

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2009-146362
(P2009-146362A)

(43) 公開日 平成21年7月2日(2009.7.2)

(51) Int.Cl. F 1 テーマコード (参考)
G 0 6 F 3 / 1 2 (2 0 0 6 . 0 1) G 0 6 F 3 / 1 2 C 5 B 0 2 1

審査請求 未請求 請求項の数 7 O L (全 9 頁)

(21) 出願番号	特願2007-326056 (P2007-326056)	(71) 出願人	000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号
(22) 出願日	平成19年12月18日(2007.12.18)	(74) 代理人	100090538 弁理士 西山 恵三
		(74) 代理人	100096965 弁理士 内尾 裕一
		(72) 発明者	高木 義博 東京都大田区下丸子3丁目30番2号キヤ ノン株式会社内
		Fターム(参考)	5B021 AA01 CC00

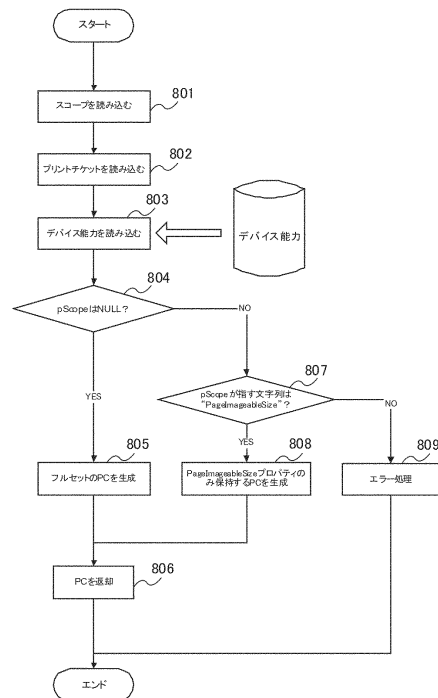
(54) 【発明の名称】 情報処理装置、情報処理方法、プログラム、及び記憶媒体

(57) 【要約】

【課題】 プリントチケットを使った印刷システムでは、アプリケーションが、デバイス能力を取得する際にもプリントケーパビリティと呼ばれる（プリントチケットと似たような）XMLデータを得ることになる。一般にプリントケーパビリティのデータサイズは肥大化する傾向にあるため、XMLの処理速度の低下が問題になる。

【解決手段】 プリントチケット、及びプリントケーパビリティのスコープを入力し、入力されたプリントチケット及びプリントケーパビリティスコープに基づきプリントケーパビリティを生成し、アプリケーションに返却することを特徴とする。

【選択図】 図 8



【特許請求の範囲】**【請求項 1】**

プリントチケット、及びプリントケーパビリティのスコープを入力する入力手段と、
前記入力手段により入力されたプリントチケットとプリントケーパビリティのスコープ
とに基づきプリントケーパビリティを生成する手段と、

前記生成したプリントケーパビリティをアプリケーションに返却する返却手段とを有す
ることを特徴とする情報処理装置。

【請求項 2】

前記入力手段は、アプリケーションから直接入力するインターフェースを介してプリン
トチケット、及びプリントケーパビリティのスコープを入力することを特徴とする請求項
1記載の情報処理装置。

10

【請求項 3】

プリントチケット、及びプリントケーパビリティのスコープを入力する入力ステップと

、
前記入力手段により入力されたプリントチケットとプリントケーパビリティのスコープ
とに基づきプリントケーパビリティを生成するステップと、

前記生成したプリントケーパビリティをアプリケーションに返却する返却ステップとを
有することを特徴とする情報処理装置。

【請求項 4】

前記入力ステップは、アプリケーションから直接入力するインターフェースを介してプ
リントチケット、及びプリントケーパビリティのスコープを入力することを特徴とする請
求項 3記載の情報処理方法。

20

【請求項 5】

プリントチケット、及びプリントケーパビリティのスコープを入力する入力ステップと

、
前記入力手段により入力されたプリントチケットとプリントケーパビリティのスコープ
とに基づきプリントケーパビリティを生成するステップと、

前記生成したプリントケーパビリティをアプリケーションに返却する返却ステップとを
コンピュータに実行させるためのプログラム。

【請求項 6】

前記入力ステップは、アプリケーションから直接入力するインターフェースを介してプ
リントチケット、及びプリントケーパビリティのスコープを入力することを特徴とする請
求項 5記載のプログラム。

30

【請求項 7】

請求項 5乃至 6いずれかに記載のプログラムを記憶したコンピュータが読み取り可能な
記憶媒体。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、情報処理装置、情報処理方法、プログラム、及び記憶媒体に関するものであ
る、特に印刷設定に関するものである。

40

【背景技術】**【0002】**

M i c r o s o f t 社のオペレーティングシステム (O S) である M i c r o s o f t
W i n d o w s (登録商標) V i s t a において印刷システムが大きく変更された。従
来の G D I (G r a p h i c D e v i c e I n t e r f a c e) ベースの印刷システ
ムと互換性を維持しつつ、 X P S (X M L P a p e r S p e c i f i c a t i o n)
ベースの新しい印刷システムが導入された (<http://www.microsoft.com/japan/whdc/device/print/default.mspx>) 。

50

【0003】

XPSとは、電子文書フォーマットの一種である。その中身は、文書構造、描画情報、印刷設定などの情報を保持するXMLとビットマップやフォントなどのバイナリリソースといった様々なパーツから構成される。XPSを構成する特に重要なパーツを図1にて示す。FixedDocumentSequence101をジョブ、FixedDocument102をドキュメント、FixedPage103をページと理解するとよい。FixedDocumentSequenceパーツ201は必ず1つだけ存在し、複数のFixedDocumentパーツ202を保持する。さらに各々のFixedDocumentパーツ202は複数のFixedPageパーツ203を保持する。さらに、これら任意のFixedDocumentSequenceパーツ201、FixedDocumentパーツ202、FixedPageパーツ203には各々印刷設定を関連付けることができる(図2)。この印刷設定のことをプリントチケットと呼ぶ。アプリケーションが、このプリントチケットを生成するためには、デバイスの能力を知る必要がある。このデバイス能力を表すドキュメント形式をプリントケーパビリティ(PrintCapabilities)と呼ぶ。プリントケーパビリティについては、特許文献1にも詳細が記されている。

10

このプリントケーパビリティは、デバイスの機能を列挙してXMLに記述するため、記述量が膨大になる傾向になる。また、プリントケーパビリティの大きな特徴として、静的なデバイスの能力を列挙して記述だけでなく、プリントチケットに応じた動的なデバイス能力を記述できることにある。具体的には、プリントチケットに記述されている用紙の種類、用紙の向きに応じた、有効印字領域といった情報をプリントケーパビリティは保持している。

20

【特許文献1】特開2006-155611号公報

【発明の開示】

【発明が解決しようとする課題】

【0004】

プリントケーパビリティの記述量は大きいと、アプリケーションなどがプリントケーパビリティを取得する際に、相応の処理時間を必要とする。アプリケーションは、ユーザーのストレスとなるような待ち時間を感じさせないように、可能な限りプリントケーパビリティの取得回数を減らし、取得したプリントケーパビリティを使いまわすことを試みるだろう。しかしながら、プリントケーパビリティには有効印字領域といった用紙の種類、用紙の向きに応じてページ毎に変更される可能性のあるデータを保持している。よって、単純に一度取得したプリントケーパビリティを使いまわすわけにはいかない。最悪、アプリケーションは、ページ毎にプリントケーパビリティを取得し直す必要がある。

30

【0005】

以上、新しいXPSプリンティングシステムでは、その土台となるプリントケーパビリティにおいて、性能劣化を引き起こす構造上の問題を抱えている。

【課題を解決するための手段】

【0006】

上述した課題を解決するために、本発明は、
プリントチケット、及びプリントケーパビリティのスコープを入力する入力手段と、
前記入力手段により入力されたプリントチケットとプリントケーパビリティのスコープとに基づきプリントケーパビリティを生成する手段と、
前記生成したプリントケーパビリティをアプリケーションに返却する返却手段とを有することを特徴とする。

40

【発明の効果】

【0007】

本発明によって、アプリケーションがプリンタドライバからデバイス能力を得る際に、必要とする情報をストレスなく入手可能になる。ひいては、ユーザーの利便性を向上させることができる。

50

【発明を実施するための最良の形態】

【0008】

[実施例1]

本発明に係る情報処理装置の一実施形態としてのコンピュータについて説明する。図3は、本実施形態のコンピュータを含むプリンタ制御システムの構成を説明するブロック図である。なお、特に断らない限り、本発明の機能が実行されるのであれば、単体の機器であっても、複数の機器からなるシステムであっても、LAN、WAN等のネットワークを介して接続が為され処理が行われるシステムであっても本発明を適用できることは言うまでもない。

【0009】

図において、300はホストコンピュータである。ホストコンピュータはROM3のプログラム用ROMあるいは外部メモリ11に記憶された文書処理プログラム等に基づいて図形、イメージ、文字、表(表計算等を含む)等が混在した文書処理を実行するCPU1を備え、システムバス4に接続される各デバイスをCPU1が総括的に制御する。

【0010】

また、このROM3のプログラム用ROMあるいは外部メモリ11には、CPU1の制御プログラムであるオペレーティングシステムプログラム(以下OS)等を記憶する。また、ROM3のフォント用ROMあるいは外部メモリ11には上記文書処理の際に使用するフォントデータ等を記憶し、ROM3のデータ用ROMあるいは外部メモリ11には上記文書処理等を行う際に使用する各種データを記憶する。

【0011】

2はRAMで、CPU1の主メモリ、ワークエリア等として機能する。5はキーボードI/Fで、キーボード9や不図示のポインティングデバイスからのキー入力を制御する。6はディスプレイI/Fで、ディスプレイ10の表示を制御する。7は外部メモリI/Fで、ブートプログラム、各種のアプリケーション、フォントデータ、ユーザファイル、編集ファイル、プリンタドライバ等を記憶するハードディスク(HD)、フロッピー(登録商標)ディスク(FD)等の外部メモリ11とのアクセスを制御する。なお、外部メモリはプリンタドライバ等の各種処理プログラムを記憶するコンピュータが読み取り可能な記憶媒体として機能する。

【0012】

8はプリンタI/Fで、所定の双方向性インターフェース21を介してプリンタ150に接続されて、プリンタ150との通信制御処理を実行する。なお、CPU1は、例えばRAM2上に設定された表示情報RAMへのアウトラインフォントの展開(ラスライズ)処理を実行し、ディスプレイ10上でのWYSIWYGを可能としている。

【0013】

また、CPU1は、CRT10上の不図示のマウスカーソル等で指示されたコマンドに基づいて登録された種々のウィンドウを開き、種々のデータ処理を実行する。ユーザは印刷を実行する際、印刷の設定に関するウィンドウを開き、プリンタの設定や、印刷モードの選択を含むプリンタドライバに対する印刷処理方法の設定を行える。

【0014】

プリンタ150において、12はプリンタCPUで、ROM13のプログラム用ROMに記憶された制御プログラム等あるいは外部メモリ14に記憶された制御プログラム等に基づいてシステムバス15に接続される印刷部(プリンタエンジン)17に出力情報としての画像信号を出力する。

【0015】

また、このROM13のプログラムROMには、CPU12の制御プログラム等を記憶する。ROM13のフォント用ROMには上記出力情報を生成する際に使用するフォントデータ等を記憶し、ROM13のデータ用ROMにはハードディスク等の外部メモリ14がないプリンタの場合には、ホストコンピュータ上で利用される情報等を記憶している。

【0016】

10

20

30

40

50

C P U 1 2 は入力部 1 8 を介してホストコンピュータとの通信処理が可能となっており、プリンタ内の情報等をホストコンピュータ 3 0 0 0 に通知可能に構成されている。1 9 は C P U 1 2 の主メモリ、ワークエリア等として機能する R A M で、図示しない増設ポートに接続されるオプション R A M によりメモリ容量を拡張することができるように構成されている。なお、R A M 1 9 は、出力情報展開領域、環境データ格納領域、N V R A M 等に用いられる。前述したハードディスク (H D)、I C カード等の外部メモリ 1 4 は、メモリコントローラ (M C) 2 0 によりアクセスを制御される。

【 0 0 1 7 】

外部メモリ 1 4 は、オプションとして接続され、フォントデータ、エミュレーションプログラム、フォームデータ等を記憶する。また、1 8 は前述した操作パネルで操作のためのスイッチおよび L E D 表示器等が配されている。また、前述した外部メモリは 1 個に限らず、少なくとも 1 個以上備え、内蔵フォントに加えてオプションフォントカード、言語系の異なるプリンタ制御言語を解釈するプログラムを格納した外部メモリを複数接続できるように構成されていてもよい。さらに、図示しない N V R A M を有し、操作パネル 2 0 からのプリンタモード設定情報を記憶するようにしてもよい。

10

【 0 0 1 8 】

図 4 は、ホストコンピュータにおける典型的な印刷データ生成方法を説明する図である。

【 0 0 1 9 】

アプリケーション 4 0 1、グラフィックエンジン 4 0 2、プリンタドライバ 4 0 3、およびシステムスプーラ 4 0 4 は、外部メモリ 1 1 に保存されたファイルとして存在し、O S やそのモジュールを利用するモジュールによって R A M 2 にロードされ実行されるプログラムモジュールである。また、アプリケーション 4 0 1 およびプリンタドライバ 4 0 3 は、外部メモリ 1 1 の F D や不図示の C D - R O M、あるいは不図示のネットワークを経由して外部ディスク 1 1 の H D に追加することが可能となっている。

20

【 0 0 2 0 】

外部メモリ 1 1 に保存されているアプリケーション 4 0 1 は R A M 2 にロードされて実行されるが、このアプリケーション 4 0 1 からプリンタ 1 5 0 0 に対して印刷を行う際には、同様に R A M 2 にロードされ実行可能となっているグラフィックエンジン 4 0 2 を利用して出力 (描画) を行う。グラフィックエンジン 4 0 2 は、印刷装置ごとに用意されたプリンタドライバ 4 0 3 を、外部メモリ 1 1 から R A M 2 にロードし、アプリケーション 2 1 0 の出力をプリンタドライバ 4 0 3 を用いてプリンタの制御コマンドに変換する。変換されたプリンタ制御コマンドは O S によって R A M 2 にロードされたシステムスプーラ 4 0 4 を経てインターフェース 2 1 経由でプリンタ 1 5 0 0 へ出力される仕組みとなっている。

30

【 0 0 2 1 】

図 5 は、アプリケーション 5 0 4 と、各プリンタ用のプリンタドライバ 5 0 1、5 0 2 との関係を示した模式図である。

【 0 0 2 2 】

アプリケーション 5 0 4 はオペレーティングシステム 5 0 3 が提供する各種関数 (A p p l i c a t i o n P r o g r a m I n t e r f a c e、通称 A P I) をコールして処理を行う。

40

【 0 0 2 3 】

オペレーティングシステム 5 0 3 は印刷に関わる処理を選択されているプリンタのプリンタドライバ 5 0 1、5 0 2 が提供する関数 (D e v i c e D r i v e r I n t e r f a c e、通称 D D I) をコールして処理を行う。

【 0 0 2 4 】

このようにしてアプリケーションから選択されたプリンタへの印刷が可能となっている。

【 0 0 2 5 】

50

このときプリンタドライバ501, 502とオペレーティングシステム503、及び、アプリケーション504間でプリンタの印刷設定データがやり取りされる。これを表したのが図6である。

【0026】

図6は、プリントチケット対応アプリケーション601や当発明のポイントであるドライバ拡張インターフェイスに対応したアプリケーション604から印刷設定する場合のプリンタドライバとの関係をあらわした模式図である。602はプリントチケット・プリントケーパビリティAPIを示している。すなわちOSの一部である。このAPIとほぼ同等の機能を持つDDIは、プリンタドライバ603にも用意されている。つまり、アプリケーション601が、PTGetPrintCapabilitiesのようなAPIを呼び出すと、API層である602を通して、ドライバ本体603に入力データであるプリントチケットが渡されることになる。ドライバ本体603は、渡されたプリントチケットに応じたプリントケーパビリティを生成し、API層である602を通して、アプリケーション601に生成したプリントケーパビリティを返す。これが従来例である。ドライバ拡張インターフェイスに対応したアプリケーション604が、図7で示すようなプリンタドライバ独自の拡張インターフェイスを呼び出すと、プリンタドライバの拡張部である605に直接入力データであるプリントチケット等が渡されることになる。プリンタドライバの拡張部である605は、渡されたプリントチケット等の情報に応じたプリントケーパビリティを生成し、直接アプリケーション604に生成したプリントケーパビリティを返す。

10

20

【0027】

図7は、プリンタドライバが公開する独自のインターフェイスの一部である。図7はC言語の関数として定義している。OSが定義するプリンタドライバインターフェイス関数PTGetPrintCapabilitiesとは名前が異なりかつ第二引数LPCWSTR pScopeが追加されている。他のところの関数仕様はほぼ同等なので、ここでは説明を省略する。なお、PTGetPrintCapabilities関数の詳細については、<http://msdn2.microsoft.com/en-us/library/ms536107.aspx>にて確認できる。

【0028】

図8は、図7の関数の処理フローを示している。ステップ801で第二引数からスコープ文字列を取得する。ステップ802で第三引数からプリントチケットを取得する。ステップ803はプリンタドライバが保持するデバイスのモデル別情報にアクセスする。一般に、803でアクセスするデータはローカルファイルであることが多い。ステップ804でpScopeがNULLだったどうかの判定をする。NULLの場合は、従来のPTGetPrintCapabilitiesと全く同じ動作する。すなわち、続くステップ805でフルセットのプリントケーパビリティを生成し、ステップ806で第四引数としてプリントケーパビリティを返却する。ステップ804でpScopeがNULL以外の場合は、ステップ807へ進む。ステップ807では、pScopeが指す文字列が“PageImageableSize”だったどうかの判定をする。“PageImageableSize”だった場合、PageImageableSizeプロパティのみ保持する小さなプリントケーパビリティを生成する。そして、ステップ806へ進む。ステップ807でpScopeが指す文字列が“PageImageableSize”以外の場合は、ステップ809にてエラー処理をする。

30

40

【0029】

図9は、図8のステップ808で生成したプリントケーパビリティの実例を示す。

【0030】

なお、本実施形態においては、新インターフェイス対応アプリ604が直接プリンタドライバのプリントチケット入出力部605を呼び出すことによって実現している。ここでは説明を省略するが、OS層で602相当のインターフェイスが用意されていて、アプリケーションはOS層を呼び出し、プリンタドライバは、そのインターフェイスを実装する

50

、といった構成であってもかまわない。

【図面の簡単な説明】

【0031】

【図1】XPSを構成する主要なパーツを使った概観図である。

【図2】XPSの構成する主要なパーツとプリントチケットの関係図である。

【図3】本発明の一実施形態としてのコンピュータを含むシステムの構成を説明するブロック図である。

【図4】本発明の一実施形態としてのコンピュータにおける典型的な印刷データ生成方法を説明するブロック図である。

【図5】本発明の一実施形態としてのコンピュータにおける、アプリケーションと各プリンタ用のプリンタドライバとの関係をあらわした図である。

【図6】本発明の一実施形態としてのコンピュータにおける、アプリ層からOS層、ドライバ層に至るプリントチケットとプリントケーパビリティの取り扱いを示す関係図である。

【図7】本発明の一実施形態としてのプリンタドライバにおける、プリンタドライバ独自のインターフェース（C言語の関数）を示す図である。

【図8】本発明の一実施形態としてのプリンタドライバにおける、ユーザーインターフェースの初期化処理のフローチャートを示す図である。

【図9】本発明の一実施形態としてのプリンタドライバにおける、プリントケーパビリティの事例を示す図である。

【符号の説明】

【0032】

150 プリンタ

300 ホストコンピュータ

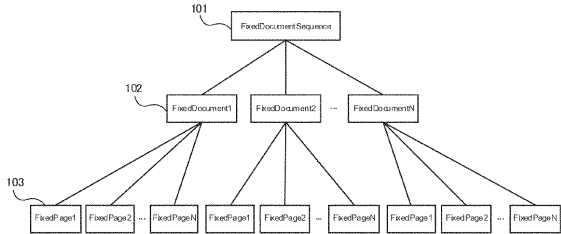
401 アプリケーション

403 プリンタドライバ

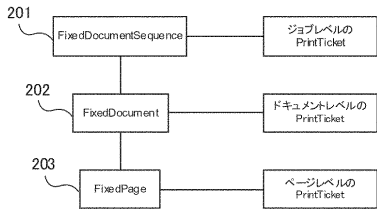
10

20

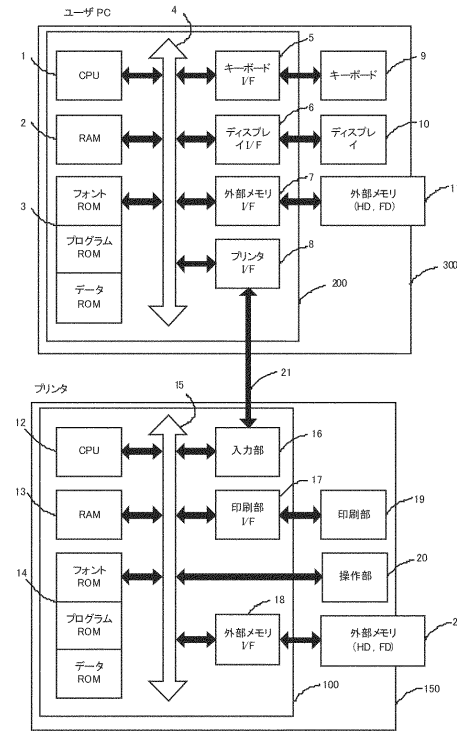
【 図 1 】



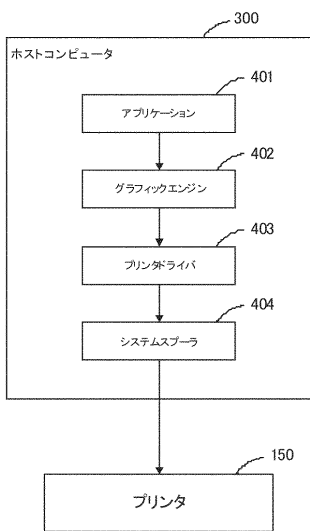
【 図 2 】



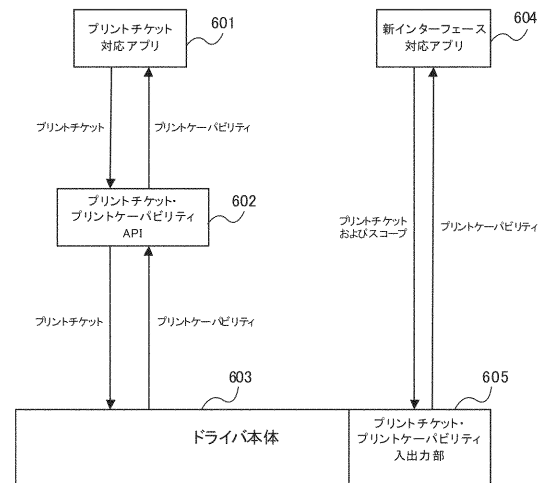
【 図 3 】



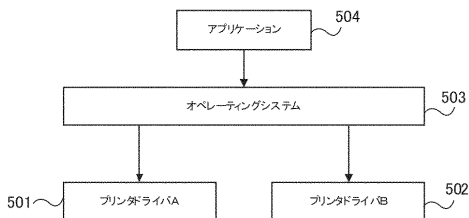
【 図 4 】



【 図 6 】



【 図 5 】



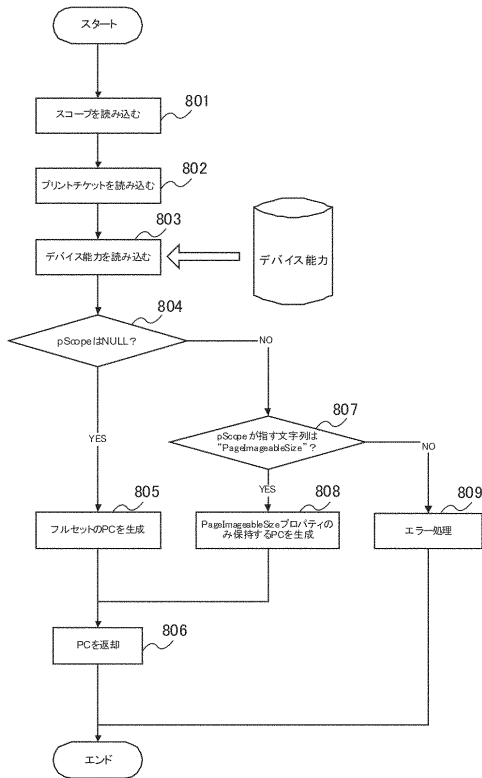
【 図 7 】

```

HRESULT CNPTGetPrintCapabilities (
    HPPProvider hProvider,
    LPCWSTR pScope,
    IStream *pPrintTicket,
    IStream *pCapabilities,
    BSTR *pbstrErrorMessage
);

```

【 図 8 】



【 図 9 】

```

<?xml version="1.0" encoding="UTF-8" ?>
<psf:PrintCapabilities version="1"
  xmlns:psf="http://schemas.microsoft.com/windows/2003/08/printing/
  print-schema-framework"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:psk="http://schemas.microsoft.com/windows/2003/08/printing/
  print-schema-keywords"
  xmlns:ns000="http://ihv">
  <psf:Property name="psk:PageImageableSize">
  <psf:Property name="psk:ImageableSizeWidth">
  <psf:Value xsi:type="xsd:integer">210000</psf:Value>
  </psf:Property>
  <psf:Property name="psk:ImageableSizeHeight">
  <psf:Value xsi:type="xsd:integer">297000</psf:Value>
  </psf:Property>
  <psf:Property name="psk:ImageableArea">
  <psf:Property name="psk:OriginWidth">
  <psf:Value xsi:type="xsd:integer">5000</psf:Value>
  </psf:Property>
  <psf:Property name="psk:OriginHeight">
  <psf:Value xsi:type="xsd:integer">5000</psf:Value>
  </psf:Property>
  <psf:Property name="psk:ExtentWidth">
  <psf:Value xsi:type="xsd:integer">200000</psf:Value>
  </psf:Property>
  <psf:Property name="psk:ExtentHeight">
  <psf:Value xsi:type="xsd:integer">287000</psf:Value>
  </psf:Property>
  </psf:Property>
  </psf:PrintCapabilities>
  
```