
FutureNet NXR,WXR 設定例集

運用管理編

Ver 1.0.0

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.....	15
2. 設定の保存・復帰.....	16
2-1. 本体への設定保存 1.....	17
2-2. 本体への設定保存 2(拡張フラッシュの利用).....	18
2-3. 設定のバックアップ 1.....	20
2-4. 設定のバックアップ 2(拡張フラッシュの利用).....	23
2-5. 設定の復帰 1.....	27
2-6. 設定の復帰 2(拡張フラッシュの利用).....	29
2-7. 起動時の設定ファイル指定.....	32
2-8. 本体内の設定の削除.....	33
2-9. 設定の初期化.....	34
3. パケットダンプ.....	35
3-1. パケットダンプを pcap 形式で取得する.....	36
3-2. パケットダンプの外部転送.....	37
3-3. パケットダンプの削除.....	38
4. スケジュール設定.....	39
4-1. PPP/PPPoE を定期的に接続/切断する.....	40
4-2. モバイル通信端末を定期的にリセットする.....	41
4-3. 機器本体を定期的に再起動する.....	42
4-4. シスログを定期的にローテートする.....	43
4-5. ファームウェアの更新日時を指定する.....	44
4-6. 日時指定例.....	45
5. シスログ設定.....	46
5-1. シスログの取得.....	47
5-2. シスログサーバへの転送.....	48
5-3. シスログのローテート.....	49
6. SNMP 設定.....	50

6-1. SNMP エージェント設定	51
7. 時刻同期設定	55
7-1. NTP 設定	56
7-2. WIN 通信モジュールを利用した時刻同期設定	60
付録	62
設定例 show config 形式サンプル	63
サポートデスクへのお問い合わせ	66
サポートデスクへのお問い合わせに関して	67
サポートデスクのご利用に関して	69

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-WM,NXR-155/C-XW,NXR-155/C-L,
NXR-230/C,NXR-350/C,NXR-1200,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 シリーズ,NXR-230/C 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 7 章 FutureNet NXR シリーズ NXR-120/C Ver5.24.1G
※2-2,2-4,2-6,2-7,2-8 は FutureNet NXR シリーズ NXR-230/C Ver5.25.2
7-2 は FutureNet NXR シリーズ NXR-155/C-XW Ver5.22.5D
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。

改版履歷

Version	更新內容
1.0.0	初版

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

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あるファームウェアを転送して、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、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、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,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は、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は起動しないよう指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、ここで指定可能なアクションは機種により異なります。

<NXR-120/C の場合>

```
[=====] 100% DOWNLOAD  
Proceed with update? [y/n]: y  
After the firmware is updated, it reboots...
```

y を指定して実行した場合、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 を指定して実行した場合、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]:
```

y を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、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)ackgroud/(n)o]:
```

y を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、ファームウェアの更新を中止します。

<NXR-230/C,NXR-350/C,WXR-250 の場合>

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

y を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、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]:
```

y を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、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。この場合は、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。

[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,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は、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は起動しないように指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、ここで指定可能なアクションは機種により異なります。

<NXR-120/C の場合>

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

y を指定して実行した場合、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 を指定して実行した場合、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]:
```

y を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、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]:
```

y を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、ファームウェアの更新を中止します。

<NXR-230/C,NXR-350/C,WXR-250 の場合>

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

y を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、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]:
```

y を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、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。この場合は、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。

【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 ポートの機種では USB に接続した場合、外部ストレージを disk0 として認識します。また USB ポートが 2 ポートの機種では 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 として認識します。

```
#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 として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,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は、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は起動しないよう指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>

ファームウェアのデコード完了後、アップデートの実行について問われますが、ここで指定可能なアクションは機種により異なります。

<NXR-120/C の場合>

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

After the firmware is updated, it reboots...
```

y を指定して実行した場合、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 を指定して実行した場合、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]:
```

y を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、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)ackgroud/(n)o]:
```

y を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、ファームウェアの更新を中止します。

<NXR-230/C,NXR-350/C,WXR-250 の場合>

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

y を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、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]:
```

y を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、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 を指定して実行した場合、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。この場合は、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,WXR-250 に対応

【 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](#) をご参照ください。

2. 設定の保存・復帰

2-1. 本体への設定保存 1

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,WXR-250 に対応

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。

(☞) 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 flash:config.tar.gz all  
succeed.
```

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

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。(2014/4 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。(2014/4 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,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,WXR-250 に対応

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,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。

(☞) 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。

(☞) 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。

(☞) 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す。

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

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

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。

- (☞) 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。

(☞) 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の更新時、ファームウェアのバージョンチェックを行います。同じバージョンの場合は、ファームウェアの更新を行いません。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. 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

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。

(~~例~~) 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

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もできます。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 】

```
(config)#schedule 1 4:00 sunday syslog rotate  
(config)#syslog  
(config-syslog)#rotate disk0 threshold files 10
```

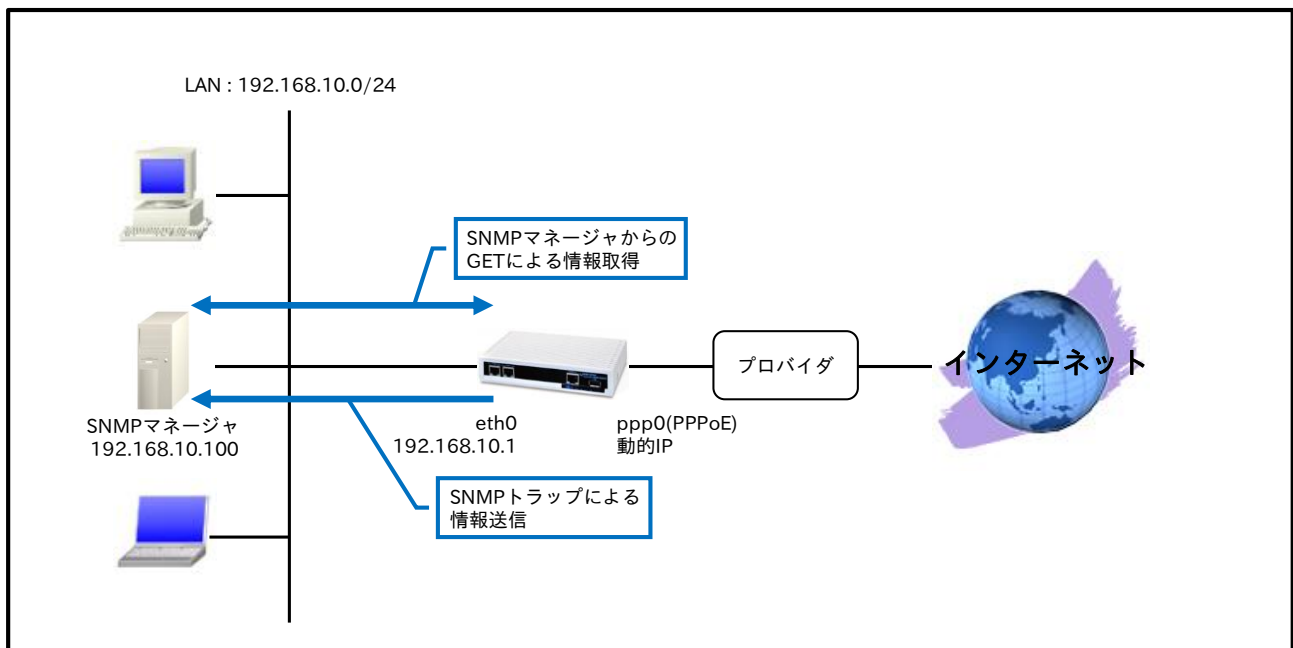
毎週日曜日 4:00 にローテートを実行し、disk0 と認識された外部ストレージにログファイルをバックアップします。なお、この設定例ではバックアップファイル数を 10 とします。

6. SNMP 設定

6-1. SNMP エージェント設定

SNMP 機能は、システムの情報を SNMP プロトコルを使用して取得する機能です。また、システム上において状態変化が発生した際に、トラップを送信することも可能です。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 】



- ・ SNMP での情報取得、SNMP トラップの送信ができるようにします。なお SNMP トラップについては SNMP Inform を設定し、Inform リクエストを送信できるようにします。
- ・ ppp0 インタフェースで IP マスカレードを設定し、出力パケットの送信元 IP アドレスを変換します。これによりルータ配下の複数台の端末からインターネットアクセスが可能になります。
- ・ ステートフルパケットインスペクションを利用し、インターネット側からのアクセスを破棄しながらも、ルータ配下の端末からは自由にアクセスできるようにします。
- ・ DNS 機能を有効にすることにより、ルータ配下の端末からの名前解決要求(クエリ要求)を、プロバイダより取得した DNS サーバに転送します。
- ・ PPPoE 接続設定については、弊社ホームページにあります FutureNet NXR,WXR 設定例集 WAN インタフェース編も合わせてご参照ください。

【 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 アドレスとして 192.168.10.1/24 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。ゲートウェイとして ppp0 インタフェースを設定します。

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 アドレスとして動的 IP アドレスの場合は 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
```

プロバイダ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
```

ethernet1 インタフェースで、ppp0 インタフェースを PPPoE クライアントとして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
```

ここでは ppp0 インタフェースの回線接続時に AUX LED1 を点灯する設定をします。

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 マネージャとして 192.168.10.100/32, SNMP コミュニティ名として public 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。

(🔍) 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 アドレスとして 192.168.10.100,SNMP トラップのコミュニティ名として public 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。ここでは再送間隔を 10 秒,再送回数を 3 回に設定します。

(🔍) 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 アドレスとして 192.168.10.1 を設定します。

(🔍) 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。

10. <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。

(🔍) 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 】

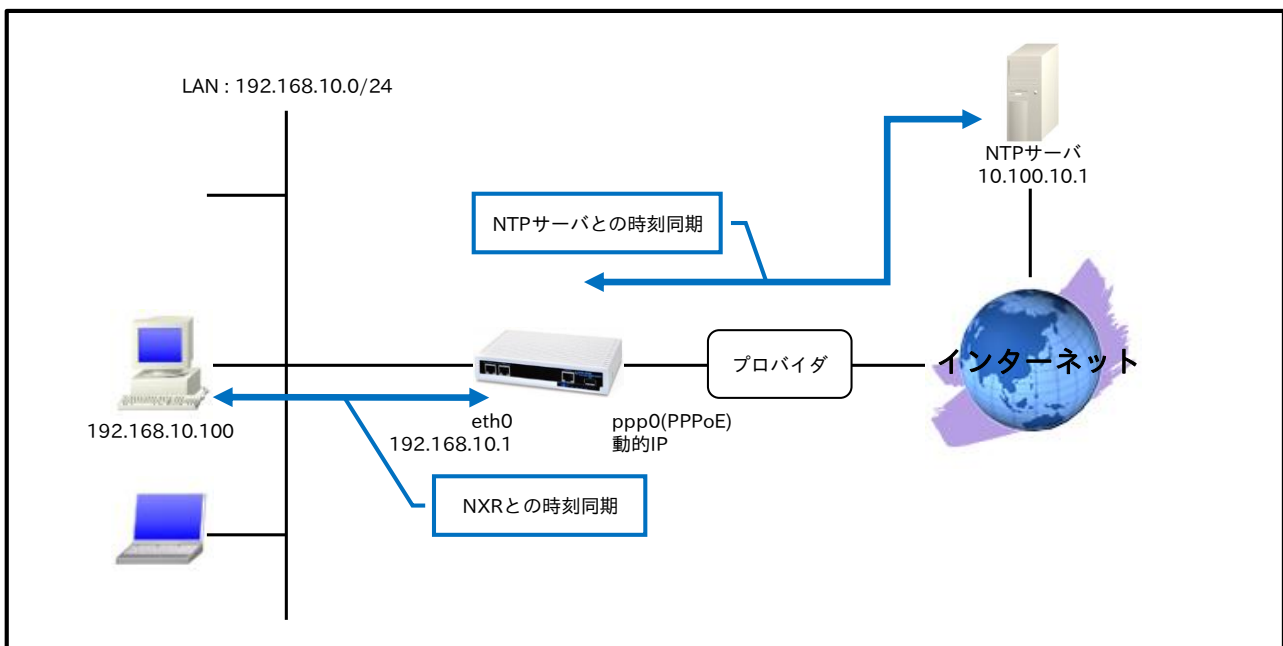
	パソコン	SNMP マネージャ
IP アドレス	192.168.10.200	192.168.10.100
サブネットマスク	255.255.255.0	255.255.255.0
デフォルトゲートウェイ	192.168.10.1	
DNS サーバの IP アドレス	192.168.10.1	

7. 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。
- ppp0 インタフェースは常時接続とします。
- ppp0 インタフェースで IP マスカレードを設定し、出力パケットの送信元 IP アドレスを変換します。これによりルータ配下の複数台の端末からインターネットアクセスが可能になります。
- ステートフルパケットインスペクションを利用し、インターネット側からのアクセスを破棄しながらも、ルータ配下の端末からは自由にアクセスできるようにします。
- DNS 機能を有効にすることにより、ルータ配下の端末からの名前解決要求(クエリ要求)を、プロバイダより取得した DNS サーバに転送します。
- PPPoE 接続設定については、弊社ホームページにあります FutureNet NXR,WXR 設定例集 WAN インタフェース編も合わせてご参照ください。

【 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 アドレスとして 192.168.10.1/24 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。ゲートウェイとして ppp0 インタフェースを設定します。

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 アドレスとして動的 IP アドレスの場合は 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。

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

プロバイダ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
```

ethernet1 インタフェースで、ppp0 インタフェースを PPPoE クライアントとして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
```

ここでは ppp0 インタフェースの回線接続時に AUX LED1 を点灯する設定をします。

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 サーバアドレスとして 10.100.10.1、ポーリング間隔として最小 4,最大 5 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として 30 秒を設定します。

(☞) 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。

-
- (※) 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。

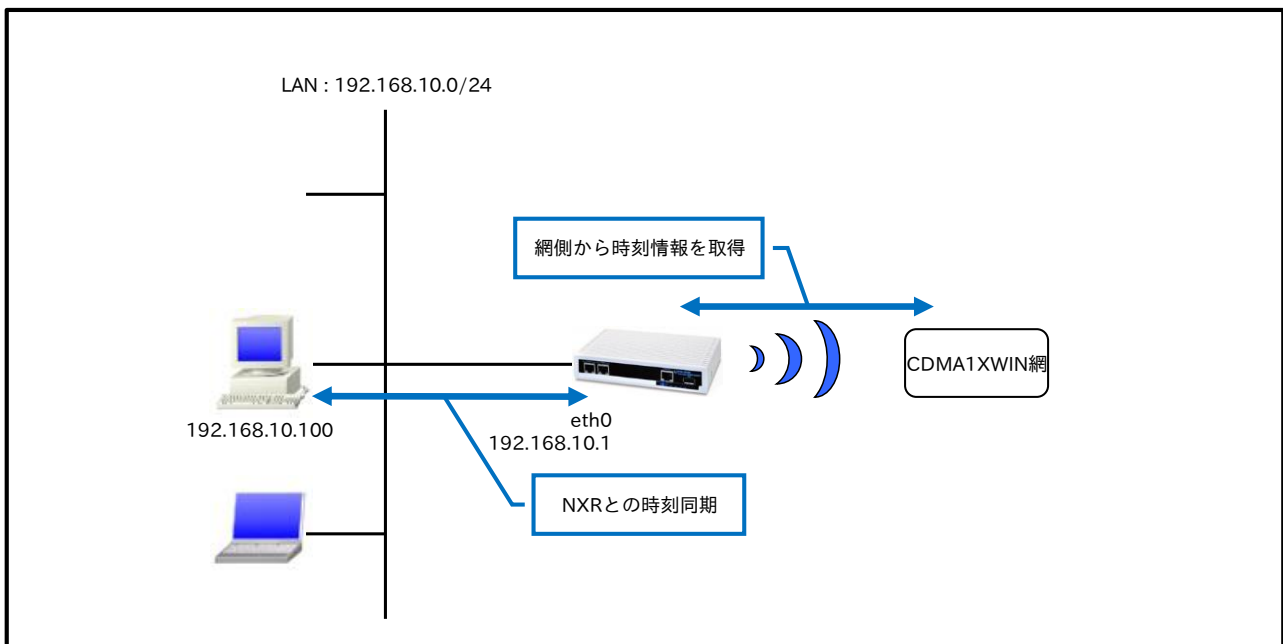
【 端末の設定例 】

	パソコン
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 サーバの IP アドレス	192.168.10.1
NTP サーバの IP アドレス	192.168.10.1

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 】



- ・ 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。
- ・ この設定例では時刻同期設定のみ行っています。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。

【 設定例 】

```

nrx155#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
nrx155(config)#interface ethernet 0
nrx155(config-if)#ip address 192.168.10.1/24
nrx155(config-if)#exit
nrx155(config)#system clock mobile 2
nrx155(config)#ntp
nrx155(config-ntp)#service enable
nrx155(config-ntp)#exit
nrx155(config)#exit
nrx155#save config

```

【 設定例解説 】

1. <LAN 側(ethernet0)インタフェース設定>

```

nrx155(config)#interface ethernet 0
nrx155(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>

```

nrx155(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>

```

nrx155(config)#ntp

```

NTP を設定します。

```

nrx155(config-ntp)#service enable

```

NTP サービスを有効にします。

(※) 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 】

	パソコン
IP アドレス	192.168.10.200
サブネットマスク	255.255.255.0
NTP サーバの IP アドレス	192.168.10.1

付録

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  
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
```

7-2. WIN 通信モジュールを利用した時刻同期設定

```
!  
! Century Systems NXR-155 Series ver 5.22.5D (build 2/20:42 30 01 2014)  
!  
hostname nxr155  
telnet-server enable  
http-server enable  
!  
!  
!  
!  
!  
!  
!  
ipv6 forwarding  
no fast-forwarding enable  
!  
!  
!  
interface ethernet 0  
  ip address 192.168.10.1/24  
!  
interface ethernet 1  
  no ip address  
!  
ntp  
  service enable  
!  
dns  
  service enable  
!  
syslog  
  local enable  
!  
!  
!  
system clock mobile 2  
system led ext 0 signal-level mobile 0  
!  
!  
!  
!  
!  
!  
!  
!  
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 設定例集

運用管理編

Ver 1.0.0

2014 年 4 月

発行 センチュリー・システムズ株式会社

Copyright(c) 2009-2014 Century Systems Co., Ltd. All Rights Reserved.
