

# NPI WEB

(Web印刷サービス)  
Ver. 1.0.1.0

## 設定・操作説明書

### ● 概要・環境構築手順

---

NPI WEBサービスの概要と環境構築手順について説明します。

### ● 設定ツール使用方法

---

設定ツールの使用方法について説明します。



日本プリメックス株式会社  
2016/01/26

# ご注意

- ・ 本書の内容については、万全を期して作成いたしましたが、万一ご不審な点や誤り、記載もれなど、お気づきの点がございましたらご連絡ください。
- ・ 本書の内容については、予告なしに変更することがあります。  
最新の情報はお問い合わせください。
- ・ 本書の内容の一部または全部を無断で転載、複写、複製、改ざんすることは固くお断りします。
- ・ ご利用いただいた結果の影響については、一切の責任を負いかねますのでご了承ください。
- ・ 本製品がお客様により不適切に使用されたり、本書の内容に従わずに取り扱われたり、または弊社以外の第三者により修理・変更されたことなどに起因して発生した損害などにつきましては、責任を負いかねますのでご了承ください。

# 記号について

本書では以下の記号が使われています。それぞれの記号の意味をよく理解してから製品を取り扱ってください。

 <b>注意</b>	ご使用上、必ずお守りいただきたいことを記載しています。 この表示を無視して間違った取り扱いをすると、製品の故障や動作不良の原因になる可能性があります。
 <b>参考</b>	補足説明や関連事項を記載しています。

## 使用制限

本製品を航空機・列車・船舶・自動車などの運行に直接関わる装置・防災防犯装置・各種安全装置など機能・精度等において高い信頼性・安全性が必要とされる用途に使用される場合は、これらのシステム全体の信頼性および安全を維持するためにフェールセーフ設計や冗長設計の措置を行うなど、システム全体の安全設計にご配慮いただいた上で当社製品をご使用いただきますようお願いいたします。

本製品は、航空宇宙機器、幹線通信機器、原子力制御機器、医療機器など、極めて高い信頼性・安全性が必要とされている用途への使用を意図しておりませんので、これらの用途には本製品の適合性をお客様において十分ご確認のうえ、ご判断頂きますようお願いいたします。

# もくじ

・ 概要	
一概要説明	4
・ 環境構築手順	
一環境の構築	6
一設定ツールアプリケーション使用方法	11
一ログファイルについて	18
・ XMLリファレンス	
一XML記述について	19
一XML要素の種類	21
一npiweb	22
一request	23
一printer	24
一command	25
一csv	26
一csvrange	28
一image	30
一pdf	31
一status	32
一exinfo	33
一ep	35
一EasyPrint XML要素の種類	36
一response	53
・ エラーコード	54
・ 拡張情報	55

当アプリケーションを使用することにより、日本プリメックス社製プリンターをWebブラウザ、及びTCP/IP通信にて制御することが出来ます。

主な機能として、

- ・Webブラウザからの印刷・ステータス取得
- ・TCP/IP通信による直接通信制御
- ・プリンターステータスのブロードキャスト送信
- ・ログファイルでのサービス状況確認

があります。

各設定は専用ツール(NPIWebSettingアプリケーション)で行います。

## 注意

プリンターとの通信には、プリンタードライバーを使用しています。  
ドライバーは **NPrinterLib.dll** が含まれているものを使用して下さい。(統合ドライバー推奨)  
当サービスを使用する前にドライバーをインストールし、プリンターとの通信が行えるようにしておく必要があります。詳しくはプリンターのインストールガイドを参照して下さい。

### <対応OSバージョン>

Windows7、8、10 (32/64bit)

### <動作確認ブラウザ>

FireFox (動作確認時バージョン 43.0.4)  
Chrome (動作確認時バージョン 48.0.2564.82)  
Internet Explorer11

サンプルhtmlではJava Script(XMLHttpRequest)を使用してデータの送信を行っています。

### <対応文字コード>

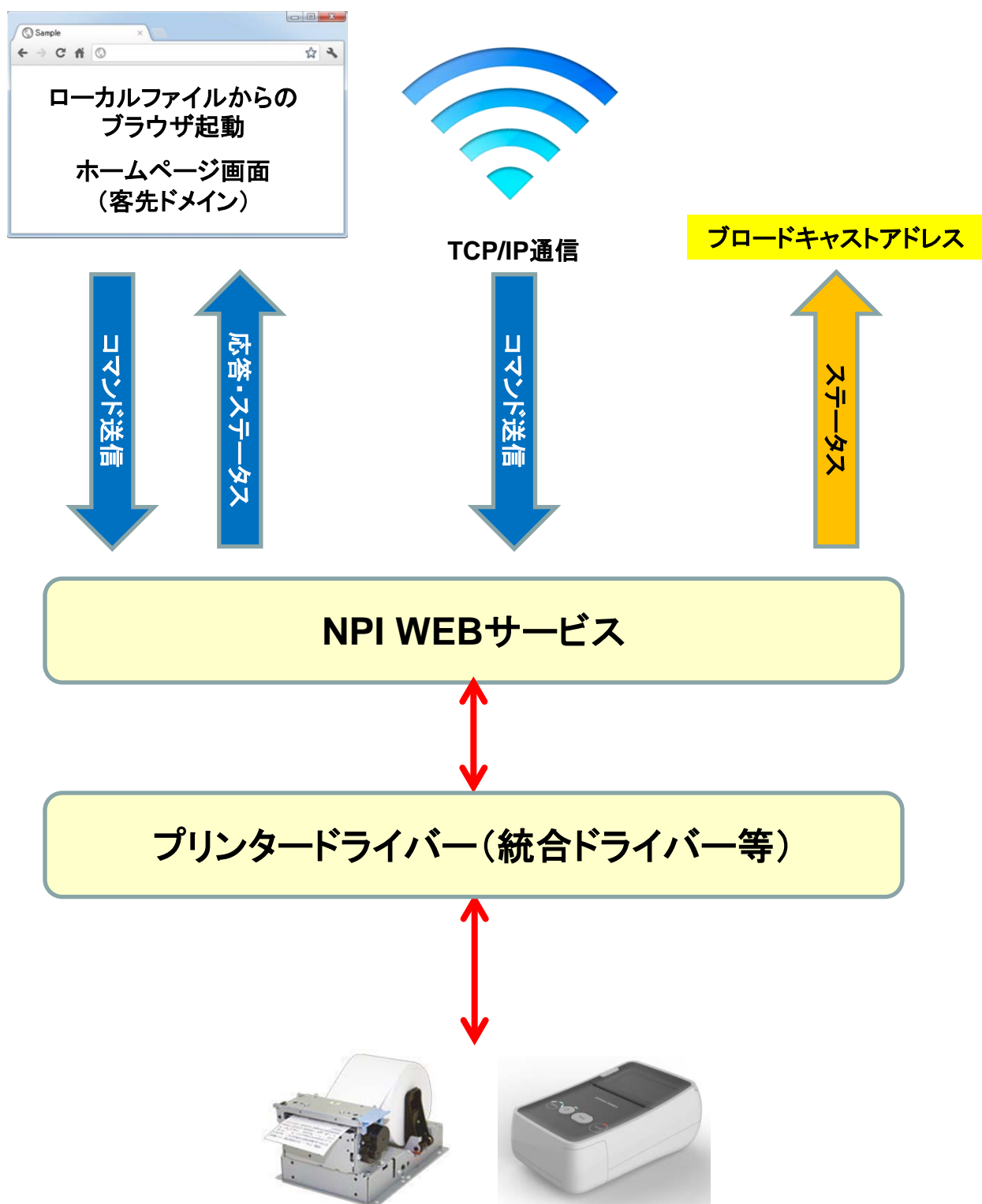
- ・UTF-8

## 注意

POST要求でデータ送信して下さい。  
また全角文字を使用する際は必ず「UTF-8」でデータを送信して下さい。

### <バージョン>

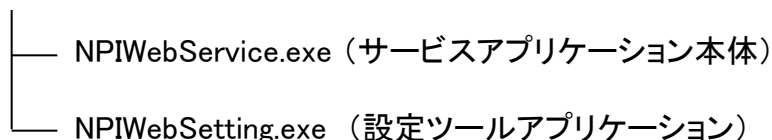
サービスプログラム : 1.0.1.0  
設定ツール : 1.0.0.0



## 環境の構築

- ・当社より提供した「NPIWeb\_Installer.zip」を解凍して下さい。  
解凍したフォルダ内は以下の構成となっております。

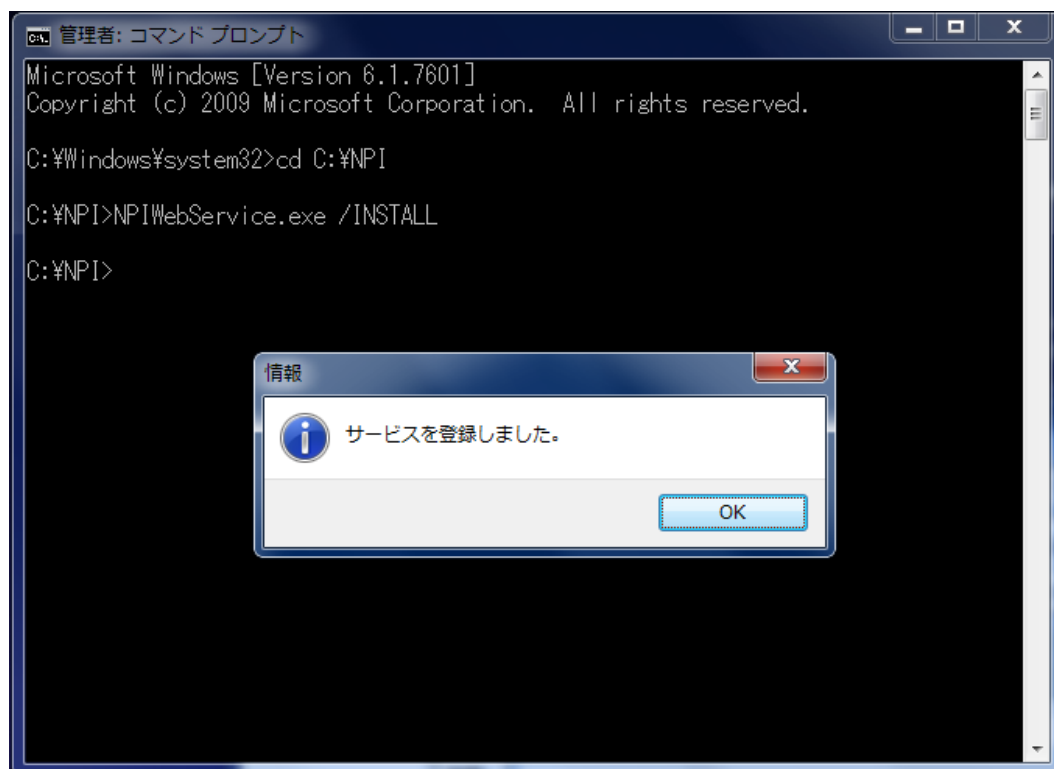
NPIWeb\_Installer



当サービスを使用するためには「サービスアプリケーション本体」をWindowsにサービス登録する必要があります。

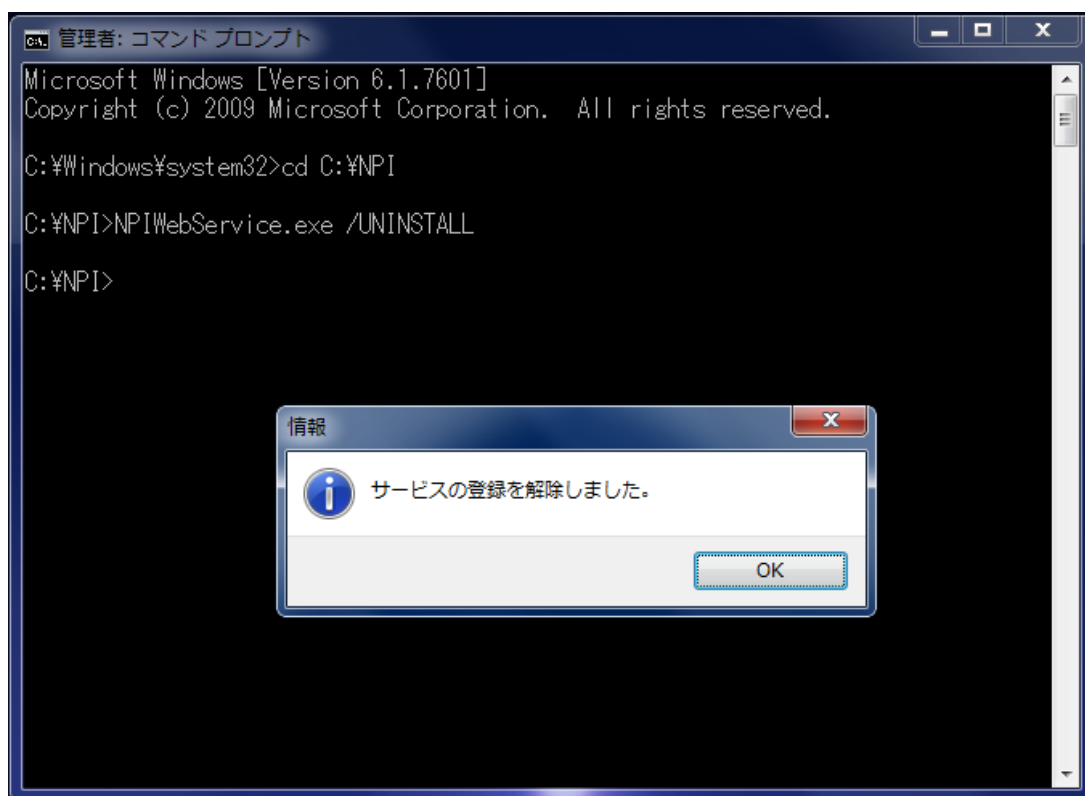
### <サービス登録方法>

1. NPIWebService.exe を適宜フォルダに配置します。  
推奨:「システムドライブ:¥NPI」フォルダ配下 (例: C:¥NPI)
2. コマンドプロンプトを「**管理者として実行**」を選択し、管理者として開きます。
3. サービスアプリケーション本体を配置したフォルダに移動し、  
「NPIWebService.exe /INSTALL」と入力します。  
「サービスを登録しました」というダイアログが出れば登録完了となります。  
(※サービス登録後もexeファイルは削除しないで下さい。アンインストール時にも使用します)



## <サービス解除方法>

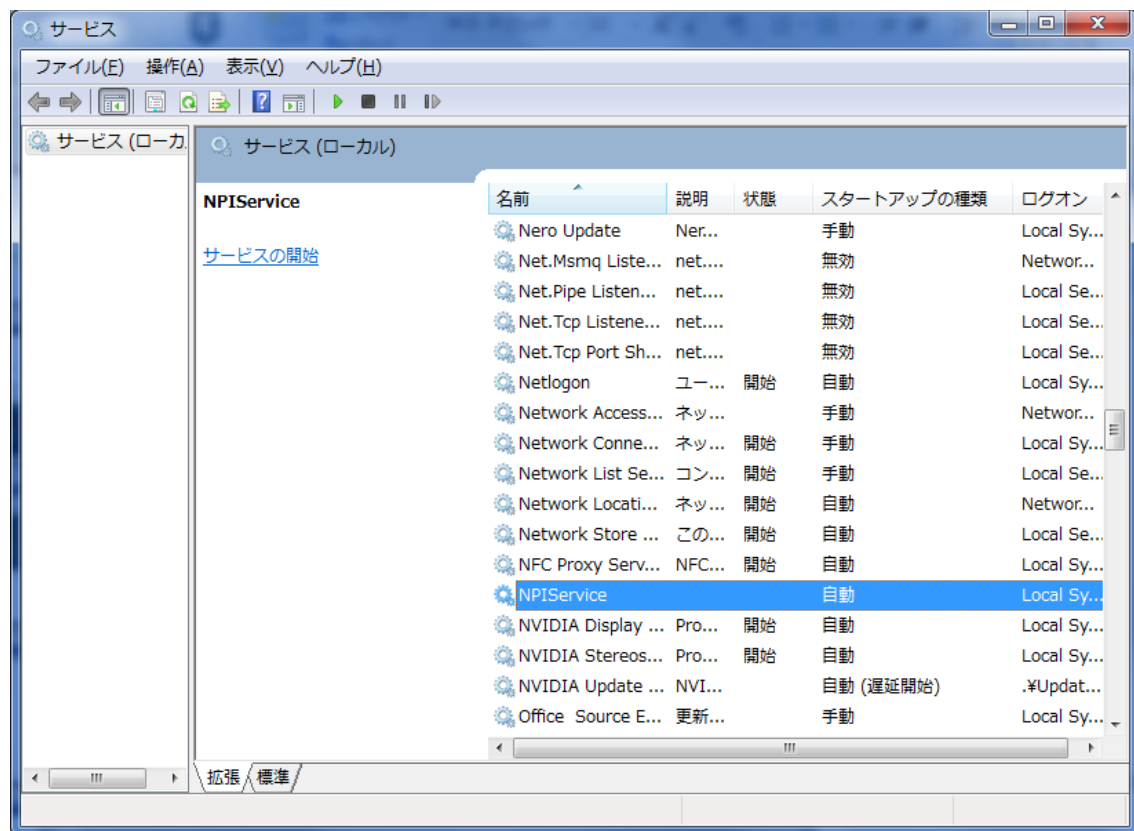
1. コマンドプロンプトを「**管理者として実行**」を選択し、管理者として開きます。
3. サービスアプリケーション本体を配置したフォルダに移動し、  
「NPIWebService.exe /UNINSTALL」と入力します。  
「サービスの登録を解除しました」というダイアログが出れば解除完了となります。





## <サービス開始/停止方法(ツールを使用しない方法)>

1. 「コントロールパネル」→「システムとセキュリティ」→「管理ツール」→「サービス」の順に開きます。
2. サービスが登録されていると以下のように「NPIService」が表示されています。



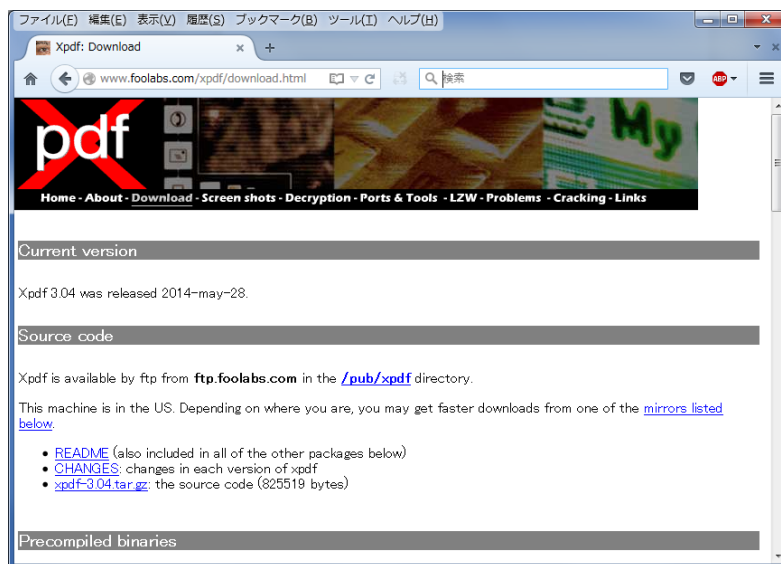
3. 「NPIService」を選択し、右クリックで「開始」「停止」を選択します。

※NPIWebSettingアプリケーションからもサービスの開始・停止を行うことができます。  
詳しくは11ページからのツール説明を参照して下さい。

## <PDF印刷について>

当サービスではPDF印刷(31ページ)の機能がありますが、使用するためにはXpdfを用意する必要があります。以下に例を示します。

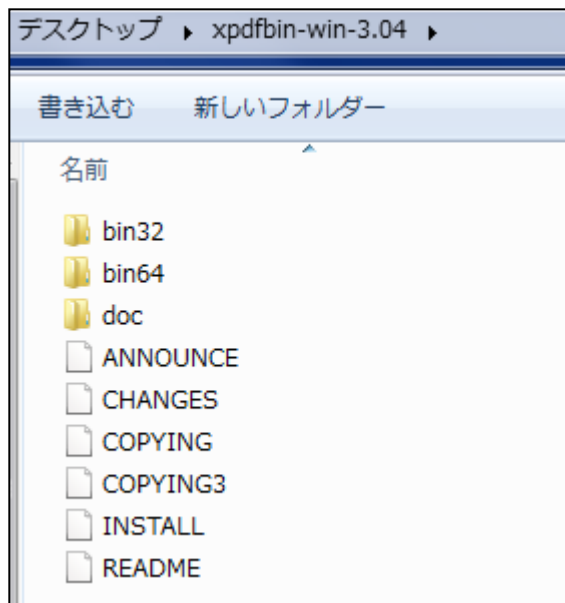
(2015年9月現在の方法となります。あくまで参考として下さい)



1. Xpdfのサイトにアクセスし、ダウンロードします。

[xpdfbin-win-3.04.zip](#)

2. 解凍すると以下に示すフォルダ構成となります。  
配置する場所はどこでも構いません。



3. PDF印刷にはbin32、またはbin64に存在する **pdftopng.exe** を使用します。  
詳しくはXML説明のPDF印刷項目を参照して下さい。

## ＜PDF印刷について(印字が文字化けする場合)＞

Xpdfを使用した場合に日本語等の全角文字が化けて印刷されることがあります。  
その場合は日本語パッケージを入れることで解決します。

## ＜EasyPrint機能を使用する場合＞

当サービスでは弊社提供のNII EasyPrintの機能を使用することもできます。  
リリースされているNII EasyPrintのdll群を、アプリケーションを使用するPCに  
配置して下さい。(サービスプログラム本体と同じフォルダ、または環境変数のPATHが  
通っているフォルダに配置して下さい。)

必要なdll

- ・NiiGF.dll
- ・NiiRasB.dll
- ・NiiRasL.dll
- ・NiiBit.dll
- ・NiiPM.dll
- ・NPEDIT.dll

### 注意

先頭に「Nii」と付いているdllファイルは32/64bitのファイルがそれぞれ存在します。  
使用しているOSが32bitの場合は32bitのdllを、64bitの場合は64bitのdllを  
配置して下さい。

またEasyPrintで生成したPFDファイル(イメージが登録されています)をPCの任意の  
フォルダに配置して下さい。後述のXMLのNReadInfo要素で読み込み、データ展開を  
行います。

### 注意

別PCでPFDファイルを作成した場合には、使用するドライバー名を同一名称にして下さい。

# 設定ツールアプリケーション使用方法

サービス開始前に予め設定ツールアプリケーションにて、Web通信や直接通信に関する事前設定を行う必要があります。

設定を行うと、システムドライブ（OSがインストールされているドライブ）直下のNPIフォルダに **NWebSettingInf** というファイルが生成されます。このファイル内に設定情報が書き込まれます。

（ファイル設定例）

[WEB]

ENABLE=1

PORT=50011

CROSSDOMAIN=0

DOMAIN=http://www.primex.co.jp

[DIRECT]

ENABLE=1

PORT=50012

DRIVER=NPI NP Series Integration Driver

[BCSTATUS]

ENABLE=1

ADDRESS=192.168.92.255

PORT=50013

INTERVAL=3

DRIVER=NPI NP Series Integration Driver

[LOG]

SAVEDAYS=2

TYPE=1,2,3

当ファイルは設定ツールで編集して下さい。  
（手動でも変更できますが、動作の保証は行いません）

# 設定ツールアプリケーション使用方法

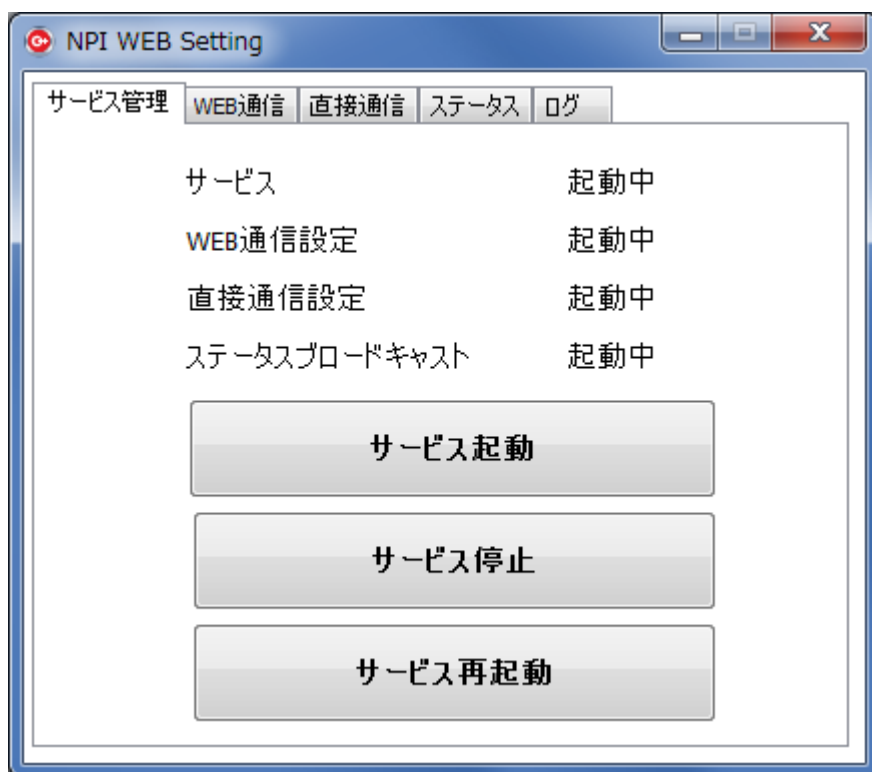
設定ツールを起動するには、当社より提供されているNPIWebSetting.exeを右クリックし、「**管理者として実行**」を選択します。



上記画面が表示されますので、言語の選択をします。  
日本語で使いたい場合は「Japanese」、英語で使いたい場合は「English」を  
押下して下さい。

# 設定ツールアプリケーション使用方法

## <サービス管理画面>



サービス管理タブでは各サービスの稼働状況の確認と、サービスの起動・停止が行えます。

※状況確認(停止中、または起動中で表示されます)

サービス	: サービスが起動しているかどうか
WEB通信設定	: ブラウザからのXMLによる通信
直接通信設定	: TCP/IP接続による直接通信
ステータスブロードキャスト	: プリンタステータスのブロードキャスト送信

※サービス起動・停止(ボタンを押下することで実行されます)

サービス起動 : サービスを開始します。既に開始されている場合はエラーダイアログが表示されます。

サービス停止 : サービスを停止します。既に停止されている場合はエラーダイアログが表示されます。

サービス再起動 : 「サービス停止」→「サービス起動」の順に処理を行います。

# 設定ツールアプリケーション使用方法

## ＜WEB通信設定画面＞



WEB通信設定のサービスを稼動することで、WebブラウザからXML送信にてプリンターへの印字や状態取得を行うことができます。

WEB通信タブでは以下の設定を行います。

WEB通信設定：「無効」「有効」を選択します。  
無効を選択すると、WEB通信は一切行えません。

待受ポート番号：1 ～ 65535の範囲でポート番号を指定します。  
ブラウザから送信する際にはここで指定したポート番号を使用します。

クロスドメイン設定：

- 指定無 → 要求を全て受け付けます。
- ドメイン指定 → 指定したドメインからの要求のみ受け付けます。  
ドメインは「許可ドメイン」が入力します。  
http://もしくはhttps://を先頭に含めた形で指定して下さい。
- ローカルファイル → ローカルに配置したhtmlファイル等からの要求のみ受け付けます。

許可ドメイン → クロスドメイン設定が「ドメイン指定」の場合のみ使用します。  
指定したドメインからの要求のみ受け付けます。

各設定を選択・入力したら、最後に「設定保存」ボタンを押下して下さい。  
設定を反映させるにはサービスの再起動を行って下さい。

# 設定ツールアプリケーション使用方法

## <直接通信設定画面>



直接通信設定のサービスを稼動することで、TCP/IP接続で直接データを送信することが出来ます。

直接通信タブでは以下の設定を行います。

直接通信設定：「無効」「有効」を選択します。  
無効を選択すると、直接通信は一切行えません。

待受ポート番号：1 ～ 65535の範囲でポート番号を指定します。  
直接通信で接続する際にはここで指定したポート番号を使用します。

使用ドライバー：PCにインストールされているドライバーが選択できます。  
印刷に使用するプリンタードライバーを指定して下さい。

各設定を選択・入力したら、最後に「設定保存」ボタンを押下して下さい。  
設定を反映させるにはサービスの再起動を行って下さい。

サービスを起動させると、指定したポート番号で待受けを行います。  
接続は1クライアントのみとなり、複数接続は出来ません。  
お客様のクライアントプログラムより接続した後は、プリンターコマンド(バイナリデータ)を送信することで、指定したドライバー経由でプリンターにデータを送信します。  
データはテキストデータで指定せず、必ずバイナリデータで指定して下さい。



# 設定ツールアプリケーション使用方法

## ＜ステータスブロードキャスト設定画面＞



ステータス送信サービスを開始することで、指定したブロードキャストアドレスへ、プリンタステータスをUDP送信することが出来ます。

プリンタステータスは1byte (0x00～0xFF)となります。  
ステータス値の詳細は各プリンタの仕様書をご参照下さい。

直接通信タブでは以下の設定を行います。

ステータスブロードキャスト：「無効」「有効」を選択します。  
無効を選択すると、ステータス送信は行いません。

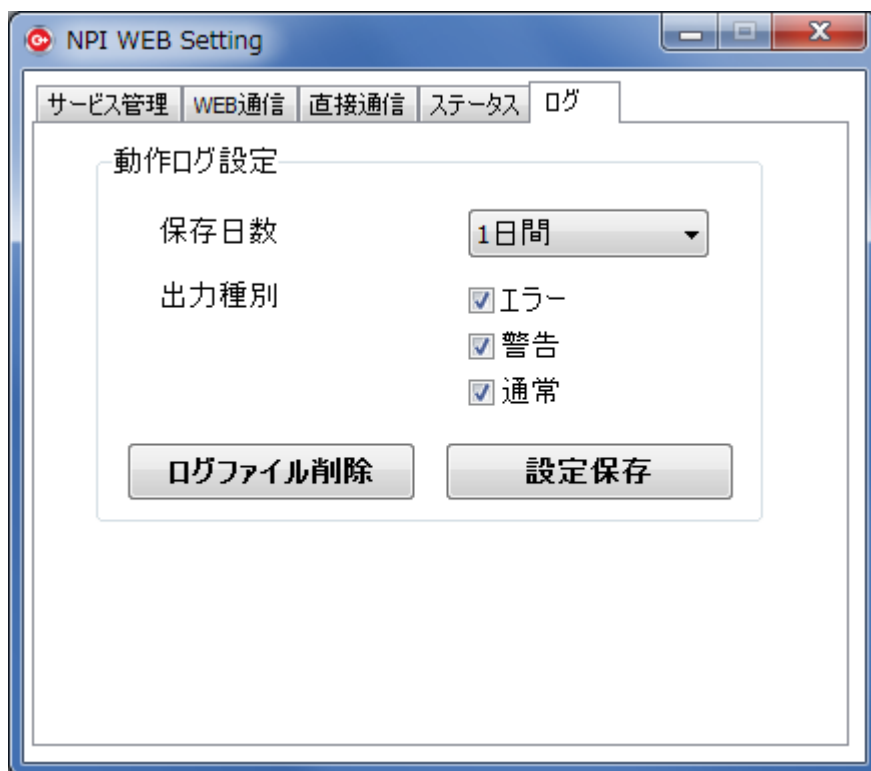
ブロードキャストアドレス：指定したアドレスへUDPでステータスを送信します。  
ポート番号：指定したポートへ送信します。  
送信間隔(秒)：指定した秒数おきにステータスを送信します。  
(1～600秒の範囲で設定できます)

使用ドライバー：PCにインストールされているドライバーが選択できます。  
印刷に使用するプリンタードライバーを指定して下さい。

各設定を選択・入力しましたら、最後に「設定保存」ボタンを押下して下さい。  
設定を反映させるにはサービスの再起動を行って下さい。

# 設定ツールアプリケーション使用方法

## <ログ設定画面>



ログの出力設定、削除を行います。  
ファイルの詳細については次ページで参照して下さい。

保存日数：「保存しない」を選択するとログ出力を停止します。  
「1日間」では当日のログのみ保存します。(翌日には削除)  
「2～7日間」はそれぞれ指定した日数分のログファイルを保存します。

出力種別：エラー、警告、標準の3種類から選択します。  
エラー(ERR)は印字やプリンター接続、アプリ挙動に問題があった場合、  
警告(WRN)はプリンター未オープン状態での処理、値が不正であった場合等、  
標準(FNC)は呼出された機能の表示などをログに出力します。

「ログファイル削除」ボタンを押すと保存されているログを全て削除します。

各設定を選択・入力したら、最後に「設定保存」ボタンを押下して下さい。  
設定を反映させるにはサービスの再起動を行って下さい。

## ログファイルについて

ログは システムドライブ:\NPI\logフォルダ に保存されます。  
ファイル名は「WebLog\_YYYYMMDD」となります。  
(YYYYは西暦、MMは月2桁、DDは日2桁となります)

ログは1行ごとに 年月日・日時 (yyyy/MM/dd HH:mm:ss.SSS) 、ログ種別、  
ログ内容の順に出力します。

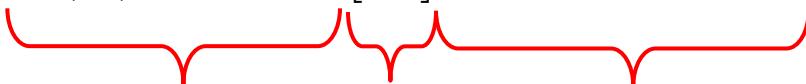
ログ種別はエラー(ERR)、警告(WRN)、通常(FNC)で表記しています。

(ログ出力例)

2015/01/21 14:01:36.456 [FNC] .....

2015/01/21 14:01:36.644 [FNC] .....

2015/01/21 15:36:20.444 [ERR] .....



年月日・日時

ログ種別

ログ内容

## XML記述について

以下の形式でXMLのデータ送受信を行います。  
サンプルのhtmlファイルでは、XMLHttpRequest (FireFox、Chrome)、  
ActiveXObject (IE)を使用してデータを送信しています。

※NPI WEBアプリでのXMLデータ送受信では「npiweb」要素で全体を括ります。  
WebブラウザからNPI WEBアプリに送信する場合は「request」を使用して下さい。  
request要素の後に各要素(printer、commandなど)を記述します。

```
<npiweb>
  <request>
    ...
    ...
  </request>
</npiweb>
```

XMLは先に記述してある要素から順に処理を行います。  
なお、requestはデータ検出を開始してから10秒でタイムアウトします。  
10秒以内にデータを送信するようにして下さい。

(例) プリンタードライバーを設定、Primexと印字し、1行改行する場合のリクエスト

```
<npiweb>
  <request>
    <printer>
      <driver>NPI NP Series Integration Driver</driver>
    </printer>
    <command>
      <data>"Primex"0A</data>
    </command>
  </request>
</npiweb>
```

※NPI WEBアプリからの応答は「response」で返されます。

例)printer要素の結果

```
<npiweb>
  <response>
    <printer>
      <result>●</result>
    </printer>
  </response>
</npiweb>
```

● にXMLの送信結果(エラーコード)が格納されます。  
エラーコードの詳細は～ページを参照して下さい。

XMLが不正などの理由で各要素が処理されない場合はresultだけを応答します。

```
<npiweb>
  <response>
    <result>●</result>
  </response>
</npiweb>
```

またep要素(EasyPrint)を使用した場合は以下の形で応答が返されます。

下記はNReadInfo、NImagePrintを使用した場合の応答例です。

```
<npiweb>
  <response>
    <ep>
      <NReadInfo>
        <result>▲</result>
      </NReadInfo>

      <NImagePrint>
        <result>▲</result>
        <jobID>2</jobID>
      </NImagePrint>
    </ep>
  </response>
</npiweb>
```

▲にはGF関数説明書に記されているエラー値が格納されます。  
詳しくはGF関数説明書を参照して下さい。

## NPI WEB XML要素の種類

以下の要素が用意されています。

EasyPrint機能(NiiGF.dll)を呼出するためのXMLは別途17ページを参照して下さい。

用 途	要素名	説 明
NPI WEB 先頭要素	npweb	NPI WEBアプリケーションでのXML送受信時に使用します。
XML送信要素	request	ブラウザからNPI WEBへの送信時に使用します。
ドライバー設定	printer	通信に使用するプリンタードライバーを設定します。
コマンド実行	command	プリンターへコマンドを送信します。
csvファイル読込	csv	csvファイルを読み込み、データ送信します。
csvファイル範囲指定	csvrange	csvファイルを範囲指定で読み込み、データ送信します。
画像印字実行	image	画像ファイルの印字を行います。
PDFファイル印字	pdf	PDFファイルの印字を行います。
ステータス要求	status	プリンターステータスの取得を行います。
拡張情報要求	exinfo	プリンターの拡張情報の取得を行います。
EasyPrint機能要素	ep	EasyPrint機能(NiiGF.dll)を使用する際に記述します。
XML応答要素	response	NPI WEBからブラウザへの応答時に使用します。
実行結果	result	XML送信結果を応答します。

要素名	npiweb	
request		
子要素	型	説明
request	-	ブラウザからNPI WEBサービスへのXML送信要求
response		
子要素	型	説明
response	-	NPI WEBサービスからブラウザへのXML応答結果
処理内容	<div><div><div></div><div>注意</div></div><div><ul style="list-style-type: none"><li>・ブラウザからNPI WEBサービスへのXML送信時にはnpiweb要素が先頭に必要となります。 子要素にrequest要素を記述して下さい。 requestは複数指定できません。</li><li>・NPI WEBサービスからブラウザへのXML応答時にはnpiweb要素が先頭に記述されます。 子要素にはresponse要素が含まれています。</li></ul><div><div></div><div>1回の送信につきnpiweb要素は1つのみ有効となります。 複数指定しないで下さい。 (例)この場合は処理1のみ有効となります。 &lt;npiweb&gt;     ~ 処理1 ~ &lt;/npiweb&gt; &lt;npiweb&gt;     ~ 処理2 ~ &lt;/npiweb&gt;</div></div></div></div>	

要素名	request	
request		
子要素	型	説明
printer	－	プリンタードライバー設定
command	－	プリンターコマンド送信
csv	－	csvファイル読込・データ送信
csvrange	－	csvファイル範囲指定読込・データ送信
image	－	画像ファイル印字
pdf	－	PDFファイル印字
status	－	プリンターステータス取得
exinfo	－	プリンター拡張情報取得
ep	－	NII EasyPrint機能の実行
response		
子要素	型	説明
result ※	数値	XML送信結果(子要素の処理が行えなかった場合) 送信時の子要素
処理内容	<p>・WEBブラウザからNPI WEBサービスへのデータ送信時に必要となる要素です。子要素には以下の要素を記述します。</p> <p>・printer ・command ・csv ・csvrange ・image ・pdf ・status ・exinfo ・ep</p> <p>詳しくは各要素の説明ページを参照して下さい。</p>	



要素名	printer	
request		
子要素	型	説明
driver	文字列	プリンタードライバー名
response		
子要素	型	説明
result	数値	処理結果
処理内容	<div>・通信に必要なプリンタードライバー名を指定します。 request要素内の先頭に記述して下さい。 driverは重複して指定することはできません。</div> <div>(例)ドライバー名「NPI NP Series Integration Driver」を指定する場合</div> <div>&lt;npiweb&gt;   &lt;request&gt;     &lt;printer&gt;       &lt;driver&gt;NPI NP Series Integration Driver&lt;/driver&gt;     &lt;/printer&gt;     ...     ...   &lt;/request&gt; &lt;/npiweb&gt;</div>	

要素名	command	
request		
子要素	型	説明
data	文字列	プリンターコマンド
response		
子要素	型	説明
result	数値	処理結果
処理内容	<div><ul style="list-style-type: none"><li>指定したコマンド文字列をプリンターへ送信します。 dataは重複して指定することはできません。 コマンドは連結して記述して下さい。</li></ul><p>(例) プリンタードライバーに「NPI NP Series Integration Driver」を指定し、 「Primex」と印字し、5行改行する。</p><pre>&lt;npiweb&gt;   &lt;request&gt;     &lt;printer&gt;       &lt;driver&gt;NPI NP Series Integration Driver&lt;/driver&gt;     &lt;/printer&gt;     &lt;command&gt;       &lt;data&gt;"ABC"0A0A0A0A0A&lt;/data&gt;     &lt;/command&gt;   &lt;/request&gt; &lt;/npiweb&gt;</pre></div>	

要素名	csv	
request		
子要素	型	説明
path	文字列	csvファイルパス(絶対パス)
x	数値	csvデータ横方向位置
y	数値	csvデータ縦方向位置
width	数値	データ桁合わせ(省略可)
position	※下記参照	データ位置揃え(省略可)
response		
子要素	型	説明
result	数値	処理結果
処理内容	<p>・pathで指定したcsvファイルを読み込み、xとyで指定したデータを文字列として 印字します。(プリンターコマンドを指定することは出来ません) 下記、子要素は重複して指定することはできません。</p> <p>path        : csvファイルを絶対パスで指定します。           pathで読み込んだデータは一度読み込むと、サービスを           停止するまでデータが保持されます。</p> <p>x           : csvデータの横方向位置を指定します(0が始点) y           : csvデータの縦方向位置を指定します(0が始点)           x、yは「*」を指定することでデータ末尾を指定します。           「*-n」(nは数値)と指定することで末尾からn番目戻ったデータと           いう使用方法も可能です。</p> <p>width       : 指定された数値で半角スペースの桁合わせを行います。           省略された場合は桁合わせは行いません。</p> <p>position   : データの位置揃えを行います。widthと組み合わせて使用する必要が           あります。           L と指定した場合は左揃え、C と指定で中央揃え、R と指定で右揃えと           なります。省略された場合は左寄せとなります。</p> <p>※csvデータはカンマ区切りで記述して下さい。 (カンマ自体をデータに含めることはできません)</p> <p>※改行は自動付与されません。</p>	

(例) csv要素を使用してファイル読込、「2-1」を10桁合わせの右揃えで印字  
続けて「2-3」を5桁合わせの中央揃えで印字  
(「2-1」と「2-3」の間には改行は含まれません)

ファイル(Cドライブ直下にtest.csv)

1-1, 1-2, 1-3  
2-1, 2-2, 2-3  
3-1, 3-2, 3-3  
4-1, 4-2, 4-3

## XML

```
<csv>  
  <path>C:\test.csv</path>  
  <x>0</x>  
  <y>1</y>  
  <width>10</width>  
  <position>R</position>  
</csv>  
<csv>  
  <x>*</x>  
  <y>*-2</y>  
  <width>5</width>  
  <position>C</position>  
</csv>
```

要素名	csvrange	
request		
子要素	型	説明
path	文字列	csvファイルパス(絶対パス)
cols	※下記参照	csvデータ横方向位置 (カンマ区切り)
colwidth	※下記参照	データ桁合わせ(省略可)
colposi	※下記参照	データ位置揃え(省略可)
rowstart	数値	csvデータ縦方向始点
rowend	数値	csvデータ縦方向終点
response		
子要素	型	説明
result	数値	処理結果
処理内容	<p>・pathで指定したcsvファイルを読み込み、cols(横方向)とrowstart、rowend(縦方向)で範囲指定したデータを文字列として印字します。 (プリンターコマンドを指定することは出来ません) 下記、子要素は重複して指定することはできません。</p> <p>path : csvファイルを絶対パスで指定します。</p> <p>cols : csvデータの横方向位置をカンマ区切りで列挙します (0が始点)</p> <p>colwidth : 横方向の桁合わせ値を数値(カンマ区切り)で列挙します。 colsと同じ数の指定が必要です。(省略可能)</p> <p>colposi : 横方向の位置揃え指定をカンマ区切りで列挙します。 colsと同じ数の指定が必要です。(省略可能) colwidthと組み合わせて使用する必要があります。 L と指定した場合は左揃え、C と指定で中央揃え、 R と指定で右揃えとなります。 省略された場合は左寄せとなります。</p> <p>rowstart : csvデータの縦方向開始位置をカンマ区切りで指定します (0が始点)</p> <p>rowend : csvデータの縦方向終端位置をカンマ区切りで指定します (0が始点)</p> <p>※cols、rowstart、rowendは「*」での末尾指定が可能です。</p> <p>※csvデータはカンマ区切りで記述して下さい。 (カンマ自体をデータに含めることはできません)</p>	

ファイル(Cドライブ直下にtest.csv)

1-1, 1-2, 1-3  
2-1, 2-2, 2-3  
3-1, 3-2, 3-3  
4-1, 4-2, 4-3

<csvrange>

<path>C:\test.csv</path>  
<cols>0, 1, 3</cols>  
<colwidth>5, 5, 10</colwidth>  
<colposi>R, C, R</colposi>  
<rowstart>0</rowstart>  
<rowend>2</rowend>

x 方向は0,1,3の順、y方向は0~2番目まで  
データの桁合わせと位置揃えを行う

</csvrange>

<csvrange>

<cols>0, \*, \*-1</cols>  
<rowstart>\*</rowstart>  
<rowend>0</rowend>

x 方向は0, \*(末尾なので3) ,  
\* - 1 (末尾から1つ戻るので2) の順、  
y方向は\* (末尾) ~0 (先頭) まで。  
末尾から先頭へ戻る形で表示。  
データの桁合わせと位置揃えは行わない。

</csvrange>

上記XMLは以下のように印字されます。

1-1	1-2	1-4
2-1	2-2	2-4
3-1	3-2	3-4
4-14-44-3		
3-13-43-3		
2-12-42-3		
1-11-41-3		

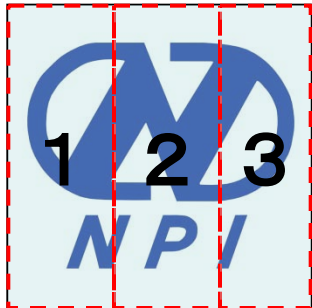
要素名	image	
request		
子要素	型	説明
path	文字列	画像ファイルパス(絶対パス)
type	数値	画像印字種別
width	数値	画像印字幅 (省略可)
response		
子要素	型	説明
result	数値	処理結果
処理内容		


・画像ファイルの印字を行います。  
ファイルはbmpファイルのみ指定できます。  
下記、子要素は重複して指定することはできません。

path : 画像ファイル(bmpファイルに限る)を絶対パスで指定します。

type : 印字種別を以下の4種類で指定します。  
0 : ラスターライン  
1 : ラスターブロック  
2 : ラスター階調表現  
3 : ビットイメージ

width : 指定した数値 (ピクセル幅) で画像を縦に分割して印字します。(省略可能)





画像を分割し、左から順に印字します。  
省略された場合は分割せず、  
左から印字有効幅分を印字します。  
※印字有効幅=2040dot 203dpi=255mm  
300dpi=170mm

要素名	pdf	
request		
子要素	型	説明
path	文字列	PDFファイルパス(絶対パス)
type	数値	印字種別
width	数値	印字幅 (省略可)
exe	文字列	画像変換exeパス(絶対パス)
page	※下記参照	PDF印刷ページ(省略可)
response		
子要素	型	説明
result	数値	処理結果
処理内容		
<p>・PDFファイルの印字を行います。PDFを一度画像に変換し、印刷を行うためXpdfが必要となります。(9ページ参照)</p> <p>下記、子要素は重複して指定することはできません。</p> <p>path : PDFファイルを絶対パスで指定します。</p> <p>type : 印字種別を以下の4種類で指定します。</p> <p>0 : ラスターライン</p> <p>1 : ラスターブロック</p> <p>2 : ラスター階調表現</p> <p>3 : ビットイメージ</p> <p>width : 指定した数値 (ピクセル幅) でPDFを縦に分割して印字します。(省略可能)</p> <p>※分割方法は「image」要素を使用した場合と同様となります。</p> <p>exe : Xpdfの「pdftopng.exe」を絶対パスで指定します。</p> <p>page : PDFファイルの印刷したいページをカンマ区切りで列挙します (省略可能)</p>		



要素名	status	
request		
子要素	型	説明
－	－	－
response		
子要素	型	説明
result value	数値 数値	処理結果 プリンターステータス値
処理内容		

・プリンタステータスを取得します。  
子要素はありません。  
request送信時は以下のように指定して下さい。  
<status></status>

response応答は以下の形で返されます。  
<npiweb>  
  <response>  
    <status>  
      <result>●</result>  
      <value>▲</value>  
    </status>  
  </response>  
</npiweb>

●はXMLの送信結果(エラーコード)となります。  
▲にプリンタステータスが0～255の値で取得できます。

注意

プリンター印字中に「status」要素の実行を行うと、処理がタイムアウトする可能性があります。印字開始/終了コマンドを使用するなどしてプリンターの印字がされていないことを確認してから、処理を行って下さい。

要素名	exinfo	
request		
子要素	型	説明
number	数値	拡張情報番号(1～31)
response		
子要素	型	説明
result	数値	処理結果
number	数値	送信時に使用した拡張情報番号(1～31)
value	文字列または、 16進数値文字列	※下記説明を参照
処理内容	<p>・プリンターの拡張情報を取得します。</p> <p>number要素には1～31の数値を指定します。numberは重複して指定できません。</p> <p>複数の拡張情報を取得したい場合は、exinfoごと複数指定して下さい。</p> <p>request送信時は以下のように指定して下さい。</p> <p>&lt;exinfo&gt;</p> <p>    &lt;number&gt;▲&lt;/number&gt;</p> <p>&lt;/exinfo&gt;</p> <p>response応答は以下の形で返されます。</p> <p>&lt;npiweb&gt;</p> <p>    &lt;response&gt;</p> <p>        &lt;exinfo&gt;</p> <p>            &lt;result&gt;●&lt;/result&gt;</p> <p>            &lt;number&gt;▲&lt;/number&gt;</p> <p>            &lt;value&gt;■&lt;/value&gt;</p> <p>        &lt;/exinfo&gt;</p> <p>    &lt;/response&gt;</p> <p>&lt;/npiweb&gt;</p> <p>numberの▲には送信時に使用した拡張情報番号が格納されます。</p> <p>valueの■には拡張番号が2,3,4の場合にはASCII文字列、それ以外の場合には16進数文字列が格納されます。(次ページに例を示します)</p>	

例) 拡張情報2を指定し、モデル名を取得する

送信XML

```
<npiweb>
  <request>
    <printer>
      <driver>NPI NP Series Integration Driver</driver>
    </printer>
    <exinfo>
      <number>2</number> ← 拡張情報2を指定
    </exinfo>
  </request>
</npiweb>
```

応答XML

```
<npiweb>
  <response>
    <exinfo>
      <result>0</result>
      <number>2</number>
      <value>NP-T320</value> ← 拡張情報2は文字列として応答
    </exinfo>
  </response>
</npiweb>
```

例) 拡張情報1を指定し、拡張ステータス(4バイト)を取得する

応答XML

```
<npiweb>
  <response>
    <exinfo>
      <result>0</result>
      <number>1</number>
      <value>0000000D</value> ← 拡張情報1は16進数文字列として応答
    </exinfo>
  </response>
</npiweb>
```

例では  
0x00、0x00、0x00、0x0D の4バイト  
となる。

要素名	ep	
request		
子要素	型	説明
下記参照	－	EasyPrint XML要素
response		
子要素	型	説明
下記参照	－	EasyPrint XML要素
処理内容	<div>・ep要素の子要素として、次ページから説明するEasyPrint XML要素を記述します。 送信時には処理したい順に子要素を記述して下さい。 また応答時には同じくep要素の子要素としてEasyPrint XML要素が返されます。 この時返されるEasyPrint XML要素は送信時に指定した要素の結果となります。</div>	

## EasyPrint XML要素の種類

以下の要素が用意されています。

ep要素の子要素として記述します。

用 途	要素名	説 明
PFDファイル読込	NReadInfo	EasyPrintで生成したPFDファイルを読み込み、データを展開します。
XMLファイル読込	NSetXml	指定されたXMLファイルを読み込み、データを展開します。
フォーマットイメージ出力	NImagePrint	指定されたフォーマットイメージをプリンタへ出力します。
コマンドデータ出力	NPrint	指定されたコマンドデータをプリンタへ出力します。
プリンタ初期化	NPrinterInit	指定されたプリンタIDに登録されている初期化コマンドをプリンタへ出力します。
ステータス取得	NGetStatus	プリンタのステータスを取得します。
拡張情報取得	NGetInfo	指定された拡張種別からプリンタの拡張情報を取得します。
Textデータ設定	NSetText	Textコンポーネントのデータを設定します。
Textデータ取得	NGetText	Textコンポーネントのデータを取得します。
Barcodeデータ設定	NSetBarcode	Barcodeコンポーネントのデータを設定します。 (16進数文字列)
Barcodeデータ取得	NGetBarcode	Barcodeコンポーネントのデータを取得します。 (16進数文字列)
Barcodeデータ設定	NSetBarcode2	Barcodeコンポーネントのデータを設定します。 (ASCII文字列)
Barcodeデータ取得	NGetBarcode2	Barcodeコンポーネントのデータを取得します。 (ASCII文字列)
Imageデータ設定	NSetImage2	Imageコンポーネントのデータを設定します。
コンポーネント Visible設定	NSetVisible	コンポーネントの可視・不可視状態を設定します。
コンポーネント Visible情報取得	NGetVisible	コンポーネントの可視・不可視状態を取得します。

要素名	NReadInfo	
request		
子要素	型	説明
file	文字列	EasyPrintで生成したPFDファイル（パス含む）
response		
子要素	型	説明
result	数値	エラーコード（※53ページを参照）
処理内容		

・Windows EasyPrintで生成したPFDファイルを読み込み、データ展開します。  
フォーマットイメージ印字、コマンドデータ印字、ステータス取得等全ての処理に  
おいて、当関数でデータ展開を行っておく必要があります。

fileで指定するパスはNPI WEBアプリケーションを実行しているPC内のファイルを  
指定して下さい。

EasyPrint GF関数の「**Ni iReadInfoEp**」を呼出して処理を行います。  
XMLとGF関数の対応は以下の通りとなります。

request	
NPIWEB XML要素	NiiGF.dll
NReadInfo file	NiiReadInfoEp i_file
response	
NiiGF.dll	NPIWEB XML要素
NiiReadInfoEp 関数戻り値	NReadInfo result

要素名	NSetXml	
request		
子要素	型	説明
file	文字列	XMLファイル(パス含む)
response		
子要素	型	説明
result	数値	エラーコード(※53ページを参照)
処理内容		

・指定されたXMLファイルを読み込み、データ展開します。  
XMLファイルの記述方法は、GF関数説明書32ページからの  
「関数説明(XMLファイル)」を参照して下さい。

EasyPrint GF関数の「**NiiSetXMLEp**」を呼出して処理を行います。  
XMLとGF関数の対応は以下の通りとなります。

request	
NPIWEB XML要素	NiiGF.dll
NSetXml file	NiiSetXMLEp _xml
response	
NiiGF.dll	NPIWEB XML要素
NiiSetXMLEp 関数戻り値	NSetXml result

要素名	NImagePrint	
request		
子要素	型	説明
pdfprtID	文字列	プリンタID
formatID	文字列	フォーマットID
debug	数値	コマンド文字列ファイル出力(1:出力あり、0:出力なし)
response		
子要素	型	説明
result	数値	エラーコード(※53ページを参照)
pdfprtID	文字列	プリンタID
formatID	文字列	フォーマットID
jobID	数値	プリントジョブID
処理内容		

- ・指定されたフォーマットIDにて作成されたイメージを、指定されたプリンタIDに出力します。
- ・debugは省略が可能です。省略された場合は0:出力なしに設定されます。

EasyPrint GF関数の「**NiiImagePrintEp**」を呼出して処理を行います。  
XMLとGF関数の対応は以下の通りとなります。

request	
NPIWEB XML要素	NiiGF.dll
NImagePrint	NiiImagePrintEp
pdfprtID	i_pid
formatID	i_fid
debug	i_debug
response	
NiiGF.dll	NPIWEB XML要素
NiiImagePrintEp	NImagePrint
関数戻り値	result
i_pid	pdfprtID
i_fid	formatID
o_jobid	jobID



要素名	NPrint																											
request																												
子要素	型	説明																										
pdfprtID	文字列	プリンタID																										
debug	数値	コマンド文字列ファイル出力(1:出力あり、0:出力なし)																										
data	16進数文字列	送信データ(16進数文字列)																										
size	数値	出力バイト数																										
response																												
子要素	型	説明																										
result	数値	エラーコード(※53ページを参照)																										
pdfprtID	文字列	プリンタID																										
jobID	数値	プリントジョブID																										
処理内容	<div>・指定された送信データを指定されたプリンタIDに出力します。 引数dataは16進コードを文字列として指定して下さい。0x等の表示は必要ありません。 詳しくはGF関数説明書9ページを参照して下さい。 例) ABCと出力し、改行する場合 data ⇒ “ABC”0A size ⇒ 7</div> <div>・debugは省略が可能です。省略された場合は0:出力なしに設定されます。 EasyPrint GF関数の「NiiPrintEp」を呼出して処理を行います。 XMLとGF関数の対応は以下の通りとなります。</div> <table><tr><td colspan="2">request</td></tr><tr><td>NPIWEB XML要素</td><td>NiiGF.dll</td></tr><tr><td>NPrint</td><td>NiiPrintEp</td></tr><tr><td>pdfprtID</td><td>i_pid</td></tr><tr><td>debug</td><td>i_debug</td></tr><tr><td>data</td><td>i_dat</td></tr><tr><td>size</td><td>i_size</td></tr><tr><td colspan="2">response</td></tr><tr><td>NiiGF.dll</td><td>NPIWEB XML要素</td></tr><tr><td>NiiPrintEp</td><td>NPrint</td></tr><tr><td>関数戻り値</td><td>result</td></tr><tr><td>i_pid</td><td>pdfprtID</td></tr><tr><td>o_jobid</td><td>jobID</td></tr></table>		request		NPIWEB XML要素	NiiGF.dll	NPrint	NiiPrintEp	pdfprtID	i_pid	debug	i_debug	data	i_dat	size	i_size	response		NiiGF.dll	NPIWEB XML要素	NiiPrintEp	NPrint	関数戻り値	result	i_pid	pdfprtID	o_jobid	jobID
request																												
NPIWEB XML要素	NiiGF.dll																											
NPrint	NiiPrintEp																											
pdfprtID	i_pid																											
debug	i_debug																											
data	i_dat																											
size	i_size																											
response																												
NiiGF.dll	NPIWEB XML要素																											
NiiPrintEp	NPrint																											
関数戻り値	result																											
i_pid	pdfprtID																											
o_jobid	jobID																											

要素名	NPrinterInit	
request		
子要素	型	説明
pdfprtID	文字列	プリンタID
response		
子要素	型	説明
result	数値	エラーコード(※53ページを参照)
pdfprtID	文字列	プリンタID
処理内容		

・指定されたプリンタIDに登録されている初期化コマンドを出力します。

EasyPrint GF関数の「NiiPrinterInitEp」を呼出して処理を行います。  
XMLとGF関数の対応は以下の通りとなります。

request	
NPIWEB XML要素	NiiGF.dll
NPrinterInit pdfprtID	NiiPrinterInitEp i_pid
response	
NiiGF.dll	NPIWEB XML要素
NiiPrinterInitEp 関数戻り値 i_pid	NPrinterInit result pdfprtID

要素名	NGetStatus	
request		
子要素	型	説明
pdfprtID	文字列	プリンタID
response		
子要素	型	説明
result	数値	エラーコード(※53ページを参照)
pdfprtID	文字列	プリンタID
status	数値	ステータス値
処理内容		

・プリンタステータスを取得します。  
responseのstatusに0～255の範囲で値が取得できます。  
0～7ビットの値でステータスを判断します。  
各プリンタによってステータス値の意味は異なりますので  
プリンタの仕様書を参照して下さい。

EasyPrint GF関数の「**NiiGetStEp**」を呼出して処理を行います。  
XMLとGF関数の対応は以下の通りとなります。

request	
NPIWEB XML要素	NiiGF.dll
NGetStatus pdfprtID	NiiGetStEp i_pid
response	
NiiGF.dll	NPIWEB XML要素
NiiGetStEp 関数戻り値 i_pid o_st	NGetStatus result pdfprtID status

要素名	NGetInfo	
request		
子要素	型	説明
pfdprtID	文字列	プリンタID
ID	数値	拡張情報種別ID(1～31)
response		
子要素	型	説明
result	数値	エラーコード(※53ページを参照)
pfdprtID	文字列	プリンタID
ID	数値	拡張情報種別ID(1～31)
data	文字列 ※	拡張情報データ
処理内容		

・指定された種別の対応する拡張情報を取得します。

※ 応答のdataに格納される拡張情報データは  
種別ID2、3、4の場合はASCII文字列表記、それ以外の場合は  
16進数文字列表記となります。

EasyPrint GF関数の「**NiiGetInfoEp**」を呼出して処理を行います。  
XMLとGF関数の対応は以下の通りとなります。

request	
NPIWEB XML要素	NiiGF.dll
NGetInfo	NiiGetInfoEp
pdfprtID	i_pid
ID	i_id
response	
NiiGF.dll	NPIWEB XML要素
NiiGetInfoEp	NGetInfo
関数戻り値	result
i_pid	pdfprtID
i_id	ID
o_dat	data

要素名	NSSetText	
request		
子要素	型	説明
formatID	文字列	フォーマットID
txtID	文字列	テキストID
data	文字列	設定する文字列データ
response		
子要素	型	説明
result	数値	エラーコード(※53ページを参照)
formatID	文字列	フォーマットID
txtID	文字列	テキストID
処理内容		

・指定したフォーマットIDに登録されているテキストIDのデータを設定します。

EasyPrint GF関数の「**NiiSetTextEp**」を呼出して処理を行います。  
XMLとGF関数の対応は以下の通りとなります。

request			
NPIWEB XML要素		NiiGF.dll	
NSSetText	formatID	NiiSetTextEp	i_fid
	txtID		i_txtid
	data		i_dat
response			
NiiGF.dll		NPIWEB XML要素	
NiiSetTextEp	関数戻り値	NSSetText	result
	i_fid		formatID
	i_txtid		txtID

要素名	NGetText	
request		
子要素	型	説明
formatID	文字列	フォーマットID
txtID	文字列	テキストID
response		
子要素	型	説明
result	数値	エラーコード(※53ページを参照)
formatID	文字列	フォーマットID
txtID	文字列	テキストID
data	文字列	取得した文字列データ
処理内容		

・指定したフォーマットIDに登録されているテキストIDのデータを取得します。

EasyPrint GF関数の「**NiiGetTextEp**」を呼出して処理を行います。  
XMLとGF関数の対応は以下の通りとなります。

request	
NPIWEB XML要素	NiiGF.dll
NGetText	NiiGetTextEp
formatID	i_fid
txtID	i_txtid
response	
NiiGF.dll	NPIWEB XML要素
NiiGetTextEp	NGetText
関数戻り値	result
i_fid	formatID
i_txtid	txtID
o_dat	data

要素名	NSetBarcode	
request		
子要素	型	説明
formatID	文字列	フォーマットID
bcdID	文字列	バーコードID
data	16進数文字列	設定するバーコードデータ
size	数値	バーコードサイズ
response		
子要素	型	説明
result	数値	エラーコード(※53ページを参照)
formatID	文字列	フォーマットID
bcdID	文字列	バーコードID
処理内容	<p>・指定したフォーマットIDに登録されているバーコードIDのデータを設定します。 要求時のdataは16進数文字列で指定します。0x等の表記は必要ありません。 例) データを「123」と指定する場合     data ⇒ 303132     size ⇒ 3</p> <p>EasyPrint GF関数の「<b>NiiSetBarcodeEp</b>」を呼出して処理を行います。 XMLとGF関数の対応は以下の通りとなります。</p>	

request			
NPIWEB XML要素		NiiGF.dll	
NSetBarcode	formatID	NiiSetBarcodeEp	i_fid
	bcdID		i_barid
	data		i_dat
	size		i_size
response			
NiiGF.dll		NPIWEB XML要素	
NiiSetBarcodeEp	関数戻り値	NSetBarcode	result
	i_fid		formatID
	i_barid		bcdID

要素名	NGetBarcode	
request		
子要素	型	説明
formatID	文字列	フォーマットID
bcdID	文字列	バーコードID
response		
子要素	型	説明
result	数値	エラーコード(※53ページを参照)
formatID	文字列	フォーマットID
bcdID	文字列	バーコードID
data	16進数文字列	取得したバーコードデータ
処理内容		

・指定したフォーマットIDに登録されているバーコードIDのデータを取得します。  
応答時のdataは16進数文字列となります。0x等の表記は付与されません。

EasyPrint GF関数の「**NiiGetBarcodeEp**」を呼出して処理を行います。  
XMLとGF関数の対応は以下の通りとなります。

request	
NPIWEB XML要素	NiiGF.dll
NGetBarcode	NiiGetBarcodeEp
formatID	i_fid
bcdID	i_barid
response	
NiiGF.dll	NPIWEB XML要素
NiiGetBarcodeEp	NGetBarcode
関数戻り値	result
i_fid	formatID
i_barid	bcdID
data	data



要素名	NSetBarcode2																									
request																										
子要素	型	説明																								
formatID	文字列	フォーマットID																								
bcdID	文字列	バーコードID																								
data	文字列	設定するバーコードデータ																								
response																										
子要素	型	説明																								
result	数値	エラーコード(※53ページを参照)																								
formatID	文字列	フォーマットID																								
bcdID	文字列	バーコードID																								
処理内容																										
<div>・指定したフォーマットIDに登録されているバーコードIDのデータを設定します。 要求時のdataはASCII文字列で指定します。</div> <div>EasyPrint GF関数の「<b>NiiSetBarcodeEp2</b>」を呼出して処理を行います。 XMLとGF関数の対応は以下の通りとなります。</div>																										
<table><tr><th colspan="2">request</th></tr><tr><th>NPIWEB XML要素</th><th>NiiGF.dll</th></tr><tr><td>NSetBarcode2</td><td>NiiSetBarcodeEp2</td></tr><tr><td>formatID</td><td>i_fid</td></tr><tr><td>bcdID</td><td>i_barid</td></tr><tr><td>data</td><td>i_dat</td></tr><tr><th colspan="2">response</th></tr><tr><th>NiiGF.dll</th><th>NPIWEB XML要素</th></tr><tr><td>NiiSetBarcodeEp2</td><td>NSetBarcode2</td></tr><tr><td>関数戻り値</td><td>result</td></tr><tr><td>i_fid</td><td>formatID</td></tr><tr><td>i_barid</td><td>bcdID</td></tr></table>			request		NPIWEB XML要素	NiiGF.dll	NSetBarcode2	NiiSetBarcodeEp2	formatID	i_fid	bcdID	i_barid	data	i_dat	response		NiiGF.dll	NPIWEB XML要素	NiiSetBarcodeEp2	NSetBarcode2	関数戻り値	result	i_fid	formatID	i_barid	bcdID
request																										
NPIWEB XML要素	NiiGF.dll																									
NSetBarcode2	NiiSetBarcodeEp2																									
formatID	i_fid																									
bcdID	i_barid																									
data	i_dat																									
response																										
NiiGF.dll	NPIWEB XML要素																									
NiiSetBarcodeEp2	NSetBarcode2																									
関数戻り値	result																									
i_fid	formatID																									
i_barid	bcdID																									

要素名	NGetBarcode2																									
request																										
子要素	型	説明																								
formatID	文字列	フォーマットID																								
bcdID	文字列	バーコードID																								
response																										
子要素	型	説明																								
result	数値	エラーコード(※53ページを参照)																								
formatID	文字列	フォーマットID																								
bcdID	文字列	バーコードID																								
data	文字列	取得したバーコードデータ																								
処理内容																										
<p>・指定したフォーマットIDに登録されているバーコードIDのデータを取得します。 応答時のdataはASCII文字列となります。</p> <p>EasyPrint GF関数の「<b>NiiGetBarcodeEp2</b>」を呼出して処理を行います。 XMLとGF関数の対応は以下の通りとなります。</p>																										
<table><tr><td colspan="2">request</td></tr><tr><td>NPIWEB XML要素</td><td>NiiGF.dll</td></tr><tr><td>NGetBarcode2</td><td>NiiGetBarcodeEp2</td></tr><tr><td>formatID</td><td>i_fid</td></tr><tr><td>bcdID</td><td>i_barid</td></tr><tr><td colspan="2">response</td></tr><tr><td>NiiGF.dll</td><td>NPIWEB XML要素</td></tr><tr><td>NiiGetBarcodeEp2</td><td>NGetBarcode2</td></tr><tr><td>関数戻り値</td><td>result</td></tr><tr><td>i_fid</td><td>formatID</td></tr><tr><td>i_barid</td><td>bcdID</td></tr><tr><td>o_dat</td><td>data</td></tr></table>			request		NPIWEB XML要素	NiiGF.dll	NGetBarcode2	NiiGetBarcodeEp2	formatID	i_fid	bcdID	i_barid	response		NiiGF.dll	NPIWEB XML要素	NiiGetBarcodeEp2	NGetBarcode2	関数戻り値	result	i_fid	formatID	i_barid	bcdID	o_dat	data
request																										
NPIWEB XML要素	NiiGF.dll																									
NGetBarcode2	NiiGetBarcodeEp2																									
formatID	i_fid																									
bcdID	i_barid																									
response																										
NiiGF.dll	NPIWEB XML要素																									
NiiGetBarcodeEp2	NGetBarcode2																									
関数戻り値	result																									
i_fid	formatID																									
i_barid	bcdID																									
o_dat	data																									

要素名	NsetImage2	
request		
子要素	型	説明
formatID	文字列	フォーマットID
imgID	文字列	イメージID
file	文字列	設定するbmpファイル名(パス含む)
response		
子要素	型	説明
result	数値	エラーコード(※53ページを参照)
formatID	文字列	フォーマットID
imgID	文字列	イメージID
処理内容		

・指定フォーマットIDに登録されているイメージIDのデータを設定します。

EasyPrint GF関数の「**NiiSetImageEp2**」を呼出して処理を行います。  
XMLとGF関数の対応は以下の通りとなります。

request	
NPIWEB XML要素	NiiGF.dll
NsetImage2	NiiSetImageEp2
formatID	i_fid
imgID	ImgID
file	i_file
response	
NiiGF.dll	NPIWEB XML要素
NiiSetImageEp2	NsetImage2
関数戻り値	result
i_fid	formatID
i_imgid	imgID

要素名	NSSetVisible	
request		
子要素	型	説明
formatID	文字列	フォーマットID
cmpID	文字列	コンポーネントID
visible	文字列	可視・不可視設定(1:表示、0:非表示)
response		
子要素	型	説明
result	数値	エラーコード(※53ページを参照)
formatID	文字列	フォーマットID
cmpID	文字列	コンポーネントID
処理内容		

- ・指定フォーマットIDに登録されているコンポーネントIDの可視・不可視状態を設定します。

EasyPrint GF関数の「**NiiSetVisibleEp**」を呼出して処理を行います。  
XMLとGF関数の対応は以下の通りとなります。

request	
NPIWEB XML要素	NiiGF.dll
NSetVisible	NiiSetVisibleEp
formatID	i fid
cmpID	i cid
visible	i flg
response	
NiiGF.dll	NPIWEB XML要素
NiiSetVisibleEp	NSetVisible
関数戻り値	result
i fid	formatID
i cid	cmpID

※-9999、-10000番台のエラーコードについては

要素名	NGetVisible																							
request																								
子要素	型	説明																						
formatID	文字列	フォーマットID																						
cmpID	文字列	コンポーネントID																						
response																								
子要素	型	説明																						
result	数値	可視化フラグ(0:非表示、1:表示) ※取得失敗時も0																						
formatID	文字列	フォーマットID																						
cmpID	文字列	コンポーネントID																						
処理内容																								
<p>・指定フォーマットIDに登録されているコンポーネントIDの可視・不可視状態を取得します。</p> <p>EasyPrint GF関数の「<b>NiiGetVisibleEp</b>」を呼出して処理を行います。 XMLとGF関数の対応は以下の通りとなります。</p>																								
<table><tr><td colspan="2">request</td></tr><tr><td>NPIWEB XML要素</td><td>NiiGF.dll</td></tr><tr><td>NGetVisible</td><td>NiiGetVisibleEp</td></tr><tr><td>formatID</td><td>i_fid</td></tr><tr><td>cmpID</td><td>i_cid</td></tr><tr><td colspan="2">response</td></tr><tr><td>NiiGF.dll</td><td>NPIWEB XML要素</td></tr><tr><td>NiiGetVisibleEp</td><td>NGetVisible</td></tr><tr><td>関数戻り値</td><td>result</td></tr><tr><td>i_fid</td><td>formatID</td></tr><tr><td>i_cid</td><td>cmpID</td></tr></table>			request		NPIWEB XML要素	NiiGF.dll	NGetVisible	NiiGetVisibleEp	formatID	i_fid	cmpID	i_cid	response		NiiGF.dll	NPIWEB XML要素	NiiGetVisibleEp	NGetVisible	関数戻り値	result	i_fid	formatID	i_cid	cmpID
request																								
NPIWEB XML要素	NiiGF.dll																							
NGetVisible	NiiGetVisibleEp																							
formatID	i_fid																							
cmpID	i_cid																							
response																								
NiiGF.dll	NPIWEB XML要素																							
NiiGetVisibleEp	NGetVisible																							
関数戻り値	result																							
i_fid	formatID																							
i_cid	cmpID																							

要素名	response	
request		
子要素	型	説明
－	－	NPI WEBアプリからの応答時要素のため、使用しません
response		
子要素	型	説明
result ※	数値 －	XML送信結果エラーコード requestで指定した子要素
処理内容	<p>・NPI WEBサービスからWEBブラウザへのデータ応答時に含まれる要素です。</p> <p>resultでXMLの送信結果が返されます。 0が応答された場合はXMLの処理が正常終了しています。</p> <p>EasyPrint XML要素のエラーコードについてはGF関数仕様書を参照して下さい。 (ただし下記、-9999と-10000番台を除く)</p> <p>-9999が応答された場合はdll関数の呼出に失敗しています。 NPrinterLib.dll および EasyPrint関連のdllが読み込める場所にあるか確認して下さい。</p> <p>-10000番台のエラーコードは次ページに示すエラーコード表を参照して下さい。 それ以外のエラーはプリンタードライバー内の関数実行時でのエラーとなります。ドライバーの説明書を確認して下さい。</p>	

エラー定数名称	エラー値	エラー内容
SUCCESS	0	正常終了
ERR_FILEOPEN	-10010	ファイルオープンエラーです
ERR_PRT_OUTPUT	-10013	プリンターへの出力エラーです
ERR_LOADFROMFILE	-10050	画像ファイルの読み込みに失敗しました
ERR_IMGDATA	-10056	画像データが不正です
ERR_EXINFONUMBER	-10080	拡張情報番号が不正です
ERR_GET_EXINFO	-10081	拡張情報番号の取得に失敗しました
ERR_EXINFOTIMEOUT	-10082	拡張情報番号の取得でタイムアウトしました
ERR_XMLILLEGAL	-10090	XMLが不正です
ERR_XML_TIMEOUT	-10091	XML受信にてデータ末尾検出でタイムアウトしました
ERR_XML_EXCEPTION	-10092	XML受信にて予期せぬエラーが発生しました
ERR_XML_CLOSED	-10093	XML受信にてクライアントとの接続が切断されました
ERR_ORIGIN	-10094	許可されていない送信元からのアクセスのため切断しました
ERR_XMLPARAM_VALUE	-10095	XML内のパラメータが不正です
ERR_XMLPARAM_NOTFOUND	-10096	XML内のパラメータが存在しません
ERR_XML_RESPONSE	-10097	XMLの応答送信に失敗しました
ERR_PDFCONVERT	-10101	PDFファイルの変換に失敗しました

※Android版 NPIWEBアプリケーションとの互換のため、空き番号があります。

## 拡張情報

種別 1	: 4 バイト (固定)	: 更新フラグ (4バイト)	<拡張ステータス> 1Byte : 7~0, 2Byte: 15~8, 3Byte : 23~16, 4Byte : 31~24
種別 2	: 32バイト (デリミタ)	: 更新フラグ (4バイト)	<モデル名>
種別 3	: 8 バイト (固定)	: 更新フラグ (4バイト)	<F/Wバージョン>
種別 4	: 8 バイト (固定)	: 更新フラグ (4バイト)	<Bootバージョン>
種別 5	: 4 バイト (固定)	: 更新フラグ (4バイト)	<予約>
種別 6	: 4 バイト (固定)	: 更新フラグ (4バイト)	<ヘッド通電ドットライン数>
種別 7	: 4 バイト (固定)	: 更新フラグ (4バイト)	<走行ドットライン数>
種別 8	: 4 バイト (固定)	: 更新フラグ (4バイト)	<カット回数>
種別 9	: 16バイト (固定)	: 更新フラグ (4バイト)	<ユーザーメンテナンスカウンタ : ヘッド通電ドットライン数, 走行ドットライン数, カット回数, 予備>
種別 10	: 16バイト (固定)	: 更新フラグ (4バイト)	<予約>
種別 11	: 64バイト (デリミタ)	: 更新フラグ (4バイト)	
種別 12	: 32バイト (デリミタ)	: 更新フラグ (4バイト)	
種別 13	: 32バイト (固定)	: 更新フラグ (4バイト)	<NV登録状況>
種別 14	: 32バイト (固定)	: 更新フラグ (4バイト)	<予約>
種別 15	: 16バイト (固定)	: 更新フラグ (4バイト)	
種別 16	: 16バイト (固定)	: 更新フラグ (4バイト)	
種別 17	: 16バイト (固定)	: 更新フラグ (4バイト)	
種別 18	: 16バイト (固定)	: 更新フラグ (4バイト)	
種別 19	: 8 バイト (固定)	: 更新フラグ (4バイト)	<印字完了通知 : 印字開始/終了コマンドの指定に より、終了コマンド処理時に任意ID, 終了ステータス記載>
種別 20	: 8 バイト (固定)	: 更新フラグ (4バイト)	<予約>
種別 21	: 8 バイト (固定)	: 更新フラグ (4バイト)	
種別 22	: 8 バイト (固定)	: 更新フラグ (4バイト)	
種別 23	: 8 バイト (固定)	: 更新フラグ (4バイト)	
種別 24	: 4 バイト (固定)	: 更新フラグ (4バイト)	<予約>
種別 25	: 4 バイト (固定)	: 更新フラグ (4バイト)	<転送完了通知 : 転送の完了したジョブID記載>
種別 26	: 4 バイト (固定)	: 更新フラグ (4バイト)	<予約>
種別 27	: 4 バイト (固定)	: 更新フラグ (4バイト)	<予約>
種別 28	: 2 バイト (固定)	: 更新フラグ (4バイト)	<F/Wチェックサム>
種別 29	: 2 バイト (固定)	: 更新フラグ (4バイト)	
種別 30	: 2 バイト (固定)	: 更新フラグ (4バイト)	
種別 31	: 2 バイト (固定)	: 更新フラグ (4バイト)	<通信状態情報 : USB : 0 x 0000固定 COM : 1バイト目CTS 2バイト目DSR ※更新フラグに最終信号状況取得タイムスタンプ>

※種別25、31を除きプリンタにて機能実装していない情報に付いては取得できた内容に妥当性はありません。  
※ここで記載した内容が全てのプリンタで使用出来るとは限りません。