

# PrinterSignalStation for ラベルプリンタ アプリケーションインターフェース 説明書

## ご注意

1. 本書の内容の一部または全部を無断転載することは禁止されています。
2. 本書の内容に関しては将来予告なしに変更することがあります。
3. NECの許可なく複製・改変などを行うことはできません。
4. 本書は内容について万全を期して作成致しましたが、万一ご不審な点や誤り、記載もれなどお気づきのことがありましたら、お買い求めの販売店にご連絡ください。
5. 運用した結果の影響については4項にかかわらず責任を負いかねますのでご了承ください。
6. 本製品を第三者に売却・譲渡する際は必ず本書も添えてください。

# 目 次

1	はじめに.....	3
2	PrinterSignalStation for ラベルプリンタが提供する機能.....	4
2.1	PrinterSignalStation for ラベルプリンタを使用する際のご注意.....	4
3	アプリケーションプログラムインターフェース (API).....	5
3.1	プリンタステータス情報 .....	5
3.1.1	NesTSGetPrinterStatusEx().....	5
3.1.2	プリンタステータス情報の内容.....	6
3.2	プリンターオプション情報 .....	7
3.2.1	NesTSGetPaperSource().....	7
3.2.2	プリンターオプション情報の内容.....	7
3.3	プリンター状態変更通知.....	8
3.3.1	NesTSPrinterNotifyStart()／NesTSPrinterNotifyEnd().....	8
3.3.2	プリンタステータス情報の取得.....	9
A	API一覧.....	10
B	API仕様 .....	11
B.1	NesTSPrinterNotifyStart().....	11
B.2	NesTSPrinterNotifyEnd().....	12
B.3	NesTSGetPrinterStatusEx() .....	13
B.4	NesTSGetPaperSource().....	15
C	アプリケーション作成例.....	16
C.1	プリンター監視開始時の処理 .....	16
C.2	状態変更時の処理 .....	17
C.3	監視終了時の処理 .....	18

# 1. はじめに

本書では、PrinterSignalStation for ラベルプリンタが提供するプリンター監視機能を、アプリケーションから使用するためのアプリケーションプログラムインターフェース（API）について説明します。

PrinterSignalStation for ラベルプリンタを使用してプリンター監視を行うWindowsアプリケーションを開発する際にご覧ください。

なお、PrinterSignalStation for ラベルプリンタの操作方法や動作については、CD-ROMに収録されているオンラインマニュアルに記載していますので、本書と併せてご覧ください。



本書はWindows APIおよびWindowsアプリケーション作成についての知識を持つ技術者向けの説明書です。

## 2. PrinterSignalStation for ラベルプリンタが提供する機能

PrinterSignalStation for ラベルプリンタはアプリケーションに対して、以下のプリンター監視情報を提供します。

### (1) プリンターステータス情報\* 1

PrinterSignalStation for ラベルプリンタが保持しているプリンターのステータス情報をアプリケーションに提供します。  
提供される情報は、「用紙なし」や「オフライン」などのプリンターエラーに関する情報のほか、「印刷中」や「ジョブ削除中」などのジョブ操作に関する情報が含まれます。

### (2) プリンターオプション情報\* 2

PrinterSignalStation for ラベルプリンタが保持しているプリンターのオプション情報をアプリケーションに提供します。  
提供される情報は、「カッターモジュール」や「ハクリモジュール」などのオプションに関する情報が含まれます。

### (3) プリンター状態変更通知

PrinterSignalStation for ラベルプリンタが、プリンター状態（プリンターエラーの発生や、オプションの構成）の変化をアプリケーションに通知します。プリンター状態変更通知と上記の（1）プリンターステータス情報および（2）プリンターオプション情報を組み合わせることで、アプリケーションから素早くプリンターの状態変化を監視することができます。

## 2.1 PrinterSignalStation for ラベルプリンタを使用する際のご注意

PrinterSignalStation for ラベルプリンタを使用してプリンター監視を行うWindowsアプリケーションを利用する際は、以下の点にご注意ください。

### • 印刷完了監視を行う際のご注意

[環境設定]-[通知の設定]タブの[詳細設定]ダイアログにおいて[印刷時にジョブの同期を行う]を有効にし、プリンターの印刷完了監視を行う場合は、以下の点に注意してください。

- － プリンターに添付のプリンタソフトウェアCD-ROMに収録されているPrinterSignalStation for ラベルプリンタ対応のプリンタードライバーをご使用ください。
- － スプーラーを介さずにプリンターに直接印刷データを送信するアプリケーションをご使用の場合、プリンター状態監視および、プリンターの印刷完了監視が正常に動作しない場合があります。

### • プリンタードライバーのスプール設定に関するご注意

プリンタードライバーのプロパティ [詳細設定]シート内のスプール設定は、[印刷ドキュメントをスプールし、プログラムの印刷処理を高速に行う]-[すぐに印刷データをプリンタに送る]（プリンタードライバーの標準設定）でご利用ください。この設定で印刷を行った場合、スプーラーはアプリケーションからの印刷要求をすべてスプールし、すぐにアプリケーションに制御を返すため、印刷及び、プリンター監視をスムーズかつ、より正確に行うことができます。

スプール設定を[プリンタに直接印刷データを送る]に設定している場合、スプーラーは、アプリケーションからの印刷要求をスプールせずに印刷データをプリンターに直接送信します。そのため、プリンターエラー発生中などにスプーラー APIを実行すると、スプーラーが要求されたAPI処理を完了できず、アプリケーションに制御を戻さないことがあります。

その場合は、プリンターで発生しているプリンターエラーを解除するか、印刷中のジョブを削除することで、アプリケーション側に制御を戻すことができます。

\*1 PrinterSignalStation for ラベルプリンタの監視設定が「高度監視」の場合と「通常監視」の場合とで、提供される情報の内容が異なります。  
通知を受け取らない場合でも、イベント通知によるプリンター状態変更通知サービスの開始と終了のインターフェースを実行する必要があります。

\*2 本機能は、PrinterSignalStation for ラベルプリンタの監視設定が「高度監視」に設定されている必要があります。  
通知を受け取らない場合でも、イベント通知によるプリンター状態変更通知サービスの開始と終了のインターフェースを実行する必要があります。

## 3. アプリケーションプログラムインターフェース (API)

PrinterSignalStation for ラベルプリンタでは、プリンター監視機能をアプリケーションでご利用いただくために、以下のAPIを用意しています。これらのAPIを利用することで、PrinterSignalStation for ラベルプリンタのプリンター監視機能をアプリケーションに組み込むことができます。

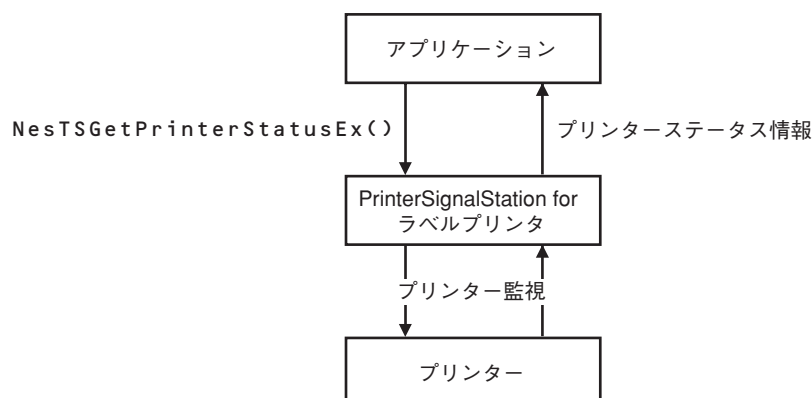
インターフェース	機能
NesTSGetPrinterStatusEx()	現在のプリンターステータス情報を取得する
NesTSGetPaperSource()	現在のプリンターオプション情報を取得する
NesTSPrinterNotifyStart()	プリンター状態変更通知サービスを開始する
NesTSPrinterNotifyEnd()	プリンター状態変更通知サービスを終了する

### 3.1 プリンターステータス情報

#### 3.1.1 NesTSGetPrinterStatusEx()

NesTSGetPrinterStatusEx()は、関数実行時点でPrinterSignalStation for ラベルプリンタが保持しているプリンターステータス情報をアプリケーションに返します。PrinterSignalStation for ラベルプリンタがアプリケーションに返す情報は、次の2種類に分類されます。

情報の種類	説明
ステータス情報	「用紙なし」や「オフライン」などのプリンターエラーに関する情報です。
ジョブ操作情報	「ジョブ削除中」や「印刷完了」などの印刷ジョブ操作に関する情報です。



プリンターステータス情報の取得

### 3.1.2 プリンタステータス情報の内容

プリンタステータス情報の内容は、以下の通りです。

#### (1) ステータス情報

アプリケーションがPrinterSignalStation for ラベルプリンタから取得できるステータス情報は、次の表の通りです。PrinterSignalStation for ラベルプリンタから取得できるステータス情報は、監視設定（高度監視または通常監視）で異なりますのでご注意ください。

ステータス情報	内 容	ステータス情報の取得可否	
		USB / TCP/IP	
		通常監視	高度監視
印刷可能	プリンターは印刷可能である	○	○
印刷中	プリンターは印刷中である	○	○
ハクリ待ち	プリンターはハクリ待ちである	△*1	○
オフライン (印刷可ランプ消灯)	プリンターはオフライン状態である	○	○
RFIDエラー	RFIDの処理に失敗した	△	○
コマンドエラー	プリンターでコマンドエラーが発生している	△*2	○
ヘッド断線エラー	印刷ヘッド断線エラーが発生している	△*2	○
用紙なし	プリンターの用紙がない	○	○
紙づまり	プリンターで紙づまりが発生している	△*2	○
リボンエラー	プリンターでリボンエラーが発生している	△*2	○
カバーオープン	プリンターのカバーが開いている	△*2	○
プリンタ装置エラー	プリンターで何らかの警告が発生している	△*2	○
ネットワークエラー	ネットワーク通信で問題が発生している	○	○
電源OFF/ケーブル断	プリンターの電源がOFFになっているか、コンピュータとの接続が切断されている	○	○
情報取得エラー	情報取得に失敗	○	○
情報取得中	プリンターからの情報取得中	○	○
印刷終了	印刷が終了した	○	○
ドキュメント情報あり	ドキュメント情報がある	○	○

○取得可能、×取得不可

\* 1 「印刷中」として検出。

\* 2 「オフライン（印刷可ランプ消灯）」として検出。

#### (2) ジョブ操作情報

アプリケーションがPrinterSignalStation for ラベルプリンタから取得できるジョブ操作情報は、次の表の通りです。これらの情報は、PrinterSignalStation for ラベルプリンタの監視設定およびプリンター接続ポートに関わらず取得可能です。

ジョブ操作情報	意 味
プリンター一時停止	プリンターは一時停止中である
ドキュメント削除中	印刷ジョブは、削除処理中である
ドキュメント削除終了	印刷ジョブの削除処理が終了した
ドキュメント一時停止	印刷ジョブの送信が一時停止されている
ドキュメント準備中	印刷ジョブは、スプール中である
ドキュメント印刷終了	印刷ジョブの印刷が終了した

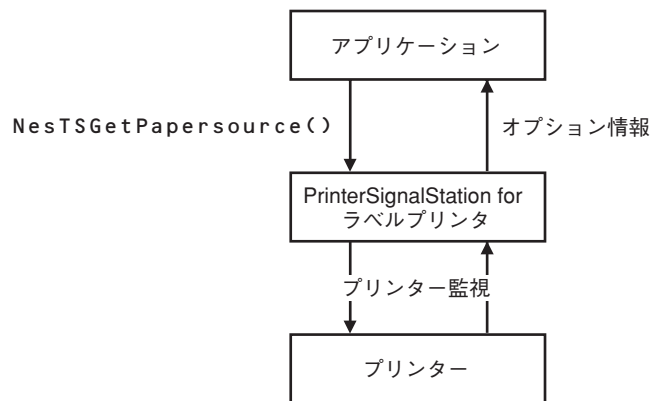
## 3.2 プリンターオプション情報

### 3.2.1 NesTSGetPaperSource()

NesTSGetPaperSource()は、関数実行時点でPrinterSignalStation for ラベルプリンタが保持しているプリンターオプション情報をアプリケーションに返します。

ただし、PrinterSignalStation for ラベルプリンタがプリンターオプション情報を取得できるのは次の条件に当てはまる場合に限られますのでご注意ください。

- (1) PrinterSignalStation for ラベルプリンタが「高度監視」で動作していること
- (2) PrinterSignalStation for ラベルプリンタが [ 状態ダイアログ ] でオプションを正しく表示していること



プリンター給紙装置情報の取得

### 3.2.2 プリンターオプション情報の内容

PrinterSignalStation for ラベルプリンタは、プリンターからオプション情報を取得します。プリンターが装着可能なオプションの種類とそれらの装着状態を調べることができます。

利用可能なオプションは、プリンターの機種によって異なります。オプションの詳細については、プリンターのユーザーズマニュアルをご覧ください。各プリンター機種から取得できるオプション情報は次の表の通りです。

オプション	内容
カッターモジュール	カッターモジュールの装着状況を取得できます。
ハクリモジュール	ハクリモジュールの装着状況を取得できます。

## 3.3 プリンター状態変更通知

PrinterSignalStation for ラベルプリンタは、監視対象プリンターのステータス情報が変化した場合に、プリンターの状態変化をアプリケーションに通知することができます。

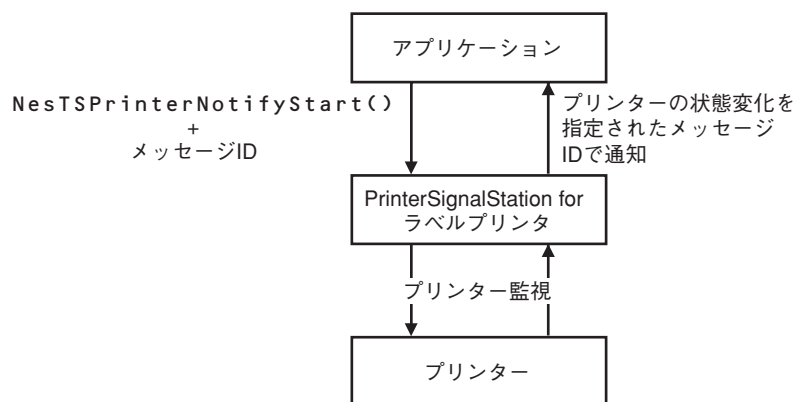
アプリケーションは、プリンター状態変更通知を利用することにより、プリンターの状態変化を素早く認識することができます。

### 3.3.1 NesTSPrinterNotifyStart() / NesTSPrinterNotifyEnd()

アプリケーションは、NesTSPrinterNotifyStart() を実行した時点から、NesTSPrinterNotifyEnd() を実行するまでの間、PrinterSignalStation が提供するプリンター状態変更通知機能を利用することができます。

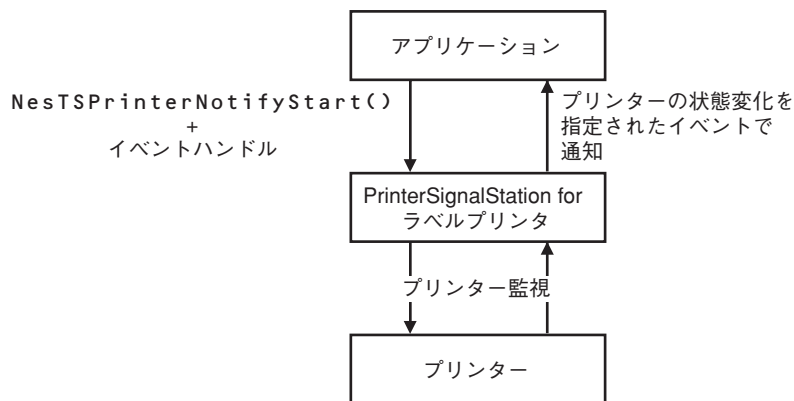
アプリケーションは、PrinterSignalStation からのプリンター状態変更通知方法として次の2種類のいずれかを選択することができます。

通知方法	内容
メッセージ通知	プリンターに状態変化があった場合に、PrinterSignalStation for ラベルプリンタはウィンドウメッセージの形態でアプリケーションへ通知します。 アプリケーションは、ウィンドウプロシージャによって、指定したメッセージIDを監視してください。
イベント通知	プリンターに状態変化があった場合に、PrinterSignalStation for ラベルプリンタはイベントの形態でアプリケーションへ通知します。 アプリケーションは、指定したイベントの発生を WaitForSingleObject() などの待機関数を利用して待機してください。



メッセージ通知の流れ

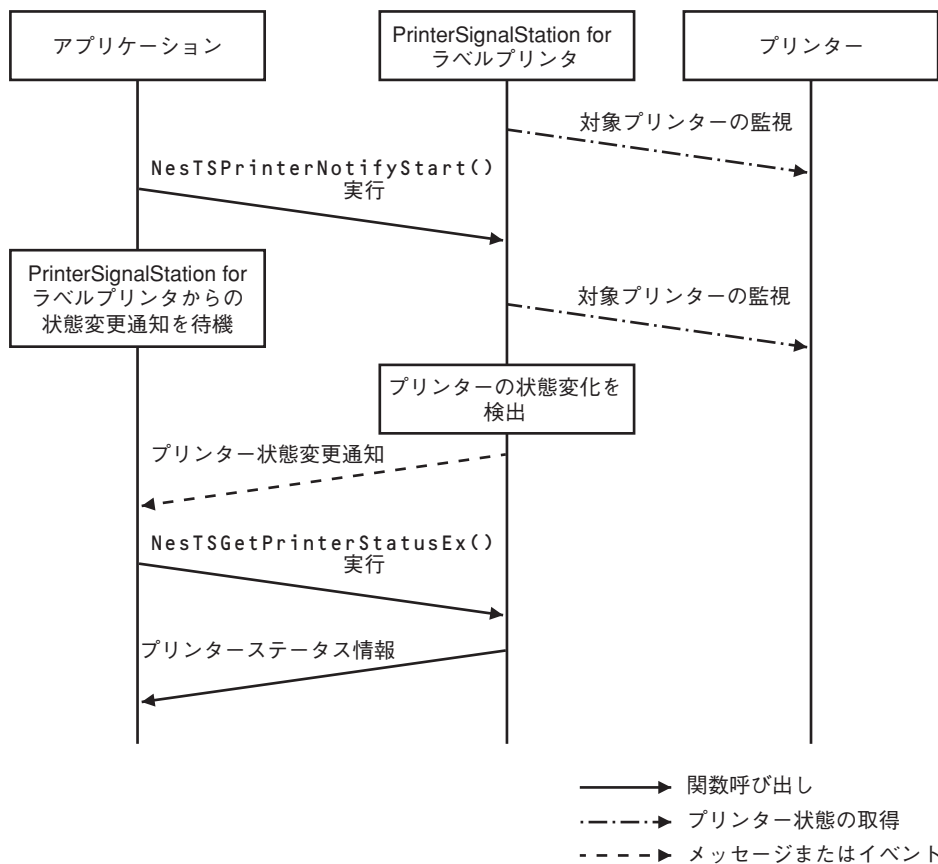




イベント通知の流れ

### 3.3.2 プリンターステータス情報の取得

アプリケーションは、NesTSPrinterNotifyStart()を実行後、PrinterSignalStation for ラベルプリンタからプリンター状態変更通知を受けるまで待機してください。アプリケーションは、PrinterSignalStation for ラベルプリンタからプリンター状態変更通知を受けた後で、NesTSGetPrinterStatusEx()あるいはNesTSGetPaperSource()を実行することにより、最新のプリンターステータス情報を取得することができます。



プリンター状態変更通知によるステータス情報取得の流れ

# 付録A. API一覧

PrinterSignalStation for ラベルプリンタのインターフェースは以下の通りです。

インターフェース	機能	参照
NesTSGetPrinterStatusEx()	現在のプリンターステータス情報を取得する。	<a href="#">11ページ</a>
NesTSGetPaperSource()	現在のプリンターオプション情報を取得する。	<a href="#">12ページ</a>
NesTSPrinterNotifyStart()	プリンター状態変更通知サービスを開始する。	<a href="#">13ページ</a>
NesTSPrinterNotifyEnd()	プリンター状態変更通知サービスを終了する。	<a href="#">15ページ</a>

# 付録B. API仕様

各API関数の仕様詳細は次の通りです。

## B.1 NesTSPrinterNotifyStart()

- 書式

```

BOOL WINAPI NesTSPrinterNotifyStart(
    TCHAR*      IN      pszPrinterName,
    DWORD       IN      dwStatus,
    NOTIFY*     IN      pNotify,
    HANDLE*     IN OUT  phNotify)

```

- 引数

pszPrinterName	:	監視を行うプリンター名を示す文字列へのポインタ。
dwStatus	:	Reserved。0固定。
pNotify	:	状態の通知先を示す以下の構造体へのポインタ。
<pre> typedef struct _NOTIFY {     DWORD      dwFlags; // 以下のいずれかの値を示すビットフィールド     #define NOTIFY_MSG      0x00000001     #define NOTIFY_EVENT   0x00000002     union {         HANDLE hEvent; // dwFlagsがNOTIFY_EVENTの場合に使用する。                         // 通知先のイベントハンドル         struct {             HWND  hWnd; // dwFlagsがNOTIFY_MSGの場合に使用する。                         // 通知先のウィンドウハンドル             UINT  uiMsg; // 通知するメッセージID         } st1;     } un1; } NOTIFY, *PNOTIFY; </pre>		
phNotify	:	管理用のハンドルを示すアドレス。

- 戻り値

0以外：成功（監視開始）

0：失敗

- 機能

プリンターの状態通知サービスを開始する。

## B.2 NesTSPrinterNotifyEnd()

- 書式

```
BOOL WINAPI NesTSPrinterNotifyEnd(  
    HANDLE          IN          hNotify)
```

- 引数

hNotify: 監視を終了するハンドル。(NesTSPrinterNotifyStart())で取得したハンドル)

- 戻り値

0以外: 成功 (監視終了)

0: 失敗

- 機能

プリンターの状態通知サービスを終了する。

## B.3 NesTSGetPrinterStatusEx()

- 書式

```

BOOL WINAPI NesTSGetPrinterStatusEx (
    TCHAR*           IN           pszPrinterName,
    DWORD           IN OUT       dwStatus[5],
    DWORD           IN           dwSize,
    STATUSVALUE*    IN OUT       pStatus,
    DWORD*          IN OUT       lpdwNeeded)
  
```

- 引数

pszPrinterName: 情報を取得するプリンター名を示す文字列へのポインタ。

dwStatus: プリンターの状態を示す5つのDWORD。ビットフィールドでプリンターの状態を示す。下記で指定されていないビットは予約されている。

- dwStatus[0]

名前	値	意味
NESTS_STS1_NOERROR	0x00000000	正常
NESTS_STS1_WARNING	0x20000000	警告発生中
NESTS_STS1_ERROR	0x40000000	エラー発生中
NESTS_STS1_UNKNOWN	0x80000000	不明
NESTS_STS1_ONLINE	0x00000001	印刷可能
NESTS_STS1_PRINTING	0x00000002	印刷中
NESTS_STS1_OFFLINE	0x00000040	オフライン
NESTS_STS1_COVEROPEN	0x00000080	カバーオープン
NESTS_STS1_PAPERJAM	0x00000100	紙づまり
NESTS_STS1_NOPAPER	0x00001000	用紙なし
NESTS_STS1_POWEROFF	0x00010000	電源OFF
NESTS_STS1_NETERROR	0x00020000	ネットワーク通信エラー
NESTS_STS1_CALLERROR	0x00080000	保守員コールエラー
NESTS_STS1_RIBBONJAM	0x00200000	リボンエラー
NESTS_STS1_PLGINSTATE	0x01000000	プラグイン情報あり (dwStatus[1]に情報あり)
NESTS_STS1_SPOOLERSTATE	0x08000000	印刷ジョブ情報あり (dwStatus[4]に情報あり)

- dwStatus[1]

名前	値	意味
NESTS_STS2_RIBBONERROR	0x00000001	リボンエラー
NESTS_STS2_HEADERROR	0x00000002	ヘッド断線エラー
NESTS_STS2_COMMANDERROR	0x00000004	コマンドエラー
NESTS_STS2_RFIDERROR	0x00000008	RFIDエラー
NESTS_STS2_PEELOFFWAIT	0x00000010	ハクリ待ち

- dwStatus[2]

予約

- dwStatus[3]

予約

## - dwStatus[4]

名前	値	意味	備考
NESTS_STS5_PRINTERPAUSED	0x00000001	プリンター一時停止	
NESTS_STS5_JOBDELETING	0x00020000	ドキュメント削除中	一時的なステータス
NESTS_STS5_JOBDELETED	0x00010000	ドキュメント削除終了	一時的なステータス
NESTS_STS5_JOBPAUSED	0x00040000	ドキュメント一時停止	
NESTS_STS5_JOBSPoolING	0x00080000	ドキュメント準備中	
NESTS_STS5_JOBPRINTED	0x00100000	ドキュメント印刷終了	一時的なステータス

dwSize : pStatusのサイズを示す値。

pStatus : 現在の状態を示すバッファへのポインタ。以下に示す構造体。

```
typedef struct _STATUSVALUE {
    DWORD        dwFlags;           // 現在の状態を表すフラグ
    DWORD        dwReserved1;      // 予約
    DWORD        dwReserved2;      // 予約
    TCHAR        pszStatusTip[256]; // ステータスパルーンを示す文字列
    TCHAR        pszStatusShort[32]; // ステータス略称を示す文字列
    TCHAR        pszStatusTitle[64]; // ステータスタイトルを示す文字列
    TCHAR        pszStatusDetail[1]; // ステータス詳細を示す文字列
} STATUSVALUE, *PSTATUSVALUE;
```

dwFlagsは以下のうちのいずれかの値

名前	値	意味
STATUSVALUE_NOERROR	0	正常
STATUSVALUE_WARNING	1	警告
STATUSVALUE_ERROR	2	エラー

lpdwNeeded : pStatusに書き込んだバイト数を示す値へのポインタ。  
dwSizeが0の場合、必要なバイト数を返す。

- 戻り値  
0以外 : 成功  
0 : 失敗
- 機能  
プリンターの現在の状態を取得する。

## B.4 NesTSGetPaperSource()

- 書式

```

BOOL WINAPI NesTSGetPaperSource(
    TCHAR*          IN          pszPrinterName,
    DWORD          IN          dwSize,
    SOURCEVALUE*   IN OUT      pSource,
    DWORD*         IN OUT      lpdwNeeded)

```

- 引数

**pszPrinterName** : 情報を取得するプリンター名を示す文字列へのポインタ。

**dwSize** : pSourceのサイズを示す値。

**pSource** : サポートオプションの情報を列挙するバッファへのポインタ。以下に示す構造体。

```

typedef struct _SOURCEVALUE {
    DWORD          dwID           // オプションの種類を示す値
    DWORD          dwFlags;      // オプションの状態を示すビットフィールド
                                // 以下のいずれかの値を示すビットフィールド
    #define SOURCE_ATTACH 0x00000001 // 装着済み
    TCHAR          pszName[64];  // オプション名を示す文字列
    TCHAR          pszStatus[64]; // オプションの状態を示す文字列
} SOURCEVALUE, *PSOURCEVALUE;

```

dwIDは以下のうちのいずれかの値。

名前	値	意味
SOURCEID_STRIP	101	ハクリモジュール
SOURCEID_CUT	102	カッターモジュール

**lpdwNeeded** : pSourceに書き込んだバイト数を示す値へのポインタ。  
dwSizeが0の場合、必要なバイト数を返す。

- 戻り値

0以外 : 成功

0 : 失敗

- 機能

プリンター固有のオプション情報を取得する。

# 付録C. アプリケーション作成例



[環境設定]-[通知の設定]タブの[詳細設定]ダイアログにおいて[印刷時にジョブの同期を行う]を有効にし、プリンターの印刷完了監視を行う場合は、以下の点に注意してください。

- プリンターに添付のプリンタソフトウェアCD-ROMに収録されているPrinterSignalStation for ラベルプリンタ対応のプリンタードライバーをご使用ください。
- スプーラーを介せずにプリンターに直接印刷データを送信するアプリケーションをご使用の場合、プリンター状態監視および、プリンターの印刷完了監視が正常に動作しない場合があります。

## C.1 プリンター監視開始時の処理

アプリケーションからプリンター監視を開始する場合、次のようにNesTSPrinterNotifyStart()をコールすることで、プリンター状態変更時にメッセージで通知されます。通常、WM\_CREATEやWM\_INITIALIZEDIALOG等のアプリケーション起動時に実行してください。

```
char* pPrinterName; // プリンター指定用
NOTIFY stNotify; // 状態変化を通知するための情報の構造体
HANDLE hNotify; // PrinterSignalStation for ラベルプリンタ管理用のHandle。監視終了時に使用する。

// 監視開始APIをコール
stNotify.dwFlags = NOTIFY_MSG; // 状態変更時にメッセージを受け取る
stNotify.un1.st1.hWnd = hWnd; // 通知するウィンドウHandleを指定
stNotify.un1.st1.uiMsg = WM_USER+1; // 複数のプリンターの監視を行う場合、
// プリンター毎にメッセージIDを変えます。

if (!NesTSPrinterNotifyStart(pPrinterName, // プリンター名を指定
                             0, // 0固定
                             &stNotify, // 監視情報
                             &hNotify)) // 管理用Handle
{
    // エラー処理
}
```



## C.2 状態変更時の処理

プリンターの状態が変わった場合、指定したウィンドウメッセージで通知されます。ウィンドウプロシージャに次の処理を追加することで、プリンター状態変更時に変更された状態を取得することが可能です。

```
// 状態変更をメッセージで受け取る
// 変数定義
char* pPrinterName; // プリンター指定用
DWORD dwStatus[5]; // プリンターの状態を示すDWORD
DWORD dwNeeded; // サイズ
STATUSVALUE* pStatus; // プリンターの状態を示すSTATUSVALUE構造体
SOURCEVALUE* pSourceValue; // プリンターのオプションモジュールを示すSOURCEVALUE構造体
DWORD dwNeeded; // 情報取得に必要なデータサイズ

switch(msg) {
case WM_USER+1: // stNotify.un1.st1.uiMsgで指定したメッセージを受け取ります。
// プリンターの状態が変化。情報取得APIをコール
NesTSGetPrinterStatusEx(
    pPrinterName, // プリンター名を指定
    dwStatus, // プリンターの状態を取得
    0, // pStatusのサイズ
    NULL, // STATUSVALUE構造体へのポインター
    &dwNeeded // pStatusに必要なサイズ
);
pStatus = (STATUSVALUE*)GlobalAlloc(GPTR,dwNeeded); //pStatusのメモリーを確保
if (!NesTSGetPrinterStatusEx(
    pPrinterName, // プリンター名を指定
    dwStatus, // プリンターの状態を取得
    dwNeeded, // pStatusのサイズ
    pStatus, // STATUSVALUE構造体へのポインター
    &dwNeeded // pStatusに必要なサイズ
)) {
// エラー処理
}

// オプション構成取得APIをコール
NesTSGetPaperSource(
    pPrinterName, //プリンター名を指定
    0, // pSourceValueのサイズ
    NULL, //SOURCEVALUE構造体へのポインター
    &dwNeeded // pSourceValueに必要なサイズ
);
pSourceValue = (SOURCEVALUE*)GlobalAlloc(GPTR,dwNeeded); //pSourceValueのメモリーを確保
if (!NesTSGetPaperSource(
    pPrinterName, //プリンター名を指定
    dwNeeded, // pSourceValueのサイズ
    pSourceValue, //SOURCEVALUE構造体へのポインター
    &dwNeeded // pSourceValueに必要なサイズ
)) {
// エラー処理
}

// 取得した状態の処理

GlobalFree(pStatus); //pStatusのメモリーを開放
GlobalFree(pSourceValue); // pSourceValueのメモリーを開放
break;
// その他のcase処理
```

### C.3 監視終了時の処理

アプリケーションで監視が不要となった場合、以下のようにNesTSPrinterNotifyEnd()をコールすることで、監視サービスを停止します。NesTSPrinterNotifyStart()の戻り値であるhNotifyを渡して、コールする必要があります。

```
// 監視終了APIコール  
NesTSPrinterNotifyEnd(hNotify); // NesTSPrinterNotifyStart()で取得したハンドルを指定して、コールします。
```