.10.100/reachability-log.txt
test@192.168.10.100's password:[パスワード]
succeed.
```

モニタログ(Ping/Ping6 監視)ファイルを SSH サーバ 192.168.10.100 上に転送します。なおこの実行例ではファイル名を reachability-log.txt とします。この時 SSH サーバの IP アドレスとともにログイン用のユーザ名を合わせて指定します。この実行例では、ユーザ名を test とします。コマンド実行後、SSH サーバへのログイン用パスワードの入力を求められますので、パスワードを入力します。ログイン後、モニタログファイルを転送します。

(☞) サーバ内でモニタログファイルをディレクトリ毎に管理している場合は、下記のように IP アドレスの後に「ディレクトリ名/ファイル名」を指定します。

```
#copy reachability-log ssh://test@192.168.10.100/nxrg100monitor-log/reachability-log.txt
```

<モニタログ(システムリソース監視)ファイルの転送>

```
#copy resource-log ssh://test@192.168.10.100/resource-log.txt
test@192.168.10.100's password:[パスワード]
succeed.
```

モニタログ(システムリソース監視)ファイルを SSH サーバ 192.168.10.100 上に転送します。なおこの実行例ではファイル名を resource-log.txt とします。この時 SSH サーバの IP アドレスとともにログイン用のユーザ名を合わせて指定します。この実行例では、ユーザ名を test とします。コマンド実行後、SSH サーバへのログイン用パスワードの入力を求められますので、パスワードを入力します。ログイン後、モニタログファイルを転送します。

(☞) サーバ内でモニタログファイルをディレクトリ毎に管理している場合は、下記のように IP アドレスの後に「ディレクトリ名/ファイル名」を指定します。

```
#copy resource-log ssh://test@192.168.10.100/nxrg100monitor-log/resource-log.txt
```

【外部ストレージの利用】

モニタログ機能では閾値チェック時のバックアップや定期バックアップで指定した外部ストレージに保存することができます。それ以外にも任意のタイミングでモニタログファイルを外部ストレージに転送することができます。

<モニタログ(Ping/Ping6 監視)ファイルの転送>

```
#copy reachability-log disk0:reachability-log.txt  
succeed.
```

モニタログ(Ping/Ping6 監視)ファイルを disk0 と認識された外部ストレージに転送します。なおこの実行例ではファイル名を reachability-log.txt とします。

(☞) 外部ストレージでモニタログファイルをディレクトリ毎に管理している場合は、下記のように「ディレクトリ名/ファイル名」を指定します。

```
#copy reachability-log disk0:nxrg100monitor-log/reachability-log.txt
```

<モニタログ(システムリソース監視)ファイルの転送>

```
#copy resource-log disk0:resource-log.txt  
succeed.
```

モニタログ(システムリソース監視)ファイルを disk0 と認識された外部ストレージに転送します。なおこの実行例ではファイル名を resource-log.txt とします。

(☞) 外部ストレージでモニタログファイルをディレクトリ毎に管理している場合は、下記のように「ディレクトリ名/ファイル名」を指定します。

```
#copy resource-log disk0:nxrg100monitor-log/resource-log.txt
```

付録

設定例 show config 形式サンプル

設定例 show config 形式サンプル

6-1. SNMP エージェント設定

```
!  
! Century Systems NXR-120 Series ver 5.24.1G (build 1/14:41 04 03 2014)  
!  
hostname nxr120  
telnet-server enable  
http-server enable  
!  
!  
!  
!  
!  
ipv6 forwarding  
fast-forwarding enable  
!  
!  
!  
interface ppp 0  
 ip address negotiated  
 no ip redirects  
 ip tcp adjust-mss auto  
 ip masquerade  
 ip spi-filter  
 ppp username test1@example.jp password test1pass  
!  
interface ethernet 0  
 ip address 192.168.10.1/24  
!  
interface ethernet 1  
 no ip address  
 pppoe-client ppp 0  
!  
dns  
 service enable  
!  
syslog  
 local enable  
!  
snmp  
 bind address 192.168.10.1  
 security 192.168.10.100/32 public  
 syslocation Honsya  
 syscontact admin  
 sysname Honsya-NXR  
 trap manager 192.168.10.100 public inform interval 10  
!  
!  
!  
no system led ext 0  
system led aux 1 interface ppp 0  
!  
!  
!  
!  
!  
!  
ip route 0.0.0.0/0 ppp 0  
!  
!  
!
```



```
end
```

7-1. NTP 設定

```
!  
! Century Systems NXR-120 Series ver 5.24.1G (build 1/14:41 04 03 2014)  
!  
hostname nxr120  
telnet-server enable  
http-server enable  
!  
!  
!  
!  
! IPv6 forwarding  
ipv6 forwarding  
fast-forwarding enable  
!  
!  
!  
interface ppp 0  
 ip address negotiated  
 no ip redirects  
 ip tcp adjust-mss auto  
 ip masquerade  
 ip spi-filter  
 ppp username test1@example.jp password test1pass  
!  
interface ethernet 0  
 ip address 192.168.10.1/24  
!  
interface ethernet 1  
 no ip address  
 pppoe-client ppp 0  
!  
ntp  
 service enable  
 server 10.100.10.1 polling 4 5  
!  
dns  
 service enable  
!  
syslog  
 local enable  
!  
!  
!  
no system led ext 0  
system led aux 1 interface ppp 0  
!  
!  
!  
!  
!  
!  
ip route 0.0.0.0/0 ppp 0  
!  
!  
!  
end
```



```
!  
!  
interface ppp 0  
 ip address negotiated  
 no ip redirects  
 ip tcp adjust-mss auto  
 ip masquerade  
 ip spi-filter  
 ppp username lte password lte  
 dial-up string *99**1#  
 dial-up timeout 30  
 mobile apn mopera.net cid 1 pdp-type ip  
 mail send server 1  
 mail send to admin@example.jp  
 mail send from nxr@example.jp  
!  
interface ethernet 0  
 ip address 192.168.10.1/24  
!  
interface ethernet 1  
 no ip address  
!  
dns  
 service enable  
!  
syslog  
 local enable  
!  
!  
mail server 1  
 server authentication smtp-auth-login  
 server smtp address smtp.example.jp  
 server smtp port 587  
 username nxr password testpass  
!  
mobile 0 ppp 0  
mobile error-recovery-reset  
!  
system led ext 0 signal-level mobile 0  
!  
!  
!  
!  
!  
!  
ip route 0.0.0.0/0 ppp 0  
!  
!  
!  
end
```

8-2. ログメール送信設定

```
!  
! Century Systems NXR-G100 Series ver 6.1.3 (build 4/14:05 09 06 2014)  
!   DIP-SW : 1:off 2:off 3:off 4:off  
!  
hostname nxrg100  
telnet-server enable  
http-server enable  
!  
!  
!
```

```
!  
ipv6 forwarding  
!  
!  
!  
interface ppp 0  
 ip address negotiated  
 no ip redirects  
 ip tcp adjust-mss auto  
 ip masquerade  
 ip spi-filter  
 ppp username test1@example.jp password test1pass  
!  
interface ethernet 0  
 ip address 192.168.10.1/24  
!  
interface ethernet 1  
 no ip address  
 pppoe-client ppp 0  
!  
dns  
 service enable  
!  
syslog  
 local enable  
 mail send enable  
 mail to admin@example.jp  
 mail from nxr@example.jp  
 mail subject Firmware Check  
 mail strings 1 scheduler:  
 mail server authentication smtp-auth-login  
 mail server smtp address smtp.example.jp  
 mail server smtp port 587  
 mail server username nxr password testpass  
!  
!  
!  
system led status 2 interface ppp 0  
!  
!  
!  
!  
!  
schedule 1 09:00 monday firmware check official  
!  
ip route 0.0.0.0/0 ppp 0  
!  
!  
end
```

9-1. Wake ON LAN による端末起動

```
!  
! Century Systems NXR-G100 Series ver 6.4.3 (build 11/12:47 17 10 2014)  
!   DIP-SW : 1:off 2:off 3:off 4:off  
!  
hostname nxrg100  
telnet-server enable  
http-server enable  
!  
!  
system power-management mode balance  
!
```

```
!  
!  
ipv6 forwarding  
no fast-forwarding enable  
!  
!  
!  
!  
!  
interface ethernet 0  
  ip address 192.168.10.1/24  
!  
interface ethernet 1  
  no ip address  
!  
dns  
  service enable  
!  
syslog  
  local enable  
!  
!  
!  
!  
!  
!  
!  
!  
wol name PC interface ethernet 0 00:80:6D:XX:XX:XX ethernet broadcast  
!  
!  
end
```

9-2. VPN 経由での Wake ON LAN

〔NXR_A の設定〕

```
!  
! Century Systems NXR-G100 Series ver 6.4.3 (build 11/12:47 17 10 2014)  
!   DIP-SW : 1:off 2:off 3:off 4:off  
!  
hostname NXR_A  
telnet-server enable  
http-server enable  
!  
!  
system power-management mode balance  
!  
!  
!  
ipv6 forwarding  
fast-forwarding enable  
!  
!  
!  
!  
ipsec local policy 1  
  address ip  
!  
!  
ipsec isakmp policy 1
```



```
ip access-list ppp0_in permit any 10.10.10.1 udp 500 500
ip access-list ppp0_in permit any 10.10.10.1 50
!
ipsec access-list ipsec_acl ip any any
!
!
end
```

【NXR_B の設定】

```
!
! Century Systems NXR-G100 Series ver 6.4.3 (build 11/12:47 17 10 2014)
!   DIP-SW : 1:off 2:off 3:off 4:off
!
hostname NXR_B
telnet-server enable
http-server enable
!
!
system power-management mode balance
!
!
!
ipv6 forwarding
fast-forwarding enable
!
!
!
!
ipsec local policy 1
  address ip
  self-identity fqdn nxrb
!
!
ipsec isakmp policy 1
  description NXR_A
  authentication pre-share ipseckey
  hash sha256
  encryption aes128
  group 5
  isakmp-mode aggressive
  remote address ip 10.10.10.1
  local policy 1
!
!
ipsec tunnel policy 1
  description NXR_A
  set transform esp-aes128 esp-sha256-hmac
  set pfs group5
  set key-exchange isakmp 1
  match address ipsec_acl
!
!
interface tunnel 1
  no ip address
  ip tcp adjust-mss auto
  tunnel mode ipsec ipv4
  tunnel protection ipsec policy 1
!
interface ppp 0
  ip address negotiated
  no ip redirects
```

```
ip tcp adjust-mss auto
ip access-group in ppp0_in
ip masquerade
ip spi-filter
ppp username test2@example.jp password test2pass
ipsec policy 1
!
interface ethernet 0
 ip address 192.168.20.1/24
 ip directed-broadcast
!
interface ethernet 1
 no ip address
 pppoe-client ppp 0
!
dns
 service enable
!
syslog
 local enable
!
!
!
!
!
!
!
!
!
!
!
ip route 192.168.10.0/24 tunnel 1
ip route 192.168.10.0/24 null 254
ip route 0.0.0.0/0 ppp 0
!
ip access-list ppp0_in permit 10.10.10.1 any udp 500 500
ip access-list ppp0_in permit 10.10.10.1 any 50
!
ipsec access-list ipsec_acl ip any any
!
!
end
```

サポートデスクへのお問い合わせ

サポートデスクへのお問い合わせに関して

サポートデスクのご利用に関して

サポートデスクへのお問い合わせに関して

サポートデスクにお問い合わせ頂く際は、以下の情報をお知らせ頂けると効率よく対応させて頂くことが可能ですので、ご協力をお願い致します。

※FutureNet サポートデスク宛にご提供頂きました情報は、製品のお問合せなどサポート業務以外の目的には利用致しません。

なおご提供頂く情報の取り扱いについて制限等がある場合には、お問い合わせ時または事前にその旨ご連絡下さい。(設定ファイルのプロバイダ情報や IPsec の事前共有鍵情報を削除してお送り頂く場合など)

弊社のプライバシーポリシーについては下記 URL の内容をご確認下さい。

<http://www.centurysys.co.jp/company/philosophy.html#tab3>

<http://www.centurysys.co.jp/company/philosophy.html#tab4>

■ ご利用頂いている NXR,WXR 製品を含むネットワーク構成図

(ご利用頂いている回線やルータを含むネットワーク機器の IP アドレスを記載したもの)

■ 障害・不具合の内容およびその再現手順

(いつどこで何を行った場合にどのような問題が発生したのかをできるだけ具体的にお知らせ下さい)

□ 問い合わせ内容例 1

○月○日○時○分頃より拠点 A と拠点 B の間で IPsec による通信ができなくなった。障害発生前までは問題なく利用可能だった。現在当該拠点のルータの LAN 側 IP アドレスに対して Ping による疎通は確認できたが、対向ルータの LAN 側 IP アドレス、配下の端末に対しては Ping による疎通は確認できない。障害発生前後で拠点 B のバックアップ回線としてモバイルカードを接続し、ppp1 インタフェースの設定を行った。設定を元に戻すと通信障害は解消する。

機器の内蔵時計は NTP で同期を行っている。

□ 問い合わせ内容例 2

- 発生日時

○月○日○時○分頃

- 発生拠点

拠点 AB 間

- 障害内容

IPsec による通信ができなくなった。

- 切り分け内容

ルータ配下の端末から当該拠点のルータの LAN 側 IP アドレスに対して Ping による疎通確認可能。

対向ルータの LAN 側 IP アドレス、配下の端末に対しては Ping による疎通確認不可。

- 障害発生前後での作業

ルータの設定変更やネットワークに影響する作業は行っていない。

- 備考

障害発生前までは問題なく利用可能だった。

機器の内蔵時計は拠点 A の機器で 10 分、拠点 B の機器で 5 分遅れている。

□ 問い合わせ内容例 3

現在 IPsec の設定中だが、一度も IPsec SA の確立および IPsec の通信ができていない。IPsec を設定している拠点からのインターネットアクセスおよび該当拠点への Ping による疎通確認も可能。
設定例集および設定例集内のログ一覧は未確認。

□ 良くない問い合わせ内容例 1

VPN ができない。

→VPN として利用しているプロトコルは何か。VPN のトンネルが確立できないのか、通信ができないのかなど不明。

□ 良くない問い合わせ内容例 2

通信ができない。

→どのような通信がいつどこでできない(またはできなくなった)のかが不明。

NXR,WXR での情報取得方法は以下のとおりです。

※情報を取得される前に

シリアル接続で情報を取得される場合は取得前に下記コマンドを実行してください。

#terminal width 180(初期値に戻す場合は terminal no width)

■ ご利用頂いている NXR,WXR 製品での不具合発生時のログ

ログは以下のコマンドで出力されます。

#show syslog message

■ ご利用頂いている NXR,WXR 製品のテクニカルサポート情報の結果

テクニカルサポート情報は以下のコマンドで出力されます。

show tech-support

■ 障害発生時のモバイル関連コマンドの実行結果(モバイルカード利用時のみ)

#show mobile <N> ap

#show mobile <N> phone-number

#show mobile <N> signal-level

※<N>はモバイルデバイスナンバ

サポートデスクのご利用に関して

電話サポート

電話番号：0422-37-8926

電話での対応は以下の時間帯で行います。

月曜日 ～ 金曜日 10:00 - 17:00

ただし、国の定める祝祭日、弊社の定める年末年始は除きます。

電子メールサポート

E-mail： support@centurysys.co.jp

FAXサポート

FAX 番号：0422-55-3373

電子メール、FAX は 毎日 24 時間受け付けております。

ただし、システムのメンテナンスやビルの電源点検のため停止する場合があります。その際は弊社ホームページ等にて事前にご連絡いたします。

FutureNet NXR,WXR シリーズ

設定例集

運用管理編

Ver 1.4.0

2015 年 12 月

発行 センチュリー・システムズ株式会社

Copyright(c) 2009-2015 Century Systems Co., Ltd. All Rights Reserved.