(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5268597号
(P5268597)
(45)発行日　平成25年8月21日(2013.8.21)　(24)登録日　平成25年5月17日(2013.5.17)

(51)Int.Cl.　F I
G06F 9/445 (2006.01)　G06F 9/06 610L

請求項の数 11 (全 15 頁)

(21)出願番号　特願2008-304595(P2008-304595)
(22)出願日　平成20年11月28日(2008.11.28)
(65)公開番号　特開2010-128921(P2010-128921A)
(43)公開日　平成22年6月10日(2010.6.10)
審査請求日　平成23年10月26日(2011.10.26)

(73)特許権者　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
(74)代理人　100126240
弁理士　阿部　琢磨
(74)代理人　100124442
弁理士　黒岩　創吾
(72)発明者　沼田　真仁
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

審査官　相崎　裕恒

最終頁に続く

(54)【発明の名称】アプリケーションプラットフォーム、情報処理装置、情報処理装置の制御方法、プログラム、及び記憶媒体

(57)【特許請求の範囲】
【請求項１】
アプリケーションを実行させることが可能な画像形成装置であって、
前記アプリケーションの動作時に使用される使用情報と、前記アプリケーションを識別するための識別情報とを前記アプリケーションを介することなく外部装置から受信する受信手段と、
複数の前記使用情報をまとめて管理することが可能な記憶領域に対し、前記受信手段により受信された前記使用情報と前記識別情報とを登録する登録手段と、
前記画像形成装置にインストールされたアプリケーションから前記使用情報が要求されたことに応じて、前記記憶領域に登録されている前記使用情報の中から、前記インストールされたアプリケーションの前記識別情報に関連付く前記使用情報を前記画像形成装置にインストールされたアプリケーションへ送信する送信手段と、を有することを特徴とする画像形成装置。
【請求項２】
前記記憶領域に登録されている前記使用情報は、通信先のサーバーのＩＰアドレス、ファイルの保存先、およびデバッグログの出力レベルの内、少なくとも１つの使用情報であることを特徴とする請求項１に記載の画像形成装置。
【請求項３】
前記受信手段は、前記アプリケーション固有のＡＰＩ（Ａｐｐｌｉｃａｔｉｏｎ　Ｐｒｏｇｒａｍ　Ｉｎｔｅｒｆａｃｅ）を介することなく、前記アプリケーションのプラット

フォームのＡＰＩを介して前記使用情報と前記識別情報とを受信することを特徴とする請求項１または２に記載の画像形成装置。
【請求項４】
前記登録手段は、前記画像形成装置にインストールされたアプリケーションがアンインストールされたことに応じて、当該アプリケーションの前記使用情報と前記識別情報とを削除することを特徴とする請求項１乃至３の何れか１項に記載の画像形成装置。
【請求項５】
前記送信手段は、前記画像形成装置にインストールされたアプリケーションの前記使用情報が前記記憶領域に登録されていない場合、当該アプリケーションに前記使用情報が存在しない旨を送信することを特徴とする請求項１乃至４の何れか１項に記載の画像形成装置。
【請求項６】
アプリケーションを実行させることが可能な画像形成装置を制御する制御方法であって、
前記アプリケーションの動作時に使用される使用情報と、前記アプリケーションを識別するための識別情報とを前記アプリケーションを介することなく外部装置から受信する受信ステップと、
複数の前記使用情報をまとめて管理することが可能な記憶領域に対し、前記受信ステップにおいて受信された前記使用情報と前記識別情報とを登録する登録ステップと、
前記画像形成装置にインストールされたアプリケーションから前記使用情報が要求されたことに応じて、前記記憶領域に登録されている前記使用情報の中から、前記インストールされたアプリケーションの前記識別情報に関連付く前記使用情報を前記画像形成装置にインストールされたアプリケーションへ送信する送信ステップとを含む制御方法。
【請求項７】
前記登録ステップにおいて前記記憶領域に登録された前記使用情報は、通信先のサーバーのＩＰアドレス、ファイルの保存先、およびデバッグログの出力レベルの内、少なくとも１つの使用情報であることを特徴とする請求項６に記載の制御方法。
【請求項８】
前記受信ステップにおいては、前記アプリケーション固有のＡＰＩ（Ａｐｐｌｉｃａｔｉｏｎ　Ｐｒｏｇｒａｍ　Ｉｎｔｅｒｆａｃｅ）を介することなく、前記アプリケーションのプラットフォームのＡＰＩを介して前記使用情報と前記識別情報とを受信することを特徴とする請求項６または７に記載の制御方法。
【請求項９】
前記登録ステップにおいては、前記画像形成装置にインストールされたアプリケーションがアンインストールされたことに応じて、当該アプリケーションの前記使用情報と前記識別情報とを削除することを特徴とする請求項６乃至８の何れか１項に記載の制御方法
【請求項１０】
前記送信ステップにおいては、前記画像形成装置にインストールされたアプリケーションの前記使用情報が前記記憶領域に登録されていない場合、当該アプリケーションに前記使用情報が存在しない旨を送信することを特徴とする請求項６乃至９の何れか１項に記載の制御方法。
【請求項１１】
請求項６乃至１０の何れか１項に記載の制御方法をコンピュータに実行させるためのプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、アプリケーションの設定情報を管理するためのアプリケーションプラットフォーム、情報処理装置、情報処理装置の制御方法、プログラム、及び記憶媒体に関するものである。

【背景技術】
【０００２】
従来、複合機（Ｍｕｌｔｉ　Ｆｕｎｃｔｉｏｎ　Ｐｒｉｎｔｅｒ）では、複合機の中に複数のアプリケーションをインストールすることができた。また、インストールされた各アプリケーションの設定情報は、各アプリケーション固有のＡＰＩ（Ａｐｐｌｉｃａｔｉｏｎ　Ｐｒｏｇｒａｍ　Ｉｎｔｅｒｆａｃｅ）を介して、複合機の中の各アプリケーションの保存領域に入力されていた。さらに、特許文献１ではアプリケーション固有のＡＰＩを介して入力された情報を、複数のアプリケーションの共通領域に保存する技術もある。
【０００３】
ところで、ＰＣ上のアプリケーションから、こうした複合機にインストールされている複数のアプリケーションに対し、一括して設定を行いたい場合がある。例えば、複合機を管理するサーバから、管理対象の全ての複合機に対して同じ設定を一括して行う場合などがある。
【特許文献１】特開２００５－６３２５９号公報
【発明の開示】
【発明が解決しようとする課題】
【０００４】
このような場合、複合機の中のアプリケーションごとに、設定情報を入力するＡＰＩが異なるために、開発者は各ＡＰＩに適合するようにＰＣ上のアプリケーションをそれぞれ開発、またはカスタマイズする必要があった。
【０００５】
本発明は、上記の課題を解決するためになされたもので、各アプリケーション固有のＡＰＩを介することなしに、各アプリケーションの設定情報を入力可能なアプリケーションプラットフォームの提供を目的とする。
【課題を解決するための手段】
【０００６】
本発明の一実施形に係る画像形成装置は、アプリケーションを実行させることが可能な画像形成装置であって、前記アプリケーションの動作時に使用される使用情報と、前記アプリケーションを識別するための識別情報とを、前記アプリケーションを介することなく外部装置から受信する受信手段と、複数の前記使用情報をまとめて管理することが可能な記憶領域に対し、前記受信手段により受信された前記使用情報と前記識別情報とを登録する登録手段と、前記画像形成装置にインストールされたアプリケーションから前記使用情報が要求されたことに応じて、前記記憶領域に登録されている前記使用情報の中から、前記インストールされたアプリケーションの前記識別情報に関連付く前記使用情報を前記画像形成装置にインストールされたアプリケーションへ送信する送信手段と、を有することを特徴とする。
【発明の効果】
【０００７】
各アプリケーション固有のＡＰＩを介することなしに、各アプリケーションの設定情報を入力することが可能となる。
【発明を実施するための最良の形態】
【０００８】
以下、本発明を実施するための最良の形態について図面を用いて説明する。
【０００９】
図１は、本発明の第１の実施形態において情報処理装置の一例である画像形成装置１のブロック図である。
【００１０】
（画像形成装置１の説明）
画像形成装置１は、印刷装置１１及び画像処理装置１２を備える。画像処理装置１２は、ＣＰＵ１２１と、第１の記憶部１２２（例、ＲＡＭ）と、第２の記憶部１２３（例、Ｒ

ＯＭやＨＤＤ）と、ユーザインターフェース１２４と、外部インターフェース１２５とを備える。

【００１１】

第１の記憶部１２２は、ＣＰＵ１２１との間でデータを直接やり取りする記憶部であり、第２の記憶部１２３は、第１の記憶部１２２を介してＣＰＵ１２１との間でデータをやり取りする記憶部である。第１の記憶部１２２には、種々のアプリケーションプログラム及びプラットフォームプログラムが記憶されている。ＣＰＵ１２１が、記憶部１２２及び１２３に記憶されている各種プログラムを実行し、制御を行うことでアプリケーションやプラットフォームの機能を実現する。

【００１２】

ユーザインターフェース１２４は、キーボード、マウス、ディスプレイ等であり、ユーザからの指示を受付け、又は、データ（画面データ）を表示する。

【００１３】

外部インターフェース１２５は、外部装置からのデータの受信や外部装置へのデータの送信を行う。例えば、外部装置には、外付けＨＤＤや外付けＵＳＢメモリ等の外付け記憶装置や、ネットワークを介して接続された他のホストコンピュータや他の画像形成装置が含まれる。

【００１４】

（第１のアプリケーション部２１の説明）

本実施形態では、第１のアプリケーション部２１は以下の構成の組み合わせから成る。

・ＣＰＵ１２１
・第１のアプリケーションプログラムが記憶されている第１の記憶部１２２内の記憶領域
・ＣＰＵ１２１が上記第１のアプリケーションプログラムを処理した際に得られる計算結果等の情報、又は、処理する上で必要な情報が記憶されている第１の記憶部１２２内又は第２の記憶部１２３内の記憶領域

尚、本実施形態では、第１のアプリケーションプログラムの一部で、第１のアプリケーション部２１が他のプログラムとの間で情報や命令を交換する部分のことを、第１のアプリケーション部２１のＡＰＩと呼ぶ。

【００１５】

また、本実施形態では、第１のアプリケーション部２１が動作するためには、インストール後にユーザによりアプリケーション情報の一部である設定情報（例．通信先のサーバのＩＰアドレスなど）が入力される必要がある。本実施形態では、このように第１のアプリケーションが動作するために使用する設定情報の中で、第１のアプリケーションをインストールした後に別途入力が必要な情報のことを、外部設定情報と呼ぶ。この外部設定情報は、第１のアプリケーション部２１の第２の記憶部１２３ではなく、後述するプラットフォーム部２０の第２の記憶部１２３に記憶する。

【００１６】

（プラットフォーム部２０の説明）

本実施形態では、複数のアプリケーションがその上で動作するアプリケーションプラットフォームであるプラットフォーム部２０は以下の構成の組み合わせから成る。

・ＣＰＵ１２１
・プラットフォームプログラムが記憶されている第１の記憶部１２２内の記憶領域
・ＣＰＵ１２１が上記プラットフォームプログラムを処理した際に得られる計算結果等の情報、又は、処理する上で必要な情報が記憶されている第１の記憶部１２２内又は第２の記憶部１２３内の記憶領域
・第１のアプリケーションプログラムが使用する外部設定情報の登録先（管理テーブル２００）を記憶する第２の記憶部１２３内の記憶領域

尚、本実施形態では、プラットフォームプログラムの一部で、プラットフォーム部２０が他のプログラムとの間で情報や命令を交換する部分のことを、プラットフォーム部２０のＡＰＩと呼ぶ。

【００１７】
プラットフォーム部２０は、ユーザインターフェース１２４、または外部インターフェース１２５を経由し、プラットフォーム部２０のＡＰＩを介して第１のアプリケーション部２１の外部設定情報を受信する。そして、プラットフォーム部２０は、第２の記憶部１２３の内、第１のアプリケーション部２１の記憶領域ではなく、プラットフォーム部２０の記憶領域に受信した外部設定情報を登録する。尚、本実施形態では、「外部設定情報を登録する」とは、「後述の管理テーブル２００の各欄に、夫々のアプリケーション情報を格納し、管理テーブル２００内の１レコードを完成させる」ことを意味する。ここで、それぞれのアプリケーション情報とは例えばアプリケーションを一意に識別するための識別情報、アプリケーションが使用する使用情報である。また、使用情報とは例えばアプリケーションが外部設定情報を一意に識別するために使用する情報、外部設定情報である。
【００１８】
さらにプラットフォーム部２０は、プラットフォーム部２０のＡＰＩを介して、第１のアプリケーション部２１から外部設定情報の取得要求を受信する。そして、受信に応じてプラットフォーム部２０は、外部情報が登録されているプラットフォーム部２０の記憶領域から第１のアプリケーション部２１の外部設定情報を取得して第１のアプリケーション部２１に連絡する。
【００１９】
図２は、プラットフォーム部２０に記憶された、アプリケーションの外部設定情報の管理テーブル２００（管理情報）を示す図である。
【００２０】
アプリケーションＩＤ２０１は、アプリケーションを一意に識別するための識別情報である。本実施形態では、第１のアプリケーションのＩＤである「ＸＸＸＸＸＸＸＸ００１」が格納されている。
【００２１】
Ｋｅｙ２０２は、アプリケーションが外部設定情報を一意に識別するために使用する使用情報である。本実施形態では、アプリケーションの通信先サーバ情報を表す「Ｔａｒｇｅｔ　Ｓｅｒｖｅｒ　Ａｄｄｒｅｓｓ」、ファイルの保存先を表す「Ｓａｖｅ　Ｆｉｌｅ　ｔｏ」、デバッグログの出力レベルを表す「Ｌｏｇ　Ｌｅｖｅｌ」が格納されている。尚、Ｋｅｙ２０２に格納可能な値は、特に制限されるものではなく、文字列以外でもよい。
【００２２】
Ｖａｌｕｅ２０３は、Ｋｅｙ２０２に対応する外部設定情報であり、アプリケーションが使用するための使用情報である。
【００２３】
尚、外部設定情報の登録方法は、上述の方法に限られない。例えば、管理テーブル２００のテーブル構造やカラムはこれに限定するものではなく、第１のアプリケーションと外部設定情報とを夫々、一意に識別できるものであればよい。
【００２４】
（外部設定情報登録時の処理）
図３は、本実施形態における外部設定情報登録時のプラットフォーム部２０の処理フローチャートである。
【００２５】
ステップＳ３０１では、プラットフォーム部２０が、アプリケーションＩＤ、Ｋｅｙ、Ｖａｌｕｅ（この例では外部設定情報）のセットと共に外部設定情報登録要求を受信する。尚、ここでの受信は、ユーザインターフェース１２４、または外部インターフェース１２５を経由し、第１のアプリケーション部２１のＡＰＩではなく、プラットフォーム部２０のＡＰＩを介して行われる。これにより、アプリケーション固有のＡＰＩを介することなく、プラットフォーム部２０が各アプリケーションの外部設定情報の登録要求を受信することができる。そのため、開発者は各アプリケーション固有のＡＰＩに適合するように

ＰＣ上のアプリケーションをそれぞれ開発、またはカスタマイズする手間を省くことが可能となる。

【００２６】

ステップＳ３０２では、プラットフォーム部２０は、受信したアプリケーションＩＤに対応した第１のアプリケーション部２１がインストールされているかどうか存在確認をする。存在確認の結果アプリケーションＩＤ２０１に対応した第１のアプリケーション部２１がインストールされていないと判断された場合は、ステップＳ３０５に移行する。ステップＳ３０５では、プラットフォーム部２０は外部設定情報の登録を行わない。さらに、プラットフォーム部２０が外部設定情報を登録できない旨のエラー通知をユーザインターフェース１２４、または外部インターフェース１２５に送信する。これにより、インストールされていないアプリケーションの外部設定情報を登録して、記憶部１２３の記録容量を無駄に浪費することを防ぐことが出来る。また、ステップＳ３０２における存在確認の結果、アプリケーションＩＤ２０１に対応した第１のアプリケーション部２１がインストールされていると判断された場合はステップＳ３０３に移行する。

【００２７】

ステップＳ３０３では、プラットフォーム部２０が、外部設定情報登録要求により指定されたＫｅｙが有効なＫｅｙであるかどうかを第１のアプリケーション部２１に確認する。有効性を確認する方法として、プラットフォーム部２０が第１のアプリケーション部２１に問合せる方法が考えられる。またその他にも、第１のアプリケーション部２１の一部として第１の記憶部１２２に記憶されたプログラムファイルを、プラットフォーム部２０が自ら確認する方法などが考えられる。ここで、プログラムファイルとは、例えば第１のアプリケーションが使用するＫｅｙの一覧が記載された定義ファイルである。ステップＳ３０３で指定されたＫｅｙが無効なＫｅｙであると判断された場合は、ステップＳ３０５に移行する。ステップＳ３０５では、プラットフォーム部２０は外部設定情報の登録を行わない。また、プラットフォーム部２０が、無効なＫｅｙとセットになっているＶａｌｕｅについては外部設定情報を登録できない旨のエラー通知をユーザインターフェース１２４、または外部インターフェース１２５に送信する。これにより、記憶部１２３の記憶容量の無駄な浪費をＫｅｙ単位で防ぐことが出来る。ステップＳ３０３で指定されたＫｅｙが有効なＫｅｙであると判断された場合は、ステップＳ３０４に移行する。

【００２８】

ステップＳ３０４では、プラットフォーム部２０が有効な外部設定情報を管理テーブル２００に登録する。

【００２９】

また、第１のアプリケーション部２１のＡＰＩを介さずに外部設定情報を登録するためには、プラットフォーム部２０は第１のアプリケーション部２１のアプリケーションＩＤとＫｅｙを知っておく必要がある。これらを知るための方法は特に規定するものではない。例えば、第１のアプリケーションをインストールする際にプラットフォーム部２０に連絡するなどの手段があることは、当業者には明白であろう。

【００３０】

（外部設定情報取得時の処理）

図４は、本実施形態において、第１のアプリケーション部２１が外部設定情報を取得する際の、プラットフォーム部２０の処理フローチャートである。

【００３１】

ステップＳ４０１では、プラットフォーム部２０が、プラットフォーム部２０のＡＰＩを介して、第１のアプリケーション部２１から少なくともアプリケーションＩＤと共に外部設定情報取得要求を受信する。

【００３２】

ステップＳ４０２では、プラットフォーム部２０は受信したアプリケーションＩＤに対応する外部設定情報が図２の管理テーブル２００に存在するかを確認する。確認の結果、対応する外部設定情報が存在しないと判定された場合は、ステップＳ４０７に移行する。

ステップＳ４０７では、第１のアプリケーション部２１に対して対応する外部設定情報が存在しない旨を示す通知を送信する。また、確認の結果、対応する外部設定情報が存在すると判定された場合は、ステップＳ４０３に移行する。

【００３３】

ステップＳ４０３では、プラットフォーム部２０が、第１のアプリケーション部２１からの外部設定情報の取得要求を受信した時に、Ｋｅｙも併せて第１のアプリケーションから受信しているかを確認する。Ｋｅｙを受信していない場合は、ステップＳ４０４に移行する。ステップＳ４０４ででは管理テーブル２００からアプリケーションＩＤに対応する全てのＫｅｙと、夫々のＫｅｙに対応する夫々のＶａｌｕｅを、第１のアプリケーション部２１に送信する。このような処理により、全てのＫｅｙと対応するＶａｌｕｅを送信することで、第１のアプリケーション部２１は全体的な外部設定情報を取得することが可能となる。

【００３４】

ステップＳ４０３で外部設定情報の取得要求時にＫｅｙも受信している場合は、ステップＳ４０５に移行する。ステップＳ４０５ではプラットフォーム部２０が、受信したＫｅｙに対応する外部設定情報が管理テーブル２００に存在するかを確認する。対応する外部設定情報が存在しない場合は、ステップＳ４０７に移行する。ステップＳ４０７では、第１のアプリケーション部２１に対して対応する外部設定情報が存在しない旨を示す通知を送信する。対応する外部設定情報が存在する場合は、ステップＳ４０６に移行する。

【００３５】

ステップＳ４０６では、プラットフォーム部２０が、アプリケーションＩＤとＫｅｙとに対応するＶａｌｕｅを第１のアプリケーションに送信する。このような処理により、Ｋｅｙに対応するＶａｌｕｅのみ送信することで、第１のアプリケーション部２１は部分的な外部設定情報を取得することが可能となる。

【００３６】

ここで、ステップＳ４０４、ステップＳ４０６において、Ｋｅｙ及びＶａｌｕｅを第１のアプリケーションに送信する場合には、管理テーブル２００でそれらの情報を保持したまま送信を行う。このような処理により、第１のアプリケーションは管理テーブルに保持されている情報を、任意のタイミングで何度でも取得することが可能となる。

【００３７】

（外部設定情報削除時の処理）

次に、管理テーブル２００から外部設定情報を削除する方法について説明する。本実施形態では、削除する方法について２つの方法を説明する。

【００３８】

（外部設定情報の第１の削除処理の説明）

図５は、第１の削除方法におけるプラットフォーム部２０の処理フローチャートである。

【００３９】

ステップＳ５０１では、プラットフォーム部２０が、アプリケーションがアンインストールされたことを検知する（アンインストール検知）。

【００４０】

ステップＳ５０２では、プラットフォーム部２０が、管理テーブル２００に、アンインストールされたアプリケーションのアプリケーションＩＤ２０１に対応する外部設定情報が存在するかを確認する。確認の結果、存在しないと判断された場合はステップＳ５０４に移行し、特に処理は行われずに終了する。

【００４１】

確認の結果、対応する外部設定情報が存在すると判断された場合、ステップＳ５０３に移行する。ステップＳ５０３では、プラットフォーム部２０がアプリケーションＩＤ２０１に対応する外部設定情報を削除する。なお、本実施形態では、「外部設定情報を削除する」とは、「管理テーブル２００の各欄から、夫々のアプリケーション情報を削除し、管

理テーブル２００内の１レコードを削除させる」ことを意味する。ここで、それぞれのアプリケーション情報とは例えばアプリケーションを一意に識別するための識別情報、アプリケーションが外部設定情報を一意に識別するために使用する使用情報、外部設定情報である。

【００４２】

この第１の削除方法によって、アプリケーションがアンインストールされた時に記憶部１２３から対応する外部設定情報を削除することが可能となり、記憶部１２３から無駄な情報を削除することが出来る。

【００４３】

（外部設定情報の第２の削除処理の説明）

次に、外部設定情報の第２の削除方法について説明する。

【００４４】

図６の管理テーブル６００は、第２の削除方法を実現するために、図２の管理テーブル２００を拡張したテーブルである。アプリケーションＩＤ６０１、Ｋｅｙ６０２、Ｖａｌｕｅ６０３については、それぞれ図２で説明したものと同様である。

【００４５】

削除候補フラグ６０４は、アプリケーションＩＤ６０１に対応するアプリケーションがアンインストールされた時にＯＮにする情報である。削除候補フラグ６０４をＯＮにする際の処理フローは図示しないが、図５の処理フローチャートにおいて、ステップＳ５０３でアンインストールされたアプリケーションの外部設定情報を削除する変わりに削除候補フラグをＯＮにする（削除候補指定）。

【００４６】

インストール回数６０５は、削除候補フラグ６０４がＯＮになってから任意のアプリケーションがインストールされた回数をカウントした情報である。

【００４７】

管理テーブル６００のテーブル構造やカラムはこれに限定するものでない。また、後述する所定回数が１回の場合はインストール回数を示すカラムも不要である。

【００４８】

図７は、第２の削除方法におけるプラットフォーム部２０の処理フローチャートである。

【００４９】

ステップＳ７０１では、プラットフォーム部２０が、アプリケーションがインストールされたことを検知する（インストール検知）。

【００５０】

ステップＳ７０２では、プラットフォーム部２０が、管理テーブル６００に、ステップＳ７０１でインストールされたアプリケーションのアプリケーションＩＤに対応する外部設定情報が存在するかを確認する。対応する外部設定情報が存在する場合はステップＳ７０３に移行し、管理テーブル６００の削除候補フラグ６０４をＯＦＦにする。

【００５１】

一方、対応する外部設定情報が存在しない場合は、ステップＳ７０４に移行する。ステップＳ７０４では、プラットフォーム部２０が、管理テーブル６００の中で削除候補フラグ６０４がＯＮになっている外部設定情報のインストール回数６０５にそれぞれ１を加算する。

【００５２】

ステップＳ７０５では、プラットフォーム部２０が、インストール回数６０５において所定回数を超えている外部設定情報が存在するかを確認する。確認の結果、所定回数を超えている外部設定情報が存在する場合はステップＳ７０６に移行し、所定回数を超えている外部設定情報を削除する。また、確認の結果、所定回数を超えている外部設定情報が存在しない場合には、ステップ７０６に移行し、特に処理は行われずに終了する。

【００５３】

この第２の削除方法は、アプリケーションが再インストールされることを考慮して、アプリケーションがアンインストールされてもすぐには外部設定情報を削除しない。ただし、アンインストールされたアプリケーションが、アンインストールされないまま、所定回数を超えて、他のアプリケーションがインストールされた場合は、アンインストールされたアプリケーションに対応する外部設定情報を削除するという特徴も持つ。これにより、再インストールの場合は、再度、外部設定情報を入力する手間を省ける一方、所定回数以内に再インストールされることなくアンインストールと判断された場合は、記憶部１２３から無駄な外部設定情報を削除することができる。

【００５４】

尚、上記の方法では、インストール回数を外部設定情報削除の条件として用いたが、これに限ることは無い。例えば、アプリケーションがアンインストールされてからの時間が所定時間を超えた場合に外部設定情報を削除するといった構成であっても構わない。

【００５５】

図８は、本発明の第２の実施形態における画像形成装置のブロック図である。

【００５６】

尚、第２のアプリケーション部２２以外の構成要素は、図１の第１の実施形態と同様であり、説明は省略する。

【００５７】

（第２のアプリケーション部２２の説明）

本実施形態では、第２のアプリケーション部２２は以下の構成の組み合わせから成る。

・ＣＰＵ１２１

・第２のアプリケーションプログラムが記憶されている第１の記憶部１２２内の記憶領域ＣＰＵ１２１が上記第２のアプリケーションプログラムを処理した際に得られる計算結果等の情報、又は、処理する上で必要な情報が記憶されている第１の記憶部１２２内又は第２の記憶部１２３内の記憶領域

・第２のアプリケーションプログラムが使用する外部設定情報を記憶する第２の記憶部１２３内の記憶領域

第２のアプリケーション部２２は、外部設定情報をプラットフォーム部２０ではなく第２のアプリケーション部２２の記憶領域で保持する点が、第１のアプリケーション部２１と異なる。そして、第２のアプリケーション部２２は、自身の記憶領域で外部設定情報を保持しているため、第１のアプリケーション部２１とは異なり、プラットフォーム部２０から外部設定情報を取得する機能を備えていない。

【００５８】

第２の実施形態では、１つの画像処理装置１２の中に、第１のアプリケーション部２１と第２のアプリケーション部２２が混在している状態での外部設定情報の登録処理について説明する。

【００５９】

（外部設定情報削除時の処理）

本実施形態では、外部設定情報を登録する方法について２つの方法を説明する。

【００６０】

（外部設定情報の第１の登録処理の説明）

図９は、第１の登録方法におけるプラットフォーム部２０の処理フローチャートである。

【００６１】

ステップＳ９０１では、プラットフォーム部２０が、アプリケーションＩＤ、Ｋｅｙ、Ｖａｌｕｅのセットと共に外部設定情報登録要求を受信する。尚、ここでの受信は、ユーザインターフェース１２４、または外部インターフェース１２５を経由し、第１のアプリケーション部２１のＡＰＩではなく、プラットフォーム部２０のＡＰＩを介して行われる。

【００６２】

ステップＳ９０２では、プラットフォーム部２０が、受信したアプリケーションＩＤに対応したアプリケーションに、プラットフォーム部２０から外部設定情報を取得する機能が備わっているかを判断する。外部設定情報を取得する機能の有無を判断する方法としては、プラットフォーム部２０がアプリケーションに問合せる方法が考えられる。また、アプリケーションの一部として第１の記憶部１２２に記憶されたプログラムファイル（例えば、アプリケーションの能力が記載された定義ファイル）を、プラットフォーム部２０が自ら確認し、判断する方法も考えられる。

【００６３】

第２のアプリケーション部２２のように、アプリケーションＩＤに対応したアプリケーションが、プラットフォーム部２０から外部設定情報を取得する機能を備えていないと判断された場合は、ステップＳ９０４に移行する。ステップＳ９０４では、プラットフォーム部２０は外部設定情報の登録を行わない。そして、プラットフォーム部２０が、外部設定情報を登録できない旨を示すエラー通知をユーザインターフェース１２４、または外部インターフェース１２５に送信する。

【００６４】

一方、第１のアプリケーション部２１のように、アプリケーションＩＤに対応したアプリケーションが、プラットフォーム部２０から外部設定情報を取得する機能を備えていると判断された場合は、ステップＳ９０３に移行する。ステップＳ９０３では、プラットフォーム部２０が、管理テーブル２００に外部設定情報を登録する。

【００６５】

これにより、１つの画像処理装置の中に外部設定情報の取得機能の有無が異なるアプリケーションが混在している場合でも、無駄な外部設定情報を登録してしまうことを防ぎ、記憶部１２３の記憶領域を浪費することを防ぐことが出来る。

【００６６】

（外部設定情報の第２の登録処理の説明）

図１０は、第２の登録方法におけるプラットフォーム部２０の処理フローチャートである。

【００６７】

外部設定情報の登録要求の受信のステップ（Ｓ１００１）や、アプリケーションの外部設定情報取得機能の有無判断のステップ（Ｓ１００２）は、図９で説明した処理ステップ（Ｓ９０１、Ｓ９０２）と同様である。また、外部設定情報を管理テーブル２００に登録するステップ（Ｓ１００３）についても、図９で説明した処理ステップ（Ｓ９０３）と同様である。

【００６８】

第２の登録方法では、第２のアプリケーション部２２のように、アプリケーションＩＤに対応したアプリケーションが、プラットフォーム部２０から外部設定情報を取得する機能を備えていない場合にステップＳ１００４に移行する。ステップ１００４では、プラットフォーム部２０が、第２のアプリケーションの第２の記憶部１２３に記憶された第２のアプリケーションの外部設定情報を、受信したＫｅｙとＶａｌｕｅで更新する。

【００６９】

これにより、１つの画像処理装置の中に外部設定情報の取得機能の有無が異なるアプリケーションが混在している場合でも、ユーザは内部の記憶部１２３の構造の違いを意識することなく、アプリケーションを介さずに外部設定情報を登録することができる。

【００７０】

尚、第２の実施形態においても、第１の実施形態で説明したアプリケーションのインストール存在確認Ｓ３０２やＫｅｙの有効性確認Ｓ３０３、第１及び第２の削除処理を行っても構わない。

【００７１】

本発明の目的は、上記実施形態で示したフローチャートの手順を実現するプログラムコードを記憶した記憶媒体から、コンピュータが、そのプログラムコードを読出し実行する

ことによっても達成される。この場合、記憶媒体から読み出されたプログラムコード自体が上述した実施形態の機能を実現することになる。そのため、このプログラムコードやプログラムコードを記憶した記憶媒体も本発明を構成することができる。

【００７２】

プログラムコードを供給するための記憶媒体としては、以下のようなコンピュータ読取り可能な記憶媒体を用いることができる。例えば、フロッピー（登録商標）ディスク、ハードディスク、光ディスク、光磁気ディスク、ＣＤ－ＲＯＭ、ＣＤ－Ｒ、磁気テープ、不揮発性のメモリカード、ＲＯＭなどである。

【図面の簡単な説明】

【００７３】

【図１】第１の実施形態の画像形成装置の構成例を示すブロック図。

【図２】外部設定情報の管理テーブルの一例を示す図。

【図３】外部設定情報登録時のプラットフォーム部の処理フローチャート。

【図４】外部設定情報取得要求時のプラットフォーム部の処理フローチャート。

【図５】外部設定情報の第１の削除方法におけるプラットフォーム部の処理フローチャート。

【図６】拡張した外部設定情報管理テーブルの一例を示す図。

【図７】外部設定情報の第２の削除方法におけるプラットフォーム部の処理フローチャート。

【図８】第２の実施形態の画像形成装置の構成例を示すブロック図。

【図９】外部設定情報の第１の登録方法におけるプラットフォーム部の処理フローチャート。

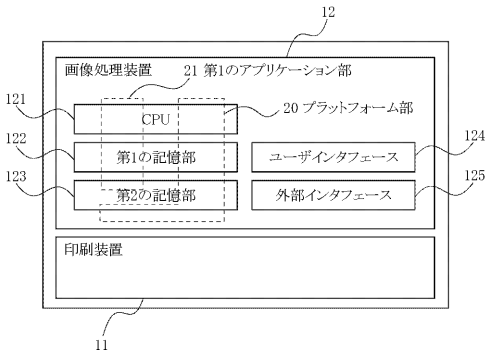
【図１０】外部設定情報の第２の登録方法におけるプラットフォーム部の処理フローチャート。

【符号の説明】

【００７４】

１　画像形成装置
２０　プラットフォーム部
２１　第１のアプリケーション部
２２　第２のアプリケーション部
１２１　ＣＰＵ
１２２　第１の記憶部
１２３　第２の記憶部
２００、６００　管理テーブル
２０１、６０１　アプリケーションＩＤ
２０２、６０２　Ｋｅｙ
２０３、６０３　Ｖａｌｕｅ

【図１】

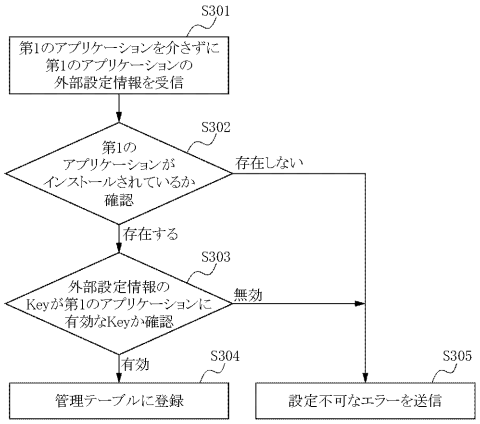


【図２】

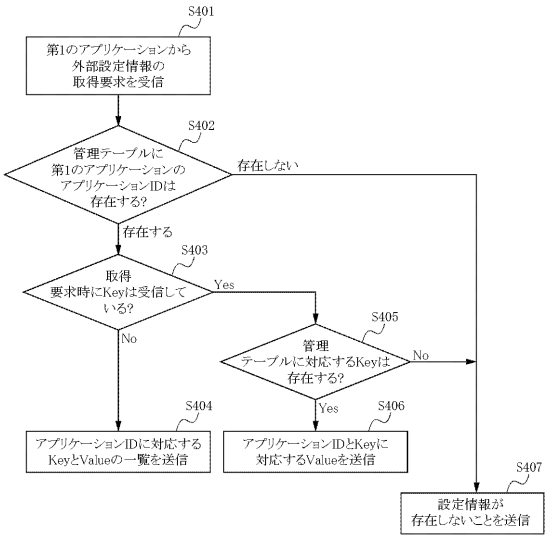
201 アプリケーションID	202 Key	203 Value
XXXXXXXX001	Target Server Address	192.168.11
XXXXXXXX001	Save File to	C:¥temp¥Appl
XXXXXXXX001	Lob Level	Debug

200

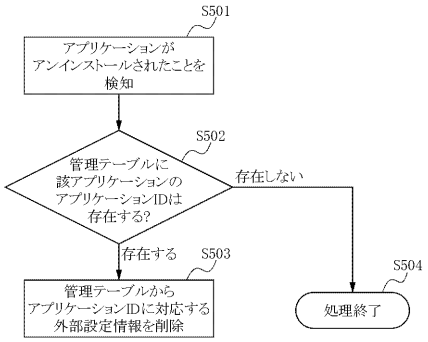
【図３】



【図４】



【図５】

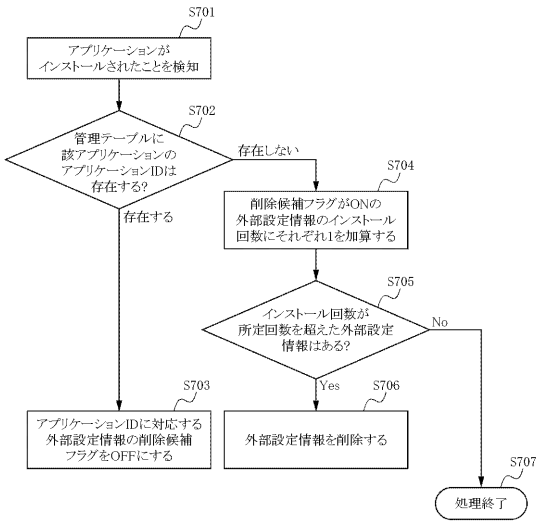


【図６】

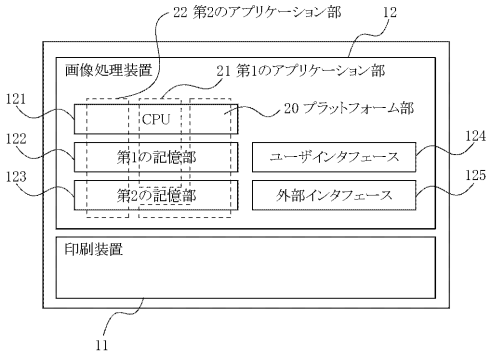
600

601 アプリケーション ID	602 Key	603 Value	604 削除候補 フラグ	605 インストール 回数
XXXXXXXX001	Target Server Address	192.168.11	ON	2
XXXXXXXX001	Save File to	C:¥temp¥Appl	ON	2
XXXXXXXX001	Lob Level	Debug	ON	2

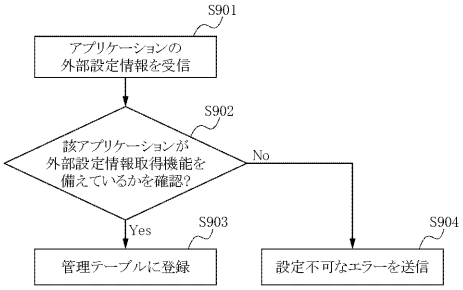
【図７】



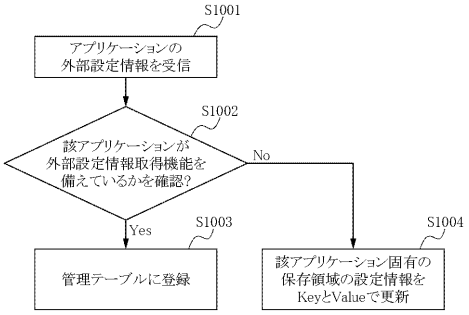
【図８】



【図９】


S901
アプリケーションの
外部設定情報を受信
S902
該アプリケーションが
外部設定情報取得機能を
備えているかを確認?
No
Yes
S903
管理テーブルに登録
S904
設定不可なエラーを送信

【図１０】


S1001
アプリケーションの
外部設定情報を受信
S1002
該アプリケーションが
外部設定情報取得機能を
備えているかを確認?
No
Yes
S1003
管理テーブルに登録
S1004
該アプリケーション固有の
保存領域の設定情報を
KeyとValueで更新

フロントページの続き

(56)参考文献 特開２００８－２６９３２３（ＪＰ，Ａ）
特開２００７－２０６８６９（ＪＰ，Ａ）
特開２０００－３５３１１９（ＪＰ，Ａ）
特開２００５－１５７６５７（ＪＰ，Ａ）
特開２００７－２２６８２８（ＪＰ，Ａ）
特開２００８－０１５９７５（ＪＰ，Ａ）
特開２００８－２７８５３５（ＪＰ，Ａ）
特開２００３－１５７１７４（ＪＰ，Ａ）
特開２００４－７４６７（ＪＰ，Ａ）
特開２００４－７８２８２（ＪＰ，Ａ）
特開２００８－２６９６３２（ＪＰ，Ａ）
特開平０９－２１８７７７（ＪＰ，Ａ）
特表２００２－５２９８５７（ＪＰ，Ａ）
特開２００３－３４５７５１（ＪＰ，Ａ）
特開２００９－０７６０６１（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｂ４１Ｊ２９／００－２９／１８
２９／２０－２９／３８
２９／４０－２９／７０
Ｇ０６Ｆ３／０９－３／１２
９／０６
９／４４－９／４４５
９／４８－９／５０
９／５４－１１／００
１１／３６
Ｈ０４Ｎ１／００