OLE for Retail POS

**Control Programmer’s Guide**

日本版仕様書 第1.7版

2002年10月24日

OLE POS技術協議会

**OLE for Retail POS**

はしがき

本仕様書は、米国OLE for Retail POS Committeeにおいて標準化された“UnifiedPOS Specification Version 1.7”を基本に作成されています。

本仕様書の著作権は、OLE　POS技術協議会にあります。本書は、あくまでも情報提供・仕様の普及を目的に供されるものであり、OLE　POS技術協議会の許諾を得ずに、本書の転用、改変、販売、他BBS等を通じての再頒布は禁じられています。また、内容は予告なく変更されることがあり、OLE　POS技術協議会は、この仕様に起因するあるいは関連するいかなる損害、リスクに関して責任を負わないこととします。

“Control Programmer’s Guide”は、次の会社の許可無しに如何なる変更、複製をすることはできません。

© 1995-2002 Fujitsu Limited. All rights reserved.  
© 1995-2002 Microsoft Corporation. All rights reserved.  
© 1995-2002 NCR Corporation. All rights reserved.  
© 1995-2002 Seiko Epson Corporation. All rights reserved.

\*Microsoft, Windows, WindowsNT, Windows2000およびWindowsXPは米国Microsoft Corporationの米国およびその他の国における登録商標です。

\*Windowsの正式名称はMicrosoft Windows Operating Systemです。

OPOSの情報については、下記のWebサイトも参照して下さい。

米国Microsoftリテイルインダストリーページ

http://www.microsoft.com/industry/retail/opos/home.asp

実装のレファレンス － コモンコントロールオブジェクト

http://monroecs.com/opos.htm

NRF-ARTS標準化団体

http:// www.nrf-arts.com

目次

[第１章 はじめに 5](#_Toc27813027)

[OLE for Retail POSコントロール インタフェース概要 6](#_Toc27813028)

[OLE for Retail POSとは？ 6](#_Toc27813029)

[マニュアル対象者 7](#_Toc27813030)

[OLE for Retail POSコントロール概要 8](#_Toc27813031)

[OPOS関連用語の説明 9](#_Toc27813032)

[インタフェース概要 12](#_Toc27813033)

[メソッド 13](#_Toc27813034)

[プロパティ 15](#_Toc27813035)

[イベント 17](#_Toc27813036)

[第２章 コントロールオブジェクトの役割とインプリメンテーション 21](#_Toc27813037)

[メソッド 21](#_Toc27813038)

[プロパティ 26](#_Toc27813039)

[イベント 28](#_Toc27813040)

[第３章 サービスオブジェクトの役割とインプリメンテーション 35](#_Toc27813041)

[メソッド 35](#_Toc27813042)

[プロパティ 44](#_Toc27813043)

[イベント 47](#_Toc27813044)

[付録A コモンコントロールオブジェクト 49](#_Toc27813045)

[特徴 49](#_Toc27813046)

[入手方法と将来 50](#_Toc27813047)

[付録B OPOSで定義する内部ヘッダーファイル 51](#_Toc27813048)

[Opos.hi : OPOS基本内部ヘッダーファイル 52](#_Toc27813049)

[OposCash.hi : ドロワー内部ヘッダーファイル 57](#_Toc27813050)

[OposTot.hi : ハードトータル内部ヘッダーファイル 59](#_Toc27813051)

[OposDisp.hi : ラインディスプレイ内部ヘッダーファイル 61](#_Toc27813052)

[OposLock.hi : キーロック内部ヘッダーファイル 65](#_Toc27813053)

[OposMsr.hi : 磁気ストライプリーダ内部ヘッダーファイル 67](#_Toc27813054)

[OposKbd.hi : POSキーボード内部ヘッダーファイル 70](#_Toc27813055)

[OposPtr.hi : POSプリンタ内部ヘッダーファイル 72](#_Toc27813056)

[OposScan.hi : スキャナ内部ヘッダーファイル 78](#_Toc27813057)

[OposCoin.hi : コインディスペンサ内部ヘッダーファイル 80](#_Toc27813058)

[OposScal.hi : 秤内部ヘッダーファイル 82](#_Toc27813059)

[OposChan.hi : 自動釣り銭機内部ヘッダーファイル 85](#_Toc27813060)

[OposTone.hi : トーンインジケータ内部ヘッダーファイル 88](#_Toc27813061)

[OposPpad.hi : PINパッド内部ヘッダーファイル 90](#_Toc27813062)

[OposCat.hi : CAT内部ヘッダーファイル 93](#_Toc27813063)

[OposPcrw.hi : ポイントカード機内部ヘッダーファイル 96](#_Toc27813064)

[OposPwr.hi : パワーマネージメント内部ヘッダファイル 100](#_Toc27813065)

[付録C 修正履歴 103](#_Toc27813066)

[第1.11版 103](#_Toc27813067)

[第1.2版 104](#_Toc27813068)

[第1.3版 105](#_Toc27813069)

[第1.4版 106](#_Toc27813070)

[第1.5版 107](#_Toc27813071)

[第1.6版 110](#_Toc27813072)

[第1.7版 111](#_Toc27813073)

# 第１章 はじめに

日本版仕様書は、米国OLE for Retail POS Committeeにおいて標準化された“UnifiedPOS Specification Version 1.7”に準拠して作成されています。

## OLE for Retail POSコントロール インタフェース概要

### OLE for Retail POSとは？

OLE for Retail POSは、POS用の周辺装置をMicrosoft Windows®95/98、Microsoft WindowsNT®、Microsoft Windows®2000およびMicrosoft Windows®XPに基づくPOSシステムに容易に統合できるオープンデバイスドライバアーキテクチャを提供するものです。

OLE for Retail POS（略して"OPOS"）の目的は次のとおりです。

1. Win32ベースPOSデバイスアクセス用アーキテクチャの定義
2. POSソリューションのサポートに十分適応できるPOSデバイスインタフェースの定義

OPOSの本リリースは次のものからなります。

1. Application Programmer’s Guide 日本版仕様書：アプリケーションディベロッパおよびハードウェアプロバイダ向け
2. Control Programmer’s Guide 日本版仕様書：ハードウェアプロバイダ向け（本マニュアル）
3. OPOSで定義するヘッダファイル
4. ソフトウェアコンポーネントはない：これらのコンポーネントはハードウェアプロバイダ、サードパーティが開発提供

### マニュアル対象者

本マニュアルの対象読者は、主としてOPOSコントロールを作成するシステム開発者を対象としています。

本マニュアルでは、読者が以下に精通していることを前提としています。

1. POS周辺装置の一般的特徴
2. ActiveXコントロール及びオートメーションの用語とアーキテクチャ
3. ActiveXコントロール コンテナ開発環境： Microsoft Visual Basic、Microsoft Visual C++ など。OPOSコントロールのテストで必要になります。
4. Application Programmer’s Guide 日本版仕様書で取り扱われているOPOSモデルとAPIの十分な知識。Application Programmer’s Guide 日本版仕様書で取り扱われている内容は、ここではほとんど繰り返されていません。

## OLE for Retail POSコントロール概要

OLE for Retail POSコントロール（OPOSコントロール）はActiveXコントロール仕様に準拠します。プロパティ、メソッド、イベントをアプリケーションに提供します。コントロールは実行時には見えず、アプリケーションだけが、メソッドおよびプロパティを通じて処理要求を行います。アプリケーションは、メソッドの戻り値とパラメータ、プロパティ、イベントを通じて、処理結果を受け取ります。

OPOSソフトウェアは、次の図に示すようなレイヤを使用してインプリメントされます。



### OPOS関連用語の説明

#### デバイスクラス

プロパティ、メソッド、イベントのセットを共有するPOSデバイスのひとつのカテゴリです。たとえば、ドロワー、POSプリンタなどです。

デバイスの中には2つ以上のデバイスクラスをサポートするものがあります。たとえば、POSプリンタでドロワー制御機能を組み込んだものがあります。また、バーコードスキャナには、秤が組み込まれているものもあります。

#### コントロールオブジェクト(CO)

デバイスクラスごとに、プロパティ、メソッド、イベントのセットをアプリケーションに提供するものです。Application Programmer’s Guide 日本版仕様書に、APIの説明が記述されています。

COは標準ActiveX（すなわち、OLE 32ビット）コントロールで、実行時には見えません。ひとつのクラスのコントロールオブジェクトのすべてのインプリメントが互換になるように、COインタフェースを設計してあります。した がって、同じクラスの別のサービスオブジェクトとは独立してコントロールオブジェクトを開発できます。

#### サービスオブジェクト(SO)

コントロールオブジェクトより呼び出されOPOSが規定する機能を、各デバイス向けに実行するものです。

SOはオートメーションサーバとしてインプリメントします。COが呼び出すメソッドのセットを提供します。COが提示した特殊メソッドを呼び出し、イベントをアプリケーションに通知することもできます。

複数のデバイスクラスをサポートするため、1つのサービスセットに複数のメソッドセットを含むこともできます。

サービスオブジェクトは、通常、インプロセスサーバとして（DLLで）インプリメントします。理論的には、アウトプロセスサーバとして（EXEで）インプリメントしてもかまいません。しかし、実際にはアウトプロセスサーバはOPOSサービスオブジェクトとしては課題があります。詳細は後述の備考を参照してください。

#### OPOSコントロールまたはコントロール

OPOSコントロールはデバイスクラスに対応して存在し、通常、デバイスクラス用のコントロールオブジェクト(CO)とサービスオブジェクト(SO)で構成されます。

COはアプリケーションインタフェースを提供し、SOはAPIを実行します。SOはCOのクラスのデバイスをサポートしなければなりません。

備考－サービスオブジェクト実装方法：アウトプロセスサーバ対インプロセスサーバ

一般的に、次の２つの状況が発生した場合に、アウトプロセスオートメションサーバを使うにあたって問題があります。

(A) サーバーがクライアントアプリケーションのためにCOM機能を処理している間（クライアントがコントロールのメソッドを呼び出している時など）に、他のサーバーがクライアントのCOM機能を呼び出した（コントロールのイベントを発行した時など）場合。

(B) サーバーがクライアントアプリケーションのCOM機能を呼び出している間（コントロールのイベントを発行している時など）に、他のクライアントアプリケーションがサーバーのCOM機能を呼び出した（クライアントがコントロールのメソッドを呼び出した時など）場合。

これら状況のうち、OPOSサービスオブジェクトでは、特に(A)が起きやすいでしょう:

1. CashDrawerの**WaitForDrawerClose**のように、いくつかのOPOSメソッドは不定期の長時間に渡る処理を必要としています。
2. **ErrorEvent**でアプリケーションのハンドラがダイアログボックスでユーザーの応答を待つように、いくつかのOPOSイベントは不定期の長時間に渡る処理を必要としています。

あるサービスオブジェクトからOPOSイベントを発生させた時に、他のサービスオブジェクトがメソッド呼び出しやプロパティへのアクセスの処理中であったりする場合は、良くあります。クライアントアプリケーションとアウトプロセスサーバが両方ともメッセージフィルタを（CoRegisterMessageFilter関数を使って）インストールしていて、それらのフィルタがOPOSに規定された方法でこれらの場面に対処していれば、この状況は処理できます。しかし、Visual BasicやVisual C++のようなクライアント環境でのデフォルトのメッセージフィルタは、この状況を適切に処理できません。その結果は（POSの操作としては不適当な）ダイアログを表示してユーザーの応答を待つとか、（POSアプリケーションにはもちろん受け入れ難い）例外を発生させてクライアントアプリケーションが終了するなどさまざまです。加えて、いくつかの環境ではアプリケーションが独自のメッセージフィルタを簡単に組み込めるような機構を用意していません。

この問題はインプロセスサーバを使う場合には存在しません。従ってサービスオブジェクトを実装するには、インプロセスサーバ方式を推奨します。しかしこの方法ではアプリケーション間でのコントロールの共有が少し複雑になります。共有のためには、共有メモリ、名前付きミューテックスやイベントのようなプロセス間通信の機構を使うことになります。

アウトプロセスサーバを使う場合には、サービスオブジェクトの開発者とアプリケーションの開発者の両方が注意深くメッセージフィルタを実装する必要があるでしょう。サービスオブジェクトのベンダは、アプリケーション開発者が行うべきことがらを適切に記述しなければなりません。

## インタフェース概要

OLE for Retail POSの主な目的は、多くのベンダーの周辺装置へ応用できる汎用周辺デバイスAPIを提供することです。その結果、コントロールオブジェクトの実装は、すべてのベンダーのサービスオブジェクトとやり取りできなければなりません。簡単な例としてプリンタが挙げられます。（ファーストフードレストランで、あるベンダー製のローカルプリンタと調理室で使う他のベンダー製リモートプリンタが必要になったとします。COの２つのインスタンスが必要となります。ここで各インスタンスは異なるSOとやり取りを行います。単独のCOが、両ベンダーのSOとうまく動作する必要があります。）

複数ベンダーのサービスオブジェクト間で動作するコントロールオブジェクトを規定するためには、コントロールオブジェクトのインタフェースはできるだけ一般的かつ簡単なものにする必要があります。結果的に、COには非常に少量の情報しか保存しません。おおまかに、次の３つの役割があります。

1. サービス オブジェクト連結： デバイスのためのサービスオブ ジェクトとのディスパッチインタフェースを管理します。
2. メソッドとプロパティ： アプリケーションのメソッド及びプロパティ要求をサービスオブジェクトへ渡します。
3. イベント： サービスオブジェクトが、コントロールオブジェクト内の特殊イベント要求メソッドの一つを呼び出すと、COは適切なイベントをアプリケーションへ発行します。

以降の節で、アプリケーション、コントロールオブジェクト、サービスオブジェクト間での数々のやり取りの経路を示します。

### メソッド

アプリケーションが、OPOSコントロールへのメソッド呼び出しを開始します。

#### Openメソッド

**Open**メソッドは、次のように処理されます。

|  |  |
| --- | --- |
| **アプリケーション** |  |
|  | 1. アプリケーションは、COの**Open**メ ソッドを呼び出します。 |
| **コントロール**  **オブジェクト** |  |
|  | 2. COは、SOの**OpenService**メソッドを呼び出します。 |
| **サービス**  **オブジェクト** |  |

#### Clolseメソッド

**Close**メソッドは、次のように処理されます。

|  |  |
| --- | --- |
| **アプリケーション** |  |
|  | 1. アプリケーションは、COの**Close**メ ソッドを呼び出します。 |
| **コントロール**  **オブジェクト** |  |
|  | 2. COは、存在していれば**CloseService**メソッドを呼び出します。それ以外なら**Close**メソッドを呼び出します、 |
| **サービス**  **オブジェクト** |  |

#### その他のメソッド

その他のすべてのメソッドは次のように処理されます。*Method*はメ ソッド名です。

|  |  |
| --- | --- |
| **アプリケーション** |  |
|  | 1. アプリケーションは、COの*Method*メソッドを呼び出します。 |
| **コントロール**  **オブジェクト** |  |
|  | 2. COは、SOの*Method*メソッドを呼び出します。 |
| **サービス**  **オブジェクト** |  |

### プロパティ

アプリケーションが、OPOSコントロールへのプロパティのアクセスを開始します。

#### 文字列型プロパティ

文字列型プロパティの取得と設定は、次のように処理されます。*StringProp*はプロパティ名です。

|  |  |
| --- | --- |
| **アプリケーション** |  |
|  | 1. アプリケーションは、COの*StringProp*プロパティをアクセスします。 |
| **コントロール**  **オブジェクト** |  |
|  | 2. 取得の場合は、COは*StringProp*を表すインデックスを使ってSOの**GetPropertyString**メソッドを呼び出します。 設定の場合は、COは*StringProp*を表すインデックスを使ってSOの**SetPropertyString**メソッドを呼び出します。  インデックスの各値は、ヘッダーファイルで宣言されています。付録を参照してください。 |
| **サービス**  **オブジェクト** |  |

#### LONGとBOOLプロパティ

longとbooleanプロパティの取得と設定は、次のように処理されます。*NumericProp*はプロパティ名です。

|  |  |
| --- | --- |
| **アプリケーション** |  |
|  | 1. アプリケーションは、COの*NumericProp*プロパティをアクセスします。 |
| **コントロール**  **オブジェクト** |  |
|  | 2. 取得の場合は、COは*NumericProp*を表すインデックスを使ってSOの**GetPropertyNumber**メソッドを呼び出します。 設定の場合は、COは*NumericProp*を表すインデックスを使ってSOの**SetPropertyNumber**メソッドを呼び出します。  インデックスの各値は、ヘッダーファイルで宣言されています。付録を参照してください。 |
| **サービス**  **オブジェクト** |  |

#### その他の型のプロパティ

その他の型のプロパティの取得と設定は、次のように処理されます。*Property*はプロパティ名です。

|  |  |
| --- | --- |
| **アプリケーション** |  |
|  | 1. アプリケーションは、COの*Property*プロパティをアクセスします。 |
| **コントロール**  **オブジェクト** |  |
|  | 2. 取得の場合は、COはSOの**Get*Property***メソッドを呼び出します。  設定の場合は、COはSOの**Set*Property***メソッドを呼び出します。 |
| **サービス**  **オブジェクト** |  |

### イベント

イベント処理の概要は、「Application Programmer’s Guide 日本版仕様書」の第１章 はじめに のイベントの項を参照してください。

アプリケーションのイベントハンドラへ発行するイベントを、サービス オブジェクトから開始します。

イベントは次のように処理されます。

|  |  |
| --- | --- |
| **アプリケーション** |  |
|  | 2. COのイベント要求メソッドは、Data,DirectIO,Error,OutputComplete, およびStatusUpdateイベントをアプリケーションへ発行します。 |
| **コントロール**  **オブジェクト** |  |
|  | 1. SOは、COのイベント要求メソッドを呼び出します。アプリケーションへ イベントを発行できるように、**SOData,SODirectIO,SOError, SOOutputComplete,** および**SOStatusUpdate**メソッドを特別に公開しています。 |
| **サービス**  **オブジェクト** |  |

#### アーキテクチャ：イベントの発行

サービスオブジェクトは、次のような理由でイベントを発行する必要が生じます。

1. メソッド呼び出し、またはプロパティ設定　アプリケーションからコントロールへ要求を出すと、イベントが発行される原因となります。

例：SOがデータを読み込み、キューイングしているものとします。アプリケーションが、**DataEventEnabled**プロパティをTRUEへ変えた場合、SOは**DataEvent**を送る必要があります。

1. 非同期アクティビティ　サービスオブジェクトは、通常デバイスの入出力を監視する目的で１つまたは複数の作業スレッドを作成します。処理時間を得るためにアプリケーションに定期的にOPOSメソッドを呼び出してもらう、または、OPOSプロパティをアクセスしてもらうといったことをせずに、SOは独立（非同期）にスケジュールされた作業スレッドを用いるのが良いでしょう。これらのスレッドが、イベントを発行しなければならないかどうかを判断します。

例：**DataEventEnabled**プロパティがTRUEで、作業スレッドはシリアルポート経由でデバイス入力を管理しているものとします。スレッドがデータメッセージを受け取った場合、SOは**DataEvent**をキューイングし、発行する必要があります。

#### SOがイベントを発行する必要がある場合、SOはCO内の特別なイベント要求メソッドを呼び出します。COは、その後にアプリケーションへイベントを発行します。

#### アーキテクチャの問題：コンテナによるイベント凍結

ActiveXコントロールコンテナは、**IOleControl::FreezeEvents**関数を呼び出してイベントを凍結させたり解凍したりする場合があります。これは**COleControl::OnFreezeEvents**関数を通してMFCで記述されたコントロールや、**IOLEControlImpl::FreezeEvents**関数を通してATLで記述されたコントロールへ通知されます。（Visual BasicのCommon Dialogの関数はこの機能を使っており、ダイアログが表示されている間はイベントを凍結しています。）

イベントが凍結されている間は、コントロールはイベントを発行すべきではありません。Visual C++資料によると、コントロールは凍結中に発生したイベントは破棄するか、または後で発行するためにバッファに保持するかのいずれかであると説明しています。

OPOSコントロールにおいては、コンテナがイベントを凍結している間はイベントを発行せずに必ずキューイングしておきます。コンテナがイベントの凍結を解除した時に、それらのイベントを発行します。

それぞれのサービスオブジェクトは、**COFreezeEvents**メソッドをサポートする必要があります。コントロールオブジェクトは、このメソッドを呼び出してイベントを凍結、または解凍します。

#### アーキテクチャの特色：アプリケーションによるイベント凍結

アプリケーションは、ある一定の時間、イベントの到着を禁止する場合、ブール型の共通プロパティ**FreezeEvents**をTRUEに設定すればできます。

コンテナ要求による凍結を考慮して実装されたイベント凍結メカニズムは、アプリケーションがイベントを凍結している間、イベント要求を記憶するのに利用されます。後にアプリケーションがプロパティをFALSEに設定してイベントの凍結を解除した時に、イベントは発行されます。

#### イベント発行の要約

イベントの発行が必要になった時、サービスオブジェクトはコントロールオブジェクト内の適切なイベント要求メソッドを呼び出します。

しかし、コントロールオブジェクトが**COFreezeEvents**関数を呼び出したり、アプリケーションが**FreezeEvents**プロパティをTRUEに設定してイベントを凍結している場合は、SOはイベントをイベントキューに保持しておく必要があります。イベントがワーカスレッドから発行される場合、通常はイベントが解凍されるまでスレッドを止めることで実現できるでしょう。

# 第２章 コントロールオブジェクトの役割とインプリメンテーション

この章ではコントロールオブジェクトの役割を説明します。コモンコントロールオブジェクトは実装のレファレンスであり、ソースコードはWebから入手できます。

## メソッド

コントロールオブジェクトのメソッドの役割を以降の章で説明します。

（「Application Programmer’s Guide 日本版仕様書」のデバイスクラスの項で説明されているように）あるデバイスクラスが共通メソッドをサポートしていない場合は、対応するコントロールオブジェクトでも定義するべきではありません。

コントロールオブジェクトは、メジャーバージョン番号が一致する限りは、どのバージョンのサービスオブジェクトとも適切に働かなければなりませんし、いくつかの機能はコントロールオブジェクトで実行する必要があります。特にコントロールオブジェクトは、サービスオブジェクトに定義されていないメソッドの呼び出しや、定義されていないプロパティへのアクセスをしてはなりません。加えて、メソッドの戻り値と**ResultCode**について状態管理を行う必要があります（「Application Programmer’s Guide 日本版仕様書」の第２章 共通プロパティ、メソッド、イベント の “**ControlObjectVersion**”の項を参照してください）。後述の処理手順では、コントロールオブジェクトはバージョンの不一致状態を管理する助けとして、ResultCodeフラグを定義しているものと想定しています。

Openメソッド

1. コントロールオブジェクトが既にオープンされている場合は、**OpenResult**にOPOS\_OR\_ALREADYOPENを設定してOPOS\_E\_ILLEGALを戻します。
2. 空のデバイス名称を渡された際は、**OpenResult**にOPOS\_OR\_REGBADNAMEを設定してOPOS\_E\_NOEXISTを戻します。
3. レジストリを調べて、このデバイスクラスとデバイス名に対応するサービスオブジェクトを探します。デバイスクラスまたはデバイス名がレジストリで見つからない場合は、**OpenResult**にOPOS\_OR \_REGBADNAMEを設定してOPOS\_E\_NOEXISTを戻します。
4. デバイス名に対応したサービスオブジェクトをロードします。これには以下の処理をおこないます (a) レジストリからデバイスのプログラマティクIDの読み取り (b) それをクラスIDに変換 (c) サービスオブジェクトのインスタンスを生成 (d) そのIDispatchインタフェースを取得。これらの内で成功しなかったら、OPOS\_E\_NOSERVICEを戻します。(a)か(b)で失敗したら**OpenResult**にOPOS\_OR\_REGPROGIDを設定し、(c)か(d)で失敗したらOPOS\_OR\_CREATEを設定します。

**MFC** (a) **RegQueryValueEx**を使用。 (b) **CLSIDFromProgID**を使用。  
(c)-(d) サービスオブジェクトクラスのインスタンスの**CreateDispatch** メンバ関数を、(b)のクラスIDを渡して、呼び出す。

サービスオブジェクトクラスは、VisualC++ウィザードを使用して生成されます：

1. “ｵｰﾄﾒｰｼｮﾝ”タブ内で、”ｸﾗｽの追加”－”ﾀｲﾌﾟﾗｲﾌﾞﾗﾘから…”のボタンを押します。次に、サービスオブジェクトプロジェクトで生成された.TLBファイル選択します。
2. クラスウィザードは、サービスオブジェクトで公開されたOLEオートメーションメソッドに対応するメンバ関数を伴った、 **COleDispatchDriver**派生クラスを生成します。

クラスウィザードは、固定されたディスパッチIDで**InvokeHelper**を呼び出すメンバ関数の実装も生成します。ディスパッチIDはオートメーションメソッドの定義順序に依存するので、この実装部位は、違う順序でメソッドを定義したサービスオブジェクトも許容するように、次のステップで更新しなければなりません。

**ATL** (a) **RegQueryValueEx**を使用。 (b) **CLSIDFromProgID**を使用。  
(c) **CoCreateInstance**を使用。 (d) (c)で返されたインタフェースポインタ上の**QueryInterface**を使用。

1. デバイスクラスで定義された全てのサービスオブジェクトメソッドのディスパッチIDを調べます。

デバイスクラスの最初のバージョンで定義されたディスパッチIDのどれか一つでもサービスオブジェクトにない場合は、ディスパッチインタフェースをクローズして、**OpenResult**にOPOS\_OR\_BADIFを設定し、OPOS\_E\_NOSERVICEを戻します。（これは、どれかメソッドを呼び出す前に、サービスオブジェクトがそのデバイスクラスにとって有効な最小限のメソッドをサポートしているかの確認です）

**MFC** サービスオブジェクトインスタンスの**m\_lpDispatch → GetIDsOfNames**関数を呼び出してディスパッチIDを調べます。**InvokeHelper**メンバ関数にこれらのディスパッチIDを渡すように、生成されたサービスオブジェクトメソッドを更新します。

**ATL** サービスオブジェクトインスタンスの**GetIDsOfNames**関数を呼び出してディスパッチIDを調べます。後で使用するためにそれらを保存します－それらはサービスオブジェクトのディスパッチ**Invoke**関数に渡さなければなりません。

1. デバイスクラス文字列、デバイス名文字列、コントロールオブジェクトへの**IDispatch**ポインタを引数にして、サービスオブ ジェクトの**OpenService**メソッドを呼び出します。**OpenService**の戻り値がOPOS\_SUCCESS以外の場合は、ディスパッチインタフェースをクローズして、**OpenService**の戻り値を戻します。サービスオブジェクトが**GetOpenResult**メソッドをサポートするなら、それを呼び出して**OpenResult**にその戻り値を設定し、サポートしないならOPOS\_OR\_FAILEDOPENを**OpenResult**に設定します。

**MFC** コントロールオブジェクトのディスパッチポインタは、**GetIDispatch(FALSE)**メンバ関数により取得されます。

**ATL** コントロールオブジェクトのディスパッチポインタは、**IDispatch**インタフェースを要求して**QueryInterface**関数を呼び出すことにより取得されます。

1. **GetPropertyNumber(PIDX\_ServiceObjectVersion)**メソッドを呼び出し、サービスオブジェクトのバージョン番号を取得します。メジャーバージョンが1でなければ、**OpenResult**にOPOS\_OR\_BADVERSIONを設定してOPOS\_E\_NOSERVICEを戻します。
2. サービスオブジェクトの対応するバージョンで定義されているべきメソッドのディスパッチIDのどれか一つでも見つからない場合：  
   存在するならサービスオブジェクトの**CloseService**メソッドを呼び出し、それ以外なら**Close**メソッドを呼び出し、  
   ディスパッチインタフェースをクローズし、  
   **OpenResult**にOPOS\_OR\_BADIFを設定し、  
   OPOS\_E\_NOSERVICEを戻します。  
   （これは、サービスオブジェクトがそのデバイスクラスと対応するバージョンでサポートされているべきすべてのメソッドをサポートしているかの確認です。サービスオブジェクトのバージョンがコントロールオブジェクトより新しいなら、コントロールオブジェクトはそのコントロールオブジェクトのバージョンでサポートしているすべてのメソッドを確認します。）
3. 上記の全ステップが正常終了した場合は、コントロールオブジェクトがオープンしていることを示す内部変数を設定し、**OpenResult**にOPOS\_SUCCESSを設定し、OPOS\_SUCCESSを戻します。その他の場合は、コントロールオブジェクトはクローズのままです。

Closeメソッド

1. コントロールオブジェクトがクローズしている場合は、OPOS\_E\_CLOSEDを戻します。
2. サービスオブジェクトが**CloseService**メソッドをサポートするなら、それを呼び出します。それ以外なら**Close**メソッドを呼び出します。
3. コントロールオブジェクトがクローズしていることを示す内部変数を設定します。
4. サービスオブジェクトをリリースします。
5. **MFC** サービスオブジェクトクラスの**ReleaseDispatch**メンバ関数を呼び出します。
6. **ATL** サービスオブジェクトのディスパッチポインタの**Release**メンバ関数を呼び出します。
7. サービスオブジェクトの**Close**メソッドの結果を戻します。

その他のメソッド呼び出し

1. コントロールオブジェクトがクローズしている場合は、OPOS\_E\_CLOSEDを戻します。
2. メソッドがサービスオブジェクトのデバイスクラスの対応するバージョンで定義されていない場合は:
3. ResultCodeフラグに“バージョン違反状態”を設定します。
4. OPOS\_E\_NOSERVICEを戻します。
5. メソッドがサービスオブジェクトで定義されている場合は:
6. サービスオブジェクトの同じ名前のメソッドを、同一の引数を使って呼び出し、サービスオブジェクトへの要求を渡します。
7. ResultCodeフラグに“ノーマル状態”を設定します。
8. サービスオブジェクトのメソッドの結果を戻します。

## プロパティ

コントロールオブジェクトは、プロパティへのアクセスを次のように処理します。

1. コントロールオブジェクトは、**ControlObjectDescription**と**ControlObjectVersion**と**OpenResult**プロパティだけを保持しています。コントロールオブジェクトは、これらのプロパティへのアクセスを直接処理し、それらの値を戻します。

2. コントロールオブジェクトがクローズしている場合は、

1. プロパティの設定であれば戻ります。（設定が失敗したことをアプリケーションへ通知する手段はありません。）
2. プロパティの取得であれば、
3. プロパティが**State**の場合はOPOS\_S\_CLOSEDを戻します。
4. プロパティが**ResultCode**の場合はOPOS\_E\_CLOSEDを戻します。
5. その他の場合は、次のプロパティのデフォルト値を戻します。  
    ブール型ならFALSE

数値型ならゼロ

文字列型なら"[Error]"

1. **ResultCode**プロパティの取得で、ResultCodeフラグが“バージョン違反状態”の場合は、OPOS\_E\_NOSERVICEを戻します。
2. プロパティがサービスオブジェクトの対応するバージョンでサポートされていない場合は:
3. プロパティの設定であれば、ResultCodeフラグに“バージョン違反状態”を設定して、戻ります。
4. プロパティの取得であれば、プロパティのデフォルト値を戻します。

***上記以外の場合は...***

1. ResultCodeフラグに“ノーマル状態”を設定します。
2. 要求をサービスオブジェクトまで渡します。
3. プロパティが４バイト数値型（booleanとlongを含む）の場合は、サービスオブジェクトの**GetPropertyNumber**または**SetPropertyNumber**を呼び出します。引数にプロパティのインデックスを指定します。これらのインデックスは、OPOS内部ヘッダーファイルで宣言されています。新デバイスにコントロールオブジェクトを供給するには新デバイスの章を作成した者がデバイスの主要な特徴ジェネレーターに定義づけるために使用する約２ページのデータファイルを用意することが求められる｡（「付録A」を参照してください。）
4. プロパティが文字列型の場合は、サービスオブジェクトの**GetPropertyString**または**SetPropertyString**を呼び出します。引数にプロパティのインデックスを指定します。これらのインデックスは、OPOS内部ヘッダーファイルで宣言されています。（「付録B」を参照してください。）
5. プロパティがその他の型の場合は、サービスオブジェクトの取得または設定メソッドを呼び出します。

## イベント

サービスオブジェクトはイベントを開始します。SOは、コントロールオブジェクトによって公開されているイベント要求メソッドを呼び出します。

サービスオブジェクトによって呼び出されるイベント要求メソッドとOPOSイベントの対応表です。

|  |  |
| --- | --- |
| イベント要求メソッド | OPOSイベント |
| SOData | DataEvent |
| SODirectIO | DirectIOEvent |
| SOError | ErrorEvent |
| SOOutputComplete | OutputCompleteEvent |
| SOStatusUpdate | StatusUpdateEvent |

これらのイベント要求メソッドの一つを受け取ると、コントロールオブジェクトはアプリケーションへ適切なイベントを発行します。サービスオブジェクトスレッドは、アプリケーションのイベントハンドラが終了するまで制御を獲得できません。

***備考：*** これらのメソッドは、サービスオブジェクトによる使用のみに限られます。アプリケーションからアクセス可能ですが、アプリケーションからは呼び出すべきではありません。

これら５つのイベント要求メソッドは、以降のページで定義されています。

SOData

形式 **void SOData (LONG** *Status***);**

*Status*パラメータには入力ステータスが入ります。この値はコントロールに応じて異なり、入力のタイプや特性を示します。

説明 コントロールオブジェクトにイベントを発行するように要求します。

**void DataEvent(LONG** *Status***);**

デバイスからアプリケーションへ、データの入力が行われたことを通知するためにサービスオブジェクトから呼び出されます。**DataEventEnabled**プロパティがTRUEでなければ、SOは決して**SOData**を呼び出してはいけません。**SOData**を呼び出す直前に、SOはこのプロパティをFALSEに変える必要があります。そうすることで、アプリケーションがこのプロパティを再びTRUEへ戻すまで、この間に起こるデータイベントの生成を防げます。実際の入力データは、デバイスクラス特有の一つあるいは複数のプロパティに保存されます。

SODirectIO

形式 **void SODirectIO (LONG** *EventNumber*, **LONG\*** *pData*,

**BSTR\*** *pString***);**

パラメータ 説明

*EventNumber* イベント番号。サービスオブジェクトが割り当てた特定値です。

*pData* 数値データのポインタ。値はイベント番号とサービスオブジェクトによって変化します。

*pString* 文字列データのポインタ。値はイベント番号とサービスオブジェクトによって変化します。

説明 コントロールオブジェクトにイベントを発行するように要求します。

**void DirectIOEvent(LONG** *EventNumber***, LONG\*** *pData***,**

**BSTR\*** *pString***);**

直接アプリケーションへ情報を伝えるために、サービスオブジェクトから呼び出されます。

このイベントは、コントロールオブジェクトではサポートしていないイベントを、サービスオブジェクトがアプリケーションに発行する手段を提供します。

サービスオブジェクトは、*pString*が適切なシステム文字列をポイントしていることを保証する必要があります。更に、**SODirectIO**が戻り次第この文字列を開放しなければなりません。

SOError

形式 **void SOError (LONG** *ResultCode*, **LONG** *ResultCodeExtended*,  
**LONG** *ErrorLocus***, LONG\*** *pErrorResponse***);**

パラメータ 説明

*ResultCode* エラーイベントが生じた原因を示すコードです。値については「Application Programmer’s Guide 日本版仕様書」の**ResultCode**の項を参照してください。

*ResultCodeExtended* エラーイベントが生じた原因を示す拡張 コードです。値については「Application Programmer’s Guide 日本版仕様書」の**ResultCodeExtended**の項を参照してください。

*ErrorLocus* エラーの位置です。下記の値を参照してください。

*pErrorResponse* エラーイベント応答のポインタです。下記の値を参照してください。

*ErrorLocus*パラメータの値は次のいずれかです。

値 意味

OPOS\_EL\_OUTPUT 非同期出力を処理しているときにエラーが発生しました。

OPOS\_EL\_INPUT イベント駆動入力で、入力データ処理をしているときにエラーが発生しました。入力データは利用できません。

OPOS\_EL\_INPUT\_DATA イベント駆動入力で、入力データ処理をしているときにエラーが発生しました。すでにバッファリングされたデータは利用できます。

*pErrorResponse*パラメータが指す位置にある内容は、*ErrorLocus*に基づいてデフォルト値があらかじめ設定されています。アプリケーションはその値を以下のいずれかに変更できます。

値 意味

OPOS\_ER\_RETRY 通常は*ErrorLocus*がOPOS\_EL\_OUTPUTの場合のみ有効です。非同期出力を再試行します。エラー状態は解除されます。  
*ErrorLocus*がOPOS\_EL\_INPUTの時でも指定可能な場合があります。  
*ErrorLocus*がOPOS\_EL\_OUTPUTの場合のデフォルトです。

OPOS\_ER\_CLEAR 非同期出力またはバッファリングされている入力データをクリアします。エラー状態は解除されます。*ErrorLocus*がOPOS\_EL\_INPUTの場合のデフォルトです。

OPOS\_ER\_CONTINUEINPUT  
*ErrorLocus*がOPOS\_EL\_INPUT\_DATAの場合だけ使用します。エラーを容認し、コントロールに処理の継続を指示するものです。コントロールはエラー状態のままですが、**DataEventEnable**プロパティの設定値に 従って、別な**DataEvent**が通知されます。  
すべての入力がイベント処理された後、**DataEventEnable**プロパティが、再びTRUEに設定されたとき、**ErrorEvent**でOPOS\_EL\_INPUTが通知されます。  
*ErrorLocus*がOPOS\_EL\_INPUT\_DATAの場合のデフォルトです。

説明 コントロールオブジェクトにイベントを発行するように要求します。

**void ErrorEvent (LONG** *ResultCode*, **LONG** *ResultCodeExtended*,  
 **LONG** *ErrorLocus***, LONG\*** *pErrorResponse***);**

一度**SOError**を呼び出すと、サービスオブジェクトはエラーが解除されるまで他のエラーイベントを要求してはいけません。しかしながら、*ErrorLocus*がOPOS\_EL\_INPUT\_DATAでエラーが送られてイベントハンドラがOPOS\_ER\_CONTINUEINPUTで応答する場合は、キューイングされた入力が引き渡しされた後に、SOはOPOS\_EL\_INPUTを使って他のエラーイベントを発行できます。

SOOutputComplete

形式 **void SOOutputComplete (LONG** *OutputID***);**

*OutputID*パラメータは、完了した非同期出力リクエストのID番号を示します。

説明 コントロールオブジェクトにイベントを発行するように要求します。

**void OutputCompleteEvent (LONG** *OutputID***);**

前に開始されている非同期出力要求が正常に終了した時、サービスオブジェクトから呼び出されます。

SOStatusUpdate

形式 **void SOStatusUpdate(LONG** *Data***);**

*Data*パラメータはデバイスクラス固有データ用で、ステータス変化のタイプを示します。

説明 コントロールオブジェクトにイベントを発行するように要求します。

**void StatusUpdateEvent (LONG** *Data***);**

SOが、アプリケーションへデバイスステータス変化の警報を出す必要がある場合に、サービスオブジェクトから呼び出されます。

たとえば、ドロワー位置の変化（開閉）、POSプリンタセンサの変化（用紙の有無）などです。

*下記メソッドはイベント発行とは関係がなく、特別な用途のためにサポートするメソッドです。*

SOProcessID

形式 **LONG SOProcessID();**

説明 コントロールオブジェクトはアプリケーションのプロセスIDを戻します。

このメソッドはローカルアウトプロセスサーバのサービスオブジェクトをサポートするために用意されています。第１章はじめに の備考に、OPOSサービスオブジェクトとしてはアウトプロセスサーバには課題があると書きました。しかし、もしベンダがアウトプロセスサービスオブジェクトの設計・実装に成功したなら、このメソッドは有用でしょう。

たとえば、MICR付きプリンタのサービスオブジェクトで、アプリケーションがプリンタを**Claim**した場合に、MICRを**Claim**するのは同じアプリケーションに制限したい場合などです。なぜなら、このように密接に関連するクラスを持つデバイスを２つのアプリケーションで分け合うのは適切ではないからです。

# 第３章 サービスオブジェクトの役割とインプリメンテーション

## メソッド

次に示すサービスオブジェクトの共通メソッドは、対応するコントロールオブジェクトのメソッドの実装のために定義されています。（「Application Programmer’s Guide 日本版仕様書」のデバイスクラスの項で説明されている）あるデバイスクラスが共通メソッドをサポートしていない場合は、対応するサービスオブジェクトでも定義するべきではありません。

それぞれのデバイスクラスに対して、デバイス特有の各メソッド用に追加的なメソッドが定義されます。

サービスオブジェクトのメソッドを定義する上での全般的な規則は次のようなものです。

1. サービスオブジェクトのメソッド名は、コントロールオブジェクトのメソッド名と同じものとします。
2. パラメータは、順序および型ともにコントロールオブジェクトのパラメータと同じです。

上の規則の例外は、**OpenService**、**CloseService**(オプション－代わりに**Close**を使用可能)、**GetOpenResult**(オプション)と**COFreezeEvents**メソッドです。

これらのメソッドはサービスオブジェクトのIDispatchインタフェース経由で常に呼び出されることを注意して下さい。

下記の各メソッドのシンタクスは、MFCではコントロールの”ﾒｿｯﾄﾞの追加”ダイアログでの入力、ATLではCOMオブジェクトの”ｲﾝﾀﾌｪｰｽへﾒｿｯﾄﾞを追加”ダイアログでの入力で表示されます。

CheckHealth

形式 **MFC long CheckHealth(long** *Level***);**

**ATL HRESULT CheckHealth(long** *Level***, [out, retval] long\*** *pRC***);**

説明 デバイスの状態をテストするときに呼び出します。

ClaimDevice / Claim

形式 **MFC long ClaimDevice(long** *Timeout***);  
 long Claim(long** *Timeout***);**

**ATL HRESULT ClaimDevice(long** *Timeout***, [out, retval] long\*** *pRC***);  
 HRESULT Claim(long** *Timeout***, [out, retval] long\*** *pRC***);**

説明 デバイスに対して排他アクセスを要求するときに、このメソッドを呼び出します。

第1.0－1.4版

これらの版のコントロールオブジェクトは**Claim**メソッドのみを探します。

第1.5版以降

これらの版のコントロールオブジェクトは最初に**ClaimDevice**メソッドを探します。もし**ClaimDevice**が無ければ、コントロールオブジェクトは**Claim**を調べます。

ClearInput

形式 **MFC long ClearInput();**

**ATL HRESULT ClearInput([out, retval] long\*** *pRC***);**

説明 キューイングされているデバイス入力をすべてクリアするときに呼び出します。

ClearOutput 第1.7版で更新

形式 **MFC long ClearOutput();**

**ATL HRESULT ClearOutput([out, retval] long\*** *pRC***);**

説明 非同期出力を全て含むバッファリングされている全てのデバイス出力をすべてクリアするときに呼び出します。また、可能ならば処理中の出力も停止します。

Close

形式 **MFC long CloseService();  
 long Close();**

**ATL HRESULT CloseService([out, retval] long\*** *pRC***);   
 HRESULT Close([out, retval] long\*** *pRC***);**

説明 デバイスとそのリソースを解放するときに呼び出します。

第1.0－1.4版

これらの版のコントロールオブジェクトは**Close**メソッドのみを探します。

第1.5版以降

これらの版のコントロールオブジェクトは最初に**CloseService**メソッドを探します。もし**CloseService**が無ければ、コントロールオブジェクトは**Close**を探します。

COFreezeEvents 内部コントロール/サービスオブジェクトメソッド  
 第1.7版で更新

形式 **MFC long COFreezeEvents(BOOL Freeze);**

**ATL HRESULT COFreezeEvents(VARIANT\_BOOL Freeze,  
 [out, retval] long\* pRC);**

*Freeze*パラメータは、イベントの発行を凍結する必要がある時にはTRUE/ VARIANT\_TRUE、イベントの発行を再有効化する時にはFALSE/ VARIANT\_FALSEです。

説明 このメソッドは、コントロールオブジェクトによる内部使用のためのものです。

COは、SOへイベント発行状態における変化を通知するためのコンテナイベント凍結要求への返答として、このメソッドを呼び出します。詳しくは"アーキテクチャの問題：コンテナによるイベント凍結を参照してください。

DirectIO

形式 **MFC long DirectIO(long** *Command***, long\*** *pData***, BSTR\*** *pString***);**

**ATL HRESULT DirectIO(long** *Command***, [in, out] long\*** *pData***,  
 [in, out] BSTR\*** *pString***, [out, retval] long\*** *pRC***);**

説明 サービスオブジェクトと直接やり取りするときに呼び出します。

GetOpenResult 内部コントロール/サービスオブジェクトメソッド  
 第1.5版で追加

形式 **MFC long GetOpenResult();**

**ATL HRESULT GetOpenResult([out, retval] long\*** *pRC***);**

説明 このメソッドはコントロールオブジェクトでのみ使用されます。  
これはオプション機能です。

もしサービスオブジェクトの**OpenService**メソッドがOPOS\_SUCCESS 以外のステータスを返し、さらにこのメソッドが実装されているなら、コントロールオブジェクトは**OpenResult**プロパティを設定するためにこのメソッドを呼び出します。

サービスオブジェクトはOPOS.Hヘッダファイルで提供される値の内の1つを選択するか、サービスオブジェクト固有の値を返します。

戻り値 MFC実装では、次の値の内の1つが返されます。ATL実装では、次の値の内の1つが*pRC*に格納され、S\_OKを戻します。

値 意味

OPOS\_ORS\_NOPORT サービスオブジェクトは、**Open**処理の間にI/Oポート（例えば、RS232Cポート）にアクセスしようとしましたが、**DeviceName**に対応するポートが無効もしくはアクセス不可です。

一般的な規則として、SOは、**DeviceEnabled**プロパティがTRUEに設定されるまで、物理デバイスへのアクセスを差し控えるべきです。しかし、なんらかの場合、オープンでアクセスが必要になるかもしれません。例えば、デバイスタイプを動的に決定して**DeviceName**と**DeviceDescription**プロパティを設定するためなどです。

OPOS\_ORS\_NOTSUPPORTED  
サービスオブジェクトは指定されたデバイスをサポートしていません。

SOは、オープンしている時に、デバイスを制御できるかどうか判定します。この判定は、デバイスに対するレジストリ内容の点検や、オープン処理での動的な問い合わせなどによりおこなわれる場合があります。

OPOS\_ORS\_CONFIG 構成情報エラー。

通常、これはレジストリの不完全な設定で発生します。そのため、デバイスをオープンするための十分な、もしくは有効なデータをSOが得ることができなかったからです。

OPOS\_ORS\_SPECIFIC この値より大きなエラー値はサービスオブジェクト固有のエラーです

上記のリターン値を適用できないなら、SOはその他の**OpenResult**値を定義します。これらの値はサービスオブジェクト固有であり、以下のケースで有用です。

1) アプリケーションが、デバッグまたはテストする際に、これらのエラーをログ、もしくはレポートする。

2) エラー条件をレポートしたり、エラーの回復を行うために、アプリケーションがSO固有のロジックを追加する。

OpenService 内部コントロール/サービスオブジェクトメソッド

形式 **MFC long OpenService(LPCTSTR** *DeviceClass***, LPCTSTR** *DeviceName***,  
 LPDISPATCH** *pDispatch***);**

**ATL HRESULT OpenService(BSTR** *DeviceClass***, BSTR** *DeviceName***,  
 IDispatch\*** *pDispatch***, [out, retval] long\*** *pRC***);**

パラメータ 説明

*DeviceClass* 要求されたデバイスクラスであり、6ページのOPOS.HIヘッダファイルで提供されます。例としては、「CashDrawer」と「POSPrinter」。

*DeviceName* このサービスオブジェクトで取り扱われるデバイス名です。デバイス名と物理デバイス（複数）の関係は、オペレーティングシステムのレジストリの登録項目により決定されます。セットアップまたは設定ユーティリティは、これらの登録項目を管理します。（「Application Programmer’s Guide 日本版仕様書」の付録「OPOSレジストリの利用法」を参照してください。）

*pDispatch* イベント要求メソッドが入っているコントロールオブジェクトのディスパッチインタフェースを指すポインタです。

説明 引き続くI/O処理のため、デバイスをオープンするために呼び出します。コントロールオブジェクトは、**Open**メソッド処理の一部としてこのメソッドを呼び出します。

サービスオブジェクトは、付属するデバイス特有情報を取得するためにレジストリを調べる際に、*DeviceClass*と*DeviceName*パラメータを使用します。

*以下に示す手順は、サービスオブジェクトをVisual C++とMFCを使って作成する場合のものです。他の開発環境を選択した場合は、それに合わせて変更してください。*

MFCの実装では、*pDispatch*パラメータは次のように使用されるでしょう。

1. コントロールオブジェクトのイベント要求メソッドを定義しているクラスインスタンスの**AttachDispatch**メンバー関数へ*pDispatch*の**IDispatch**ポインタを渡して、コントロールオブジェクトの**IDispatch**インタフェースに結び付けます。

このクラスは、Visual C++のClassWizardを使用して生成されます。

1. ［OLEオートメーション］タブ内で、［クラスを追加...］ボタンを押して［OLE TypeLibから...］を選択します。そして、コントロールオブジェクトのプロジェクトで生成された.TLBファイルを選択します。
2. ClassWizardは、コントロールオブジェクトにより公開されたOLEオートメーションのメソッドと一致するメンバー関数と共に、**COleDispatchDriver**派生クラスを生成します。
3. ClassWizardは、更に固定ディスパッチIDを使って**InvokeHelper**を呼び出すメンバー関数の実装を生成します。ディスパッチIDは、オートメーションメソッドの定義順序により左右されるので、この実装は、サービスオブジェクトがメソッドを異なる順に定義できるように次のステップで更新される必要があります。
4. すべての非イベント要求メソッドを除去するために、クラス定義と実装は更新されるべきです。
5. コントロールオブジェクトインスタンスの**m\_lpDispatch → GetIDsOfNames**関数を呼び出して、（**SOData**のような）イベント要求メソッドを調べます。これらのディスパッチIDを**InvokeHelper**メンバー関数へ渡すため、生成されたコントロールオブジェクトのメソッドを更新します。

ATLの実装では、*pDispatch*パラメータはIDispatchの**GetIDsOfNames**と **Invoke**関数を呼び出すために直接使用します。あるいは、**CComDispatchDriver**クラスのインスタンスを生成します。その**Invoke1**と**InvokeN**関数はイベント関数の呼び出しに使用されます。

備考：

CO内でイベント要求メソッドをアクセスするために、サービスオブ ジェクトは逆にコントロールのディスパッチポインタへ結び付きます。このことは次の２点を意味します。

1. コントロールオブジェクトが、サービスオブジェクトからアクセスできるようにイベント要求メソッドを公開する際には、これらのメソッドはアプリケーションからもアクセスできるようになります。しかしアプリケーションは、これらのメソッドを呼び出してはいけません。
2. サービスオブジェクトは、コントロールオブジェクト内の他のメ ソッド及びプロパティをアクセスすることが可能です。このことは、通常有益ではありませんが、SOはそれ自身とCOの間での互換性を確認する目的で**ControlObjectDescription** または**ControlObjectVersion** をアクセスする場合があります。

戻り値 MFCの実装では、以下の値の1つを返します。ATLの実装では、以下の値の1つを*pRC*に格納してS\_OKを返します。

値 意味

OPOS\_SUCCESS サービスオブジェクトはオープンされました。

これは、デバイスがオンラインかつ動作していることをコントロールオブジェクトまたはアプリケーションに示すものでは**ありません**。むしろ、サービスオブジェクトソフトウェアが初期化され、**DeviceEnabled**プロパティがTRUEに設定された時にデバイスとの交信を実行する準備ができたことを示すものです。

*Other Values* APGの**ResultCode**の値を参照。

OPOS\_SUCCESS以外の戻り値ならオープンは失敗しており、コントロールオブジェクトは（COMポインタを開放して）サービスオブジェクトを切り離し、アプリケーションにそのステータスを返すことになります。

APGでOPOS\_E\_ILLEGALとOPOS\_E\_NOEXISTが定義されているので、サービスオブジェクトは失敗がこれらの意味のひとつと同じときにのみ、これらの内のひとつを返すべきです。そうでなければ、アプリケーションは誤動作するでしょう。

第1.5版以降

失敗した時、コントロールオブジェクトは、実装されているなら、付加ステータス情報を得るために**GetOpenResult**メソッドを呼び出します。

ReleaseDevice / Release

形式 **MFC long ReleaseDevice();  
 long Release();**

**ATL HRESULT ReleaseDevice([out, retval] long\*** *pRC***);**

説明 デバイスの排他アクセスを解除するときにこのメソッドを呼び出します。

第1.0－1.4版

これらの版のコントロールオブジェクトは**Release**メソッドのみを探します。

第1.5版以降

これらの版のコントロールオブジェクトは最初に**ReleaseDevice**メソッドを調べます。もし**ReleaseDevice**が無ければ、コントロールオブジェクトは**Release**を探します。

ATL実装では**Release**メソッドを（少なくともATLクラスの更新／オーバーライトが無ければ）サポートできないことに注意して下さい。

## プロパティ

ここで説明するメソッドは、文字列型と４バイト数値型のプロパティを取得するまたは設定するために定義されています。

各メソッドの第一番目の引数は、  
 **LONG** *PropIndex*  
です。*PropIndex*の値は、共通プロパティ用としてOpos.hiで宣言されています。クラス特有のプロパティ値は、クラス特有なヘッダーファイルで宣言されています。

堅実性の点からも、サービスオブジェクトは*PropIndex*が正しいかどうか確認するべきです。無効な値が見つかった場合は、エラーを示すメッセージボックスを表示する、デバッグ例外を生成する、あるいは他の方法で開発者に警告を出すことも可能でしょう。この種のエラーは、開発中、テスト期間中、または顧客へ初公開される前の段階で発見されるべきものですから、ユーザーへ通知する方法は、幾分簡潔に出来るでしょう。

これらのメソッドはサービスオブジェクトのIDispatchインタフェース経由で常に呼び出されることを注意して下さい。

GetPropertyNumber

形式 **MFC long GetPropertyNumber(long** *PropIndex***);**

**ATL HRESULT GetPropertyNumber(long** *PropIndex***,  
 [out, retval] long\*** *pNumber***);**

戻り値 LONGまたはBOOL / VARIANT\_BOOLプロパティの現在の値です。

BOOLプロパティ－COM IDLのVARIANT\_BOOL型－に対し、コモンコントロールオブジェクトはVARIANT\_FALSEとしてゼロを返し、VARIANT\_TRUEとしてゼロ以外を返します。

GetPropertyString

形式 **MFC BSTR GetPropertyString(long** *PropIndex***);**

**ATL HRESULT GetPropertyString(long** *PropIndex***,  
 [out, retval] BSTR\*** *pString***);**

戻り値 文字列型プロパティの現在の値です。

SetPropertyNumber

形式 **MFC void SetPropertyNumber(long** *PropIndex***, long** *Number***);**

**ATL HRESULT SetPropertyNumber(long** *PropIndex***, long** *Number***);**

説明 LONGまたはBOOLプロパティを*Number*へ設定します。

BOOLプロパティ－COM IDLのVARIANT\_BOOL型－に対し、コモンコントロールオブジェクトはゼロ(0)としてゼロの値を返し、一(1)としてゼロ以外の値を返します。

SetPropertyString

形式 **MFC void SetPropertyNumber(long** *PropIndex***, long** *Number***);**

**ATL HRESULT SetPropertyNumber(long** *PropIndex***, long** *Number***);**

説明 文字列型プロパティを*String*へ設定します。

備考：プロパティの取得および設定メソッドを４つにまとめた理由

上記の４つのメソッドを使用する代わりに、サービスオブジェクトインタフェースで、すべてのプロパティに対して個別の取得メソッド、更に書き込み可能なプロパティに対して設定メソッドを定義することも可能でした。

しかし、複数のコントロールオブジェクトの中には、膨大な数のプロパティが存在するので、システム資源の節約及びサービスオブジェクトのコードを小さくするために上記の４つのメソッドが選ばれました。

他の型：BSTRでもLONGでもBOOLでもない型

コントロールが文字列でもLONGでもBOOLeanでもない型のプロパティを定義している場合は、サービスオブジェクトはこれらのプロパティのための設定・取得メソッドを追加定義する必要があります。

Visual C++とMFCを使っている場合、Class Wizardでオートメーションプロパティを追加することで簡単におこなえます。

その他の型のプロパティ値の取得

形式 **MFC *Type* Get*PropertyName*();**

**ATL HRESULT Get*PropertyName*([out, retval] *Type*\* pProp);**

***Type***の部分はプロパティの型に、***PropertyName***の部分はプロパティの名前に、それぞれ置き換えます。

戻り値 プロパティの現在の値です。

例：以下のプロパティがコントロールで定義されている場合

**CURRENCY SomeMoney**;

サービスオブジェクトは以下のメソッドを定義する必要があります。

**MFC CURRENCY GetSomeMoney();  
 ATL HRESULT GetSomeMoney([out, retval] CURRENCY\* pCY);**

その他の型のプロパティ値の設定

形式 **void Set*PropertyName* (*Type*** *value***);**

***Type***の部分はプロパティの型に、***PropertyName***の部分はプロパティの名前に、それぞれ置き換えます。

説明 プロパティを*value*へ設定します。

このメソッドは***PropertyName***プロパティが書き換え可能な場合のみ定義します。

例：以下のプロパティがコントロールで書き換え可能と定義されている場合

**CURRENCY SomeMoney**;

サービスオブジェクトは以下のメソッドを定義する必要があります。

**void GetSomeMoney(CURRENCY** *NewMoneyValue***)**;

## イベント

サービスオブジェクトは、コントロールオブジェクト内のイベントメソッドを呼び出してイベントを発行するきっかけをつくります。これらのメソッド名は次の通りです。

**SOData  
 SODirectIO  
 SOError  
 SOOutputComplete  
 SOStatusUpdate**

これらは、ページから始まるコントロールオブジェクトの章で説明されています。

これらの関数を呼び出すためのディスパッチインタフェース及びディスパッチIDの取得方法は、ページの**OpenService**の説明を参照してください。

# 付録A コモンコントロールオブジェクト

Curtiss Monroeの努力と、彼が所属する(Dayton, Ohioを根拠地とする) Research Computer Services, Inc.のOLE POS協議会への貢献により、OPOSコントロールオブジェクトの完全実装版が一般向けに開発されました。これらは、”コモンコントロールオブジェクト”と呼ばれています。

これらのコントロールオブジェクトはレファレンス的な実装として配布され、アプリケーションが直接使用しても正確かつ適切に動作すると考えられますが、全ベンダのサービスオブジェクトでの正確な動作を保証するものではありません。

## 特徴

* 全てのOPOSコントロールがサポートされています。
* ATLベースであり、デュアルインタフェースを使用しています。このため、アプリケーションは（IOPOSCashDrawerフォーム等の）IDispatchもしくはCOMインタフェースでアクセスできます。
* Microsoft Visual C++（現在はVersion 6.0、Service Pack 4）を使用して作成されています。
* サービスオブジェクトの全べてのリリースに対して互換性を持っています。旧SOに対するチェックを行い、未サポートのプロパティ、メソッドにアクセスすると適切なエラーを返します。
* 幾つかの大手ベンダのサービスオブジェクトでテストされました。
* コントロール上のスレッドからイベントを行儀良く通知するサービスオブジェクト以外に、コントロール外のスレッドからイベントを通知するサービスオブジェクトに対しても、イベント通知処理をサポートします。
* DLL依存性が無く、標準のOS DLLのみを使用しています。特に、MFCもしくはATLのDLLは使用していません。
* MBCSとUnicodeの両版でビルドされ、一定のテストが行われました。現在、MBCS版のみを公開しています。
* 全てのコントロールオブジェクトのソースコードが入手可能です。
* 将来の追加に際し、新規コントロールオブジェクトの追加と従来版のアップデードが容易に行えます。カスタマイズする開発者は、各コントロールのデータファイルを読み込んでビルドします。プロパティやメソッドの追加手順は、(a) データファイルを更新します、(b) ジェネレーションを行います、(c) そのプロジェクトをビルドします。

## 入手方法と将来

Curtissはコントロールオブジェクトの保守を意図しており、OPOSに所属している間は、必要に応じて新規リリースを含む修正をhttp://www.monroecs.com サイトで公開します。彼がこの作業を実行できなくなった際は、OLE POS協議会が引き継ぐことを承認しています。

新規デバイスのコントロールオブジェクトを提供するため、新規デバイス章の作成者に対し、ジェネレータへのキーアトリビュートを定義する2ページ程度のデータファイルの準備が要求されます。

# 付録B OPOSで定義する内部ヘッダーファイル

デバイスクラス名とヘッダーファイル名の対応は次のとおりです:

– OPOS基本 – Opos.hi  
Cash Drawer OposCash.hi  
Hard Totals OposTot.hi  
Keylock OposLock.hi  
Line Display OposDisp.hi  
MSR OposMsr.hi  
POS Keyboard OposKbd.hi  
POS Printer OposPtr.hi  
Scanner OposScan.hi  
Coin Dispenser OposCoin.hi  
MICR OposMicr.hi

Scale OposScal.hi  
Signature Capture OposSig.hi  
Cash Changer OposChan.hi  
Tone Indicator OposTone.hi  
Fiscal Printer OposFptr.hi  
PIN Pad OposPpad.hi  
Remote Order Display OposRod.hi  
Bump Bar OposBb.hi  
CAT OposCat.hi  
Point Card Reader Writer OposPcrw.hi  
POS Power OposPwr.hi  
Check Scanner OposChk.hi  
Motin Sensor OposMot.hi

最新のヘッダーファイルは以下のウェブサイトからダウンロードできます。

http://www.nrf-arts.org のOPOS標準ファイルセクション

## Opos.hi : OPOS基本内部ヘッダーファイル

//////////////////////////////////////////////////////////////////////

//

// Opos.hi

//

// General header file for OPOS Control Objects and Service Objects.

//

// Modification history

// -------------------------------------------------------------------

// 95-12-08 OPOS Release 1.0 CRM

// 96-03-18 OPOS Release 1.01 CRM

// Remove HKEY\_LOCAL\_MACHINE from the root keys, so that they

// may be directly used by RegOpenKey.

// 96-04-22 OPOS Release 1.1 CRM

// Add POS Keyboard values.

// 97-06-04 OPOS Release 1.2 CRM

// Add Cash Changer and Tone Indicator.

// Add the following properties:

// AutoDisable, BinaryConversion, DataCount

// 98-03-06 OPOS Release 1.3 CRM

// Add Bump Bar, Fiscal Printer, PIN Pad, Remote Order Display.

// Add the following properties:

// CapPowerReporting, PowerNotify, PowerState

// 98-09-29 OPOS Release 1.4 OPOS-J

// Add CAT.

// 00-09-24 OPOS Release 1.5 BKS

// Add Point Card Reader Writer and POS Power.

// 02-08-17 OPOS Release 1.7 CRM

// Add Check Scanner and Motion Sensor.

//

//////////////////////////////////////////////////////////////////////

#if !defined(OPOS\_HI)

#define OPOS\_HI

#include "Opos.h"

//////////////////////////////////////////////////////////////////////

// Registry Keys

//////////////////////////////////////////////////////////////////////

// OPOS\_ROOTKEY is the key under which all OPOS Service Object keys

// and values are placed. This key will contain device class keys.

// (The base key is HKEY\_LOCAL\_MACHINE.)

#define OPOS\_ROOTKEY \

"SOFTWARE\\OLEforRetail\\ServiceOPOS"

// Device Class Keys

// Rel 1.0

#define OPOS\_CLASSKEY\_CASH "CashDrawer"

#define OPOS\_CLASSKEY\_COIN "CoinDispenser"

#define OPOS\_CLASSKEY\_TOT "HardTotals"

#define OPOS\_CLASSKEY\_LOCK "Keylock"

#define OPOS\_CLASSKEY\_DISP "LineDisplay"

#define OPOS\_CLASSKEY\_MICR "MICR"

#define OPOS\_CLASSKEY\_MSR "MSR"

#define OPOS\_CLASSKEY\_PTR "POSPrinter"

#define OPOS\_CLASSKEY\_SCAL "Scale"

#define OPOS\_CLASSKEY\_SCAN "Scanner"

#define OPOS\_CLASSKEY\_SIG "SignatureCapture"

// Rel 1.1

#define OPOS\_CLASSKEY\_KBD "POSKeyboard"

// Rel 1.2

#define OPOS\_CLASSKEY\_CHAN "CashChanger"

#define OPOS\_CLASSKEY\_TONE "ToneIndicator"

// Rel 1.3

#define OPOS\_CLASSKEY\_BB "BumpBar"

#define OPOS\_CLASSKEY\_FPTR "FiscalPrinter"

#define OPOS\_CLASSKEY\_PPAD "PINPad"

#define OPOS\_CLASSKEY\_ROD "RemoteOrderDisplay"

// Rel 1.4

#define OPOS\_CLASSKEY\_CAT "CAT"

// Rel 1.5

#define OPOS\_CLASSKEY\_PCRW "PointCardRW"

#define OPOS\_CLASSKEY\_PWR "POSPower"

// Rel 1.7

#define OPOS\_CLASSKEY\_CHK "CheckScanner"

#define OPOS\_CLASSKEY\_MOTION "MotionSensor"

// OPOS\_ROOTKEY\_PROVIDER is the key under which a Service Object

// provider may place provider-specific information.

// (The base key is HKEY\_LOCAL\_MACHINE.)

#define OPOS\_ROOTKEY\_PROVIDER \

"SOFTWARE\\OLEforRetail\\ServiceInfo"

//////////////////////////////////////////////////////////////////////

// Common Property Base Index Values.

//////////////////////////////////////////////////////////////////////

// \* Base Indices \*

const LONG PIDX\_NUMBER = 0;

const LONG PIDX\_STRING = 1000000; // 1,000,000

// \* Range Test Functions \*

inline BOOL IsNumericPidx(LONG Pidx)

{

return ( Pidx < PIDX\_STRING ) ? TRUE : FALSE;

}

inline BOOL IsStringPidx(LONG Pidx)

{

return ( Pidx >= PIDX\_STRING ) ? TRUE : FALSE;

}

// \*\*Warning\*\*

// OPOS property indices may not be changed by future releases.

// New indices must be added sequentially at the end of the

// numeric, capability, and string sections.

// Also, the validation functions must be updated.

// Note: The ControlObjectDescription and ControlObjectVersion

// properties are processed entirely by the CO. Therefore, no

// property index values are required below.

//////////////////////////////////////////////////////////////////////

// Common Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDX\_Claimed = 1 + PIDX\_NUMBER;

const LONG PIDX\_DataEventEnabled = 2 + PIDX\_NUMBER;

const LONG PIDX\_DeviceEnabled = 3 + PIDX\_NUMBER;

const LONG PIDX\_FreezeEvents = 4 + PIDX\_NUMBER;

const LONG PIDX\_OutputID = 5 + PIDX\_NUMBER;

const LONG PIDX\_ResultCode = 6 + PIDX\_NUMBER;

const LONG PIDX\_ResultCodeExtended = 7 + PIDX\_NUMBER;

const LONG PIDX\_ServiceObjectVersion = 8 + PIDX\_NUMBER;

const LONG PIDX\_State = 9 + PIDX\_NUMBER;

// Added for Release 1.2:

const LONG PIDX\_AutoDisable = 10 + PIDX\_NUMBER;

const LONG PIDX\_BinaryConversion = 11 + PIDX\_NUMBER;

const LONG PIDX\_DataCount = 12 + PIDX\_NUMBER;

// Added for Release 1.3:

const LONG PIDX\_PowerNotify = 13 + PIDX\_NUMBER;

const LONG PIDX\_PowerState = 14 + PIDX\_NUMBER;

// \* Capabilities \*

// Added for Release 1.3:

const LONG PIDX\_CapPowerReporting = 501 + PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidNumericPidx(LONG Pidx)

{

return ( PIDX\_Claimed <= Pidx && Pidx <= PIDX\_State )

? TRUE : FALSE ;

}

inline BOOL IsValidNumericPidx12(LONG Pidx)

{

return ( PIDX\_Claimed <= Pidx && Pidx <= PIDX\_DataCount )

? TRUE : FALSE ;

}

inline BOOL IsValidNumericPidx13(LONG Pidx)

{

return ( PIDX\_Claimed <= Pidx && Pidx <= PIDX\_PowerState )

? TRUE : FALSE ;

}

inline BOOL IsValidCapPidx(LONG Pidx)

{

return FALSE ;

}

inline BOOL IsValidCapPidx13(LONG Pidx)

{

return ( PIDX\_CapPowerReporting == Pidx )

? TRUE : FALSE ;

}

//////////////////////////////////////////////////////////////////////

// Common String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDX\_CheckHealthText = 1 + PIDX\_STRING;

const LONG PIDX\_DeviceDescription = 2 + PIDX\_STRING;

const LONG PIDX\_DeviceName = 3 + PIDX\_STRING;

const LONG PIDX\_ServiceObjectDescription= 4 + PIDX\_STRING;

// \* Validation Function \*

inline BOOL IsValidStringPidx(LONG Pidx)

{

return ( PIDX\_CheckHealthText <= Pidx &&

Pidx <= PIDX\_ServiceObjectDescription )

? TRUE : FALSE ;

}

//////////////////////////////////////////////////////////////////////

// Class Property Base Index Values.

//////////////////////////////////////////////////////////////////////

// Rel 1.0

const LONG PIDX\_CASH = 1000; // Cash Drawer.

const LONG PIDX\_COIN = 2000; // Coin Dispenser.

const LONG PIDX\_TOT = 3000; // Hard Totals.

const LONG PIDX\_LOCK = 4000; // Keylock.

const LONG PIDX\_DISP = 5000; // Line Display.

const LONG PIDX\_MICR = 6000; // Magnetic Ink Character Recognition.

const LONG PIDX\_MSR = 7000; // Magnetic Stripe Reader.

const LONG PIDX\_PTR = 8000; // POS Printer.

const LONG PIDX\_SCAL = 9000; // Scale.

const LONG PIDX\_SCAN = 10000; // Scanner - Bar Code Reader.

const LONG PIDX\_SIG = 11000; // Signature Capture.

// Rel 1.1

const LONG PIDX\_KBD = 12000; // POS Keyboard.

// Rel 1.2

const LONG PIDX\_CHAN = 13000; // Cash Changer.

const LONG PIDX\_TONE = 14000; // Tone Indicator.

// Rel 1.3

const LONG PIDX\_BB = 15000; // Bump Bar.

const LONG PIDX\_FPTR = 16000; // Fiscal Printer.

const LONG PIDX\_PPAD = 17000; // PIN Pad.

const LONG PIDX\_ROD = 18000; // Remote Order Display.

// Rel 1.4

const LONG PIDX\_CAT = 19000; // CAT.

// Rel 1.5

const LONG PIDX\_PCRW = 20000; // Point Card Reader Writer.

const LONG PIDX\_PWR = 21000; // POS Power.

// Rel 1.7

const LONG PIDX\_CHK = 22000; // Check Scanner.

const LONG PIDX\_MOTION = 23000; // Motion Sensor.

#endif // !defined(OPOS\_HI)

## OposCash.hi : ドロワー内部ヘッダーファイル

//////////////////////////////////////////////////////////////////////

//

// OposCash.hi

//

// Cash Drawer header file for OPOS Controls and Service Objects.

//

// Modification history

// -------------------------------------------------------------------

// 95-12-08 OPOS Release 1.0 CRM

// 00-09-16 OPOS Release 1.5 BKS

//

//////////////////////////////////////////////////////////////////////

#if !defined(OPOSCASH\_HI)

#define OPOSCASH\_HI

#include "Opos.hi"

#include "OposCash.h"

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Property \*

const LONG PIDXCash\_DrawerOpened = 1 + PIDX\_CASH+PIDX\_NUMBER;

// \* Capabilities \*

const LONG PIDXCash\_CapStatus = 501 + PIDX\_CASH+PIDX\_NUMBER;

// Added in Release 1.5

const LONG PIDXCash\_CapStatusMultiDrawerDetect

= 502 + PIDX\_CASH+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidCashNumericPidx(LONG Pidx)

{

return ( PIDXCash\_DrawerOpened == Pidx )

? TRUE : FALSE ;

}

inline BOOL IsValidCashCapPidx(LONG Pidx)

{

return ( PIDXCash\_CapStatus == Pidx )

? TRUE : FALSE ;

}

inline BOOL IsValidCashCapPidx15(LONG Pidx)

{

return ( PIDXCash\_CapStatus >= Pidx &&

Pidx <= PIDXCash\_CapStatusMultiDrawerDetect )

? TRUE : FALSE ;

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Validation Function \*

inline BOOL IsValidCashStringPidx(LONG Pidx)

{

return FALSE;

}

#endif // !defined(OPOSCASH\_HI)

## OposTot.hi : ハードトータル内部ヘッダーファイル

//////////////////////////////////////////////////////////////////////

//

// OposTot.hi

//

// Hard Totals header file for OPOS Controls and Service Objects.

//

// Modification history

// -------------------------------------------------------------------

// 95-12-08 OPOS Release 1.0 CRM

//

//////////////////////////////////////////////////////////////////////

#if !defined(OPOSTOT\_HI)

#define OPOSTOT\_HI

#include "Opos.hi"

#include "OposTot.h"

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXTot\_FreeData = 1 + PIDX\_TOT+PIDX\_NUMBER;

const LONG PIDXTot\_NumberOfFiles = 2 + PIDX\_TOT+PIDX\_NUMBER;

const LONG PIDXTot\_TotalsSize = 3 + PIDX\_TOT+PIDX\_NUMBER;

const LONG PIDXTot\_TransactionInProgress

= 4 + PIDX\_TOT+PIDX\_NUMBER;

// \* Capabilities \*

const LONG PIDXTot\_CapErrorDetection = 501 + PIDX\_TOT+PIDX\_NUMBER;

const LONG PIDXTot\_CapSingleFile = 502 + PIDX\_TOT+PIDX\_NUMBER;

const LONG PIDXTot\_CapTransactions = 503 + PIDX\_TOT+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidTotNumericPidx(LONG Pidx)

{

return ( PIDXTot\_FreeData <= Pidx &&

Pidx <= PIDXTot\_TransactionInProgress )

? TRUE : FALSE ;

}

inline BOOL IsValidTotCapPidx(LONG Pidx)

{

return ( PIDXTot\_CapErrorDetection <= Pidx &&

Pidx <= PIDXTot\_CapTransactions )

? TRUE : FALSE ;

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Validation Function \*

inline BOOL IsValidTotStringPidx(LONG Pidx)

{

return FALSE;

}

#endif // !defined(OPOSTOT\_HI)

## OposDisp.hi : ラインディスプレイ内部ヘッダーファイル

//////////////////////////////////////////////////////////////////////

//

// OposDisp.hi

//

// Line Display header file for OPOS Controls and Service Objects.

//

// Modification history

// -------------------------------------------------------------------

// 95-12-08 OPOS Release 1.0 CRM

// 96-03-18 OPOS Release 1.01 CRM

// Add MarqueeFormat property.

// 01-07-15 OPOS Release 1.6 BKS

// Added BlinkRate, CursorType, CustomSSGlyphList, GlyphHeight

// and GlyphWidth properties.

// Added CapBlinkRate, CapCursorType, CapCustomGlyph, CapReadBack

// and CapReverse capabilities.

// 02-08-17 OPOS Release 1.7 CRM

// Added MapCharacterSet, MaximumX, MaximumY, ScreenMode,

// and ScreenModeList properties.

// Added CapBitmap, CapMapCharacterSet, and CapScreenMode

// capabilities.

//

//////////////////////////////////////////////////////////////////////

#if !defined(OPOSDISP\_HI)

#define OPOSDISP\_HI

#include "Opos.hi"

#include "OposDisp.h"

#if defined(CreateWindow) // If Win32 defines "CreateWindow":

#undef CreateWindow // Undefine it to avoid conflict

#endif // with the line display method.

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXDisp\_CharacterSet = 1 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_Columns = 2 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CurrentWindow = 3 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CursorColumn = 4 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CursorRow = 5 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CursorUpdate = 6 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_DeviceBrightness = 7 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_DeviceColumns = 8 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_DeviceDescriptors = 9 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_DeviceRows = 10 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_DeviceWindows = 11 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_InterCharacterWait = 12 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_MarqueeRepeatWait = 13 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_MarqueeType = 14 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_MarqueeUnitWait = 15 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_Rows = 16 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_MarqueeFormat = 17 + PIDX\_DISP+PIDX\_NUMBER;

// Added in Release 1.6

const LONG PIDXDisp\_BlinkRate = 18 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CursorType = 19 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_GlyphHeight = 20 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_GlyphWidth = 21 + PIDX\_DISP+PIDX\_NUMBER;

// Added in Release 1.7

const LONG PIDXDisp\_MapCharacterSet = 22 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_MaximumX = 23 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_MaximumY = 24 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_ScreenMode = 25 + PIDX\_DISP+PIDX\_NUMBER;

// \* Capabilities \*

const LONG PIDXDisp\_CapBlink = 501 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CapBrightness = 502 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CapCharacterSet = 503 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CapDescriptors = 504 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CapHMarquee = 505 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CapICharWait = 506 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CapVMarquee = 507 + PIDX\_DISP+PIDX\_NUMBER;

// Added in Release 1.6

const LONG PIDXDisp\_CapBlinkRate = 508 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CapCursorType = 509 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CapCustomGlyph = 510 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CapReadBack = 511 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CapReverse = 512 + PIDX\_DISP+PIDX\_NUMBER;

// Added in Release 1.7

const LONG PIDXDisp\_CapBitmap = 513 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CapMapCharacterSet = 514 + PIDX\_DISP+PIDX\_NUMBER;

const LONG PIDXDisp\_CapScreenMode = 515 + PIDX\_DISP+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidDispNumericPidx(LONG Pidx)

{

return ( PIDXDisp\_CharacterSet <= Pidx &&

Pidx <= PIDXDisp\_MarqueeFormat )

? TRUE : FALSE ;

}

inline BOOL IsValidDispNumericPidx12(LONG Pidx)

{

return ( PIDXDisp\_CharacterSet <= Pidx &&

Pidx <= PIDXDisp\_MarqueeFormat )

? TRUE : FALSE ;

}

inline BOOL IsValidDispNumericPidx16(LONG Pidx)

{

return ( PIDXDisp\_CharacterSet <= Pidx &&

Pidx <= PIDXDisp\_GlyphWidth )

? TRUE : FALSE ;

}

inline BOOL IsValidDispNumericPidx17(LONG Pidx)

{

return ( PIDXDisp\_CharacterSet <= Pidx &&

Pidx <= PIDXDisp\_ScreenMode )

? TRUE : FALSE ;

}

inline BOOL IsValidDispCapPidx(LONG Pidx)

{

return ( PIDXDisp\_CapBlink <= Pidx && Pidx <= PIDXDisp\_CapVMarquee )

? TRUE : FALSE ;

}

inline BOOL IsValidDispCapPidx12(LONG Pidx)

{

return ( PIDXDisp\_CapBlink <= Pidx && Pidx <= PIDXDisp\_CapVMarquee )

? TRUE : FALSE ;

}

inline BOOL IsValidDispCapPidx16(LONG Pidx)

{

return ( PIDXDisp\_CapBlink <= Pidx && Pidx <= PIDXDisp\_CapReverse )

? TRUE : FALSE ;

}

inline BOOL IsValidDispCapPidx17(LONG Pidx)

{

return ( PIDXDisp\_CapBlink <= Pidx && Pidx <= PIDXDisp\_CapScreenMode )

? TRUE : FALSE ;

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXDisp\_CharacterSetList = 1 + PIDX\_DISP+PIDX\_STRING;

// Added in Release 1.6

const LONG PIDXDisp\_CustomGlyphList = 2 + PIDX\_DISP+PIDX\_STRING;

// Added in Release 1.7

const LONG PIDXDisp\_ScreenModeList = 3 + PIDX\_DISP+PIDX\_STRING;

// \* Validation Function \*

inline BOOL IsValidDispStringPidx(LONG Pidx)

{

return ( PIDXDisp\_CharacterSetList == Pidx )

? TRUE : FALSE ;

}

inline BOOL IsValidDispStringPidx12(LONG Pidx)

{

return ( PIDXDisp\_CharacterSetList == Pidx )

? TRUE : FALSE ;

}

inline BOOL IsValidDispStringPidx16(LONG Pidx)

{

return ( PIDXDisp\_CharacterSetList <= Pidx &&

Pidx <= PIDXDisp\_CustomGlyphList )

? TRUE : FALSE ;

}

inline BOOL IsValidDispStringPidx17(LONG Pidx)

{

return ( PIDXDisp\_CharacterSetList <= Pidx &&

Pidx <= PIDXDisp\_ScreenModeList )

? TRUE : FALSE ;

}

#endif // !defined(OPOSDISP\_HI)

## OposLock.hi : キーロック内部ヘッダーファイル

//////////////////////////////////////////////////////////////////////

//

// OposLock.hi

//

// Keylock header file for OPOS Controls and Service Objects.

//

// Modification history

// -------------------------------------------------------------------

// 95-12-08 OPOS Release 1.0 CRM

//

//////////////////////////////////////////////////////////////////////

#if !defined(OPOSLOCK\_HI)

#define OPOSLOCK\_HI

#include "Opos.hi"

#include "OposLock.h"

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXLock\_KeyPosition = 1 + PIDX\_LOCK+PIDX\_NUMBER;

const LONG PIDXLock\_PositionCount = 2 + PIDX\_LOCK+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidLockNumericPidx(LONG Pidx)

{

return ( PIDXLock\_KeyPosition <= Pidx &&

Pidx <= PIDXLock\_PositionCount )

? TRUE : FALSE ;

}

inline BOOL IsValidLockCapPidx(LONG Pidx)

{

return FALSE;

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Validation Function \*

inline BOOL IsValidLockStringPidx(LONG Pidx)

{

return FALSE;

}

#endif // !defined(OPOSLOCK\_HI)

## OposMsr.hi : 磁気ストライプリーダ内部ヘッダーファイル

//////////////////////////////////////////////////////////////////////

//

// OposMsr.hi

//

// MSR header file for OPOS Controls and Service Objects.

//

// Modification history

// -------------------------------------------------------------------

// 95-12-08 OPOS Release 1.0 CRM

// 97-06-04 OPOS Release 1.2 CRM

// Add the following properties:

// ErrorReportingType, ParseDecodedData

// 00-09-16 OPOS Release 1.5 BKS

// Add the following properties:

// CapTransmitSentinels, Track4Data, TransmitSentinels

//

//////////////////////////////////////////////////////////////////////

#if !defined(OPOSMSR\_HI)

#define OPOSMSR\_HI

#include "Opos.hi"

#include "OposMsr.h"

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXMsr\_DecodeData = 1 + PIDX\_MSR+PIDX\_NUMBER;

const LONG PIDXMsr\_ParseDecodeData = 2 + PIDX\_MSR+PIDX\_NUMBER;

const LONG PIDXMsr\_TracksToRead = 3 + PIDX\_MSR+PIDX\_NUMBER;

// Added for Release 1.2:

const LONG PIDXMsr\_ParseDecodedData = 2 + PIDX\_MSR+PIDX\_NUMBER;

// ParseDecodedData = ParseDecodeData: Support both, due to

// editing error in the pre-1.2 APG.

const LONG PIDXMsr\_ErrorReportingType = 4 + PIDX\_MSR+PIDX\_NUMBER;

// Added for Release 1.5:

const LONG PIDXMsr\_TransmitSentinels = 5 + PIDX\_MSR+PIDX\_NUMBER;

// \* Capabilities \*

const LONG PIDXMsr\_CapISO = 501 + PIDX\_MSR+PIDX\_NUMBER;

const LONG PIDXMsr\_CapJISOne = 502 + PIDX\_MSR+PIDX\_NUMBER;

const LONG PIDXMsr\_CapJISTwo = 503 + PIDX\_MSR+PIDX\_NUMBER;

// Added for Release 1.5:

const LONG PIDXMsr\_CapTransmitSentinels = 504 + PIDX\_MSR+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidMsrNumericPidx(LONG Pidx)

{

return ( PIDXMsr\_DecodeData <= Pidx && Pidx <= PIDXMsr\_TracksToRead )

? TRUE : FALSE ;

}

inline BOOL IsValidMsrNumericPidx12(LONG Pidx)

{

return ( PIDXMsr\_DecodeData <= Pidx &&

Pidx <= PIDXMsr\_ErrorReportingType )

? TRUE : FALSE ;

}

inline BOOL IsValidMsrNumericPidx15(LONG Pidx)

{

return ( PIDXMsr\_DecodeData <= Pidx &&

Pidx <= PIDXMsr\_TransmitSentinels )

? TRUE : FALSE ;

}

inline BOOL IsValidMsrCapPidx(LONG Pidx)

{

return ( PIDXMsr\_CapISO <= Pidx && Pidx <= PIDXMsr\_CapJISTwo )

? TRUE : FALSE ;

}

inline BOOL IsValidMsrCapPidx15(LONG Pidx)

{

return ( PIDXMsr\_CapISO <= Pidx &&

Pidx <= PIDXMsr\_CapTransmitSentinels )

? TRUE : FALSE ;

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXMsr\_AccountNumber = 1 + PIDX\_MSR+PIDX\_STRING;

const LONG PIDXMsr\_ExpirationDate = 2 + PIDX\_MSR+PIDX\_STRING;

const LONG PIDXMsr\_FirstName = 3 + PIDX\_MSR+PIDX\_STRING;

const LONG PIDXMsr\_MiddleInitial = 4 + PIDX\_MSR+PIDX\_STRING;

const LONG PIDXMsr\_ServiceCode = 5 + PIDX\_MSR+PIDX\_STRING;

const LONG PIDXMsr\_Suffix = 6 + PIDX\_MSR+PIDX\_STRING;

const LONG PIDXMsr\_Surname = 7 + PIDX\_MSR+PIDX\_STRING;

const LONG PIDXMsr\_Title = 8 + PIDX\_MSR+PIDX\_STRING;

const LONG PIDXMsr\_Track1Data = 9 + PIDX\_MSR+PIDX\_STRING;

const LONG PIDXMsr\_Track1DiscretionaryData

= 10 + PIDX\_MSR+PIDX\_STRING;

const LONG PIDXMsr\_Track2Data = 11 + PIDX\_MSR+PIDX\_STRING;

const LONG PIDXMsr\_Track2DiscretionaryData

= 12 + PIDX\_MSR+PIDX\_STRING;

const LONG PIDXMsr\_Track3Data = 13 + PIDX\_MSR+PIDX\_STRING;

// Added for Release 1.5:

const LONG PIDXMsr\_Track4Data = 14 + PIDX\_MSR+PIDX\_STRING;

// \* Validation Function \*

inline BOOL IsValidMsrStringPidx(LONG Pidx)

{

return ( PIDXMsr\_AccountNumber <= Pidx &&

Pidx <= PIDXMsr\_Track3Data )

? TRUE : FALSE ;

}

inline BOOL IsValidMsrStringPidx15(LONG Pidx)

{

return ( PIDXMsr\_AccountNumber <= Pidx &&

Pidx <= PIDXMsr\_Track4Data )

? TRUE : FALSE ;

}

#endif // !defined(OPOSMSR\_HI)

## OposKbd.hi : POSキーボード内部ヘッダーファイル

//////////////////////////////////////////////////////////////////////

//

// OposKbd.hi

//

// POS Keyboard header file for OPOS Controls and Service Objects.

//

// Modification history

// -------------------------------------------------------------------

// 96-04-22 OPOS Release 1.1 CRM

// 97-06-04 OPOS Release 1.2 CRM

// Add the following properties:

// CapKeyUp, EventTypes, POSKeyEventType

//

//////////////////////////////////////////////////////////////////////

#if !defined(OPOSKBD\_HI)

#define OPOSKBD\_HI

#include "Opos.hi"

#include "OposKbd.h"

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXKbd\_POSKeyData = 1 + PIDX\_KBD+PIDX\_NUMBER;

// Added for Release 1.2:

const LONG PIDXKbd\_EventTypes = 2 + PIDX\_KBD+PIDX\_NUMBER;

const LONG PIDXKbd\_POSKeyEventType = 3 + PIDX\_KBD+PIDX\_NUMBER;

// \* Capabilities \*

// Added for Release 1.2:

const LONG PIDXKbd\_CapKeyUp = 501 + PIDX\_KBD+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidKbdNumericPidx(LONG Pidx)

{

return ( PIDXKbd\_POSKeyData == Pidx )

? TRUE : FALSE ;

}

inline BOOL IsValidKbdNumericPidx11(LONG Pidx)

{

return IsValidKbdNumericPidx(Pidx);

}

inline BOOL IsValidKbdNumericPidx12(LONG Pidx)

{

return ( PIDXKbd\_POSKeyData <= Pidx &&

Pidx <= PIDXKbd\_POSKeyEventType )

? TRUE : FALSE ;

}

inline BOOL IsValidKbdCapPidx(LONG Pidx)

{

return FALSE;

}

inline BOOL IsValidKbdCapPidx11(LONG Pidx)

{

return IsValidKbdCapPidx(Pidx);

}

inline BOOL IsValidKbdCapPidx12(LONG Pidx)

{

return ( PIDXKbd\_CapKeyUp == Pidx )

? TRUE : FALSE ;

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Validation Function \*

inline BOOL IsValidKbdStringPidx(LONG Pidx)

{

return FALSE;

}

inline BOOL IsValidKbdStringPidx11(LONG Pidx)

{

return IsValidKbdStringPidx(Pidx);

}

#endif // !defined(OPOSKBD\_HI)

## OposPtr.hi : POSプリンタ内部ヘッダーファイル

//////////////////////////////////////////////////////////////////////

//

// OposPtr.hi

//

// POS Printer header file for OPOS Controls and Service Objects.

//

// Modification history

// -------------------------------------------------------------------

// 95-12-08 OPOS Release 1.0 CRM

// 96-03-18 OPOS Release 1.01 CRM

// Change ...Nearend to ...NearEnd.

// Change ...Barcode to ...BarCode.

// Corrected IsValidPtrNumericPidx function.

// 96-04-22 OPOS Release 1.1 CRM

// Add the following properties:

// CapCharacterSet, CapTransaction, ErrorLevel, RotateSpecial,

// ErrorString, FontTypefaceList, RecBarCodeRotationList,

// SlpBarCodeRotationList

// 00-09-24 OPOS Release 1.5 BKS

// Add the following properties:

// CapJrnCartridgeSensor, CapJrnColor, CapRecCartrdigeSensor,

// CapRecColor, CapRecMarkFeed, CapSlpBothSidesPrint,

// CapSlpCartridgeSensor, CapSlpColor, CartridgeNotify,

// JrnCartridgeState, JrnCurrentCartridge, RecCartridgeState,

// RecCurrentCartridge, SlpPrintSide, SlpCartridgeState,

// SlpCurrentCartridge

// 02-08-17 OPOS Release 1.7 CRM

// Added MapCharacterSet, RecBitmapRotationList, and

// SlpBitmapRotationList properties.

// Added CapMapCharacterSet capability.

//

//////////////////////////////////////////////////////////////////////

#if !defined(OPOSPTR\_HI)

#define OPOSPTR\_HI

#include "Opos.hi"

#include "OposPtr.h"

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXPtr\_AsyncMode = 1 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CharacterSet = 2 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CoverOpen = 3 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_ErrorStation = 4 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_FlagWhenIdle = 5 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_JrnEmpty = 6 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_JrnLetterQuality = 7 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_JrnLineChars = 8 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_JrnLineHeight = 9 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_JrnLineSpacing = 10 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_JrnLineWidth = 11 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_JrnNearEnd = 12 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_MapMode = 13 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_RecEmpty = 14 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_RecLetterQuality = 15 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_RecLineChars = 16 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_RecLineHeight = 17 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_RecLineSpacing = 18 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_RecLinesToPaperCut = 19 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_RecLineWidth = 20 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_RecNearEnd = 21 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_RecSidewaysMaxChars = 22 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_RecSidewaysMaxLines = 23 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_SlpEmpty = 24 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_SlpLetterQuality = 25 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_SlpLineChars = 26 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_SlpLineHeight = 27 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_SlpLinesNearEndToEnd = 28 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_SlpLineSpacing = 29 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_SlpLineWidth = 30 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_SlpMaxLines = 31 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_SlpNearEnd = 32 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_SlpSidewaysMaxChars = 33 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_SlpSidewaysMaxLines = 34 + PIDX\_PTR+PIDX\_NUMBER;

// Added for Release 1.1:

const LONG PIDXPtr\_ErrorLevel = 35 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_RotateSpecial = 36 + PIDX\_PTR+PIDX\_NUMBER;

// Added for Release 1.5:

const LONG PIDXPtr\_CartridgeNotify = 37 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_JrnCartridgeState = 38 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_JrnCurrentCartridge = 39 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_RecCartridgeState = 40 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_RecCurrentCartridge = 41 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_SlpPrintSide = 42 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_SlpCartridgeState = 43 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_SlpCurrentCartridge = 44 + PIDX\_PTR+PIDX\_NUMBER;

// Added in Release 1.7

const LONG PIDXPtr\_MapCharacterSet = 45 + PIDX\_PTR+PIDX\_NUMBER;

// \* Capabilities \*

const LONG PIDXPtr\_CapConcurrentJrnRec = 501 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapConcurrentJrnSlp = 502 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapConcurrentRecSlp = 503 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapCoverSensor = 504 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapJrn2Color = 505 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapJrnBold = 506 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapJrnDhigh = 507 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapJrnDwide = 508 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapJrnDwideDhigh = 509 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapJrnEmptySensor = 510 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapJrnItalic = 511 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapJrnNearEndSensor = 512 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapJrnPresent = 513 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapJrnUnderline = 514 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRec2Color = 515 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecBarCode = 516 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecBitmap = 517 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecBold = 518 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecDhigh = 519 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecDwide = 520 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecDwideDhigh = 521 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecEmptySensor = 522 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecItalic = 523 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecLeft90 = 524 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecNearEndSensor = 525 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecPapercut = 526 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecPresent = 527 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecRight90 = 528 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecRotate180 = 529 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecStamp = 530 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecUnderline = 531 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlp2Color = 532 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpBarCode = 533 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpBitmap = 534 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpBold = 535 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpDhigh = 536 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpDwide = 537 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpDwideDhigh = 538 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpEmptySensor = 539 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpFullslip = 540 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpItalic = 541 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpLeft90 = 542 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpNearEndSensor = 543 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpPresent = 544 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpRight90 = 545 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpRotate180 = 546 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpUnderline = 547 + PIDX\_PTR+PIDX\_NUMBER;

// Added for Release 1.1:

const LONG PIDXPtr\_CapCharacterSet = 548 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapTransaction = 549 + PIDX\_PTR+PIDX\_NUMBER;

// Added for Release 1.5:

const LONG PIDXPtr\_CapJrnCartridgeSensor

= 550 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapJrnColor = 551 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecCartridgeSensor

= 552 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecColor = 553 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapRecMarkFeed = 554 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpBothSidesPrint = 555 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpCartridgeSensor

= 556 + PIDX\_PTR+PIDX\_NUMBER;

const LONG PIDXPtr\_CapSlpColor = 557 + PIDX\_PTR+PIDX\_NUMBER;

// Added in Release 1.7

const LONG PIDXPtr\_CapMapCharacterSet = 558 + PIDX\_PTR+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidPtrNumericPidx(LONG Pidx)

{

return ( PIDXPtr\_AsyncMode <= Pidx &&

Pidx <= PIDXPtr\_SlpSidewaysMaxLines )

? TRUE : FALSE ;

}

inline BOOL IsValidPtrNumericPidx11(LONG Pidx)

{

return ( PIDXPtr\_AsyncMode <= Pidx &&

Pidx <= PIDXPtr\_RotateSpecial )

? TRUE : FALSE ;

}

inline BOOL IsValidPtrNumericPidx15(LONG Pidx)

{

return ( PIDXPtr\_AsyncMode <= Pidx &&

Pidx <= PIDXPtr\_SlpCurrentCartridge )

? TRUE : FALSE ;

}

inline BOOL IsValidPtrNumericPidx17(LONG Pidx)

{

return ( PIDXPtr\_AsyncMode <= Pidx &&

Pidx <= PIDXPtr\_MapCharacterSet )

? TRUE : FALSE ;

}

inline BOOL IsValidPtrCapPidx(LONG Pidx)

{

return ( PIDXPtr\_CapConcurrentJrnRec <= Pidx &&

Pidx <= PIDXPtr\_CapSlpUnderline )

? TRUE : FALSE ;

}

inline BOOL IsValidPtrCapPidx11(LONG Pidx)

{

return ( PIDXPtr\_CapConcurrentJrnRec <= Pidx &&

Pidx <= PIDXPtr\_CapTransaction )

? TRUE : FALSE ;

}

inline BOOL IsValidPtrCapPidx15(LONG Pidx)

{

return ( PIDXPtr\_CapConcurrentJrnRec <= Pidx &&

Pidx <= PIDXPtr\_CapSlpColor )

? TRUE : FALSE ;

}

inline BOOL IsValidPtrCapPidx17(LONG Pidx)

{

return ( PIDXPtr\_CapConcurrentJrnRec <= Pidx &&

Pidx <= PIDXPtr\_CapMapCharacterSet )

? TRUE : FALSE ;

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXPtr\_CharacterSetList = 1 + PIDX\_PTR+PIDX\_STRING;

const LONG PIDXPtr\_JrnLineCharsList = 2 + PIDX\_PTR+PIDX\_STRING;

const LONG PIDXPtr\_RecLineCharsList = 3 + PIDX\_PTR+PIDX\_STRING;

const LONG PIDXPtr\_SlpLineCharsList = 4 + PIDX\_PTR+PIDX\_STRING;

// Added for Release 1.1:

const LONG PIDXPtr\_ErrorString = 5 + PIDX\_PTR+PIDX\_STRING;

const LONG PIDXPtr\_FontTypefaceList = 6 + PIDX\_PTR+PIDX\_STRING;

const LONG PIDXPtr\_RecBarCodeRotationList

= 7 + PIDX\_PTR+PIDX\_STRING;

const LONG PIDXPtr\_SlpBarCodeRotationList

= 8 + PIDX\_PTR+PIDX\_STRING;

// Added in Release 1.7

const LONG PIDXPtr\_RecBitmapRotationList= 9 + PIDX\_PTR+PIDX\_STRING;

const LONG PIDXPtr\_SlpBitmapRotationList= 10 + PIDX\_PTR+PIDX\_STRING;

// \* Validation Functions \*

inline BOOL IsValidPtrStringPidx(LONG Pidx)

{

return ( PIDXPtr\_CharacterSetList <= Pidx &&

Pidx <= PIDXPtr\_SlpLineCharsList )

? TRUE : FALSE ;

}

inline BOOL IsValidPtrStringPidx11(LONG Pidx)

{

return ( PIDXPtr\_CharacterSetList <= Pidx &&

Pidx <= PIDXPtr\_SlpBarCodeRotationList )

? TRUE : FALSE ;

}

inline BOOL IsValidPtrStringPidx17(LONG Pidx)

{

return ( PIDXPtr\_CharacterSetList <= Pidx &&

Pidx <= PIDXPtr\_SlpBitmapRotationList )

? TRUE : FALSE ;

}

#endif // !defined(OPOSPTR\_HI)

## OposScan.hi : スキャナ内部ヘッダーファイル

//////////////////////////////////////////////////////////////////////

//

// OposScan.hi

//

// Scanner header file for OPOS Controls and Service Objects.

//

// Modification history

// -------------------------------------------------------------------

// 95-12-08 OPOS Release 1.0 CRM

// 97-06-04 OPOS Release 1.2 CRM

// Add the following properties:

// DecodeData, ScanDataType, ScanDataLabel

//

//////////////////////////////////////////////////////////////////////

#if !defined(OPOSSCAN\_HI)

#define OPOSSCAN\_HI

#include "Opos.hi"

#include "OposScan.h"

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

// Added for Release 1.2:

const LONG PIDXScan\_DecodeData = 1 + PIDX\_SCAN+PIDX\_NUMBER;

const LONG PIDXScan\_ScanDataType = 2 + PIDX\_SCAN+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidScanNumericPidx(LONG Pidx)

{

return FALSE;

}

inline BOOL IsValidScanNumericPidx12(LONG Pidx)

{

return ( PIDXScan\_DecodeData <= Pidx &&

Pidx <= PIDXScan\_ScanDataType )

? TRUE : FALSE ;

}

inline BOOL IsValidScanCapPidx(LONG Pidx)

{

return FALSE;

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Property \*

const LONG PIDXScan\_ScanData = 1 + PIDX\_SCAN+PIDX\_STRING;

// Added for Release 1.2:

const LONG PIDXScan\_ScanDataLabel = 2 + PIDX\_SCAN+PIDX\_STRING;

// \* Validation Functions \*

inline BOOL IsValidScanStringPidx(LONG Pidx)

{

return ( PIDXScan\_ScanData == Pidx )

? TRUE : FALSE ;

}

inline BOOL IsValidScanStringPidx12(LONG Pidx)

{

return ( PIDXScan\_ScanData <= Pidx &&

Pidx <= PIDXScan\_ScanDataLabel )

? TRUE : FALSE ;

}

#endif // !defined(OPOSSCAN\_HI)

## OposCoin.hi : コインディスペンサ内部ヘッダーファイル

//////////////////////////////////////////////////////////////////////

//

// OposCoin.hi

//

// Coin Dispenser header file for OPOS Controls and Service Objects.

//

// Modification history

// -------------------------------------------------------------------

// 95-12-08 OPOS Release 1.0 CRM

//

//////////////////////////////////////////////////////////////////////

#if !defined(OPOSCOIN\_HI)

#define OPOSCOIN\_HI

#include "Opos.hi"

#include "OposCoin.h"

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Property \*

const LONG PIDXCoin\_DispenserStatus = 1 + PIDX\_COIN+PIDX\_NUMBER;

// \* Capabilities \*

const LONG PIDXCoin\_CapEmptySensor = 501 + PIDX\_COIN+PIDX\_NUMBER;

const LONG PIDXCoin\_CapJamSensor = 502 + PIDX\_COIN+PIDX\_NUMBER;

const LONG PIDXCoin\_CapNearEmptySensor = 503 + PIDX\_COIN+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidCoinNumericPidx(LONG Pidx)

{

return ( PIDXCoin\_DispenserStatus == Pidx )

? TRUE : FALSE ;

}

inline BOOL IsValidCoinCapPidx(LONG Pidx)

{

return ( PIDXCoin\_CapEmptySensor <= Pidx &&

Pidx <= PIDXCoin\_CapNearEmptySensor )

? TRUE : FALSE ;

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Validation Function \*

inline BOOL IsValidCoinStringPidx(LONG Pidx)

{

return FALSE;

}

#endif // !defined(OPOSCOIN\_HI)

## OposScal.hi : 秤内部ヘッダーファイル

//////////////////////////////////////////////////////////////////////

//

// OposScal.hi

//

// Scale header file for OPOS Controls and Service Objects.

//

// Modification history

// -------------------------------------------------------------------

// 95-12-08 OPOS Release 1.0 CRM

// 96-03-18 OPOS Release 1.01 CRM

// Correct WeightUnits value from 1 to 2.

// 97-06-04 OPOS Release 1.2 CRM

// Add the following properties: CapDisplay, WeightUnit

// 98-03-06 OPOS Release 1.3 CRM

// Add the following properties:

// CapDisplayText, CapPriceCalculating, CapTareWeight,

// CapZeroScale, AsyncMode, MaxDisplayTextChars, SalesPrice,

// TareWeight, UnitPrice

//

//////////////////////////////////////////////////////////////////////

#if !defined(OPOSSCAL\_HI)

#define OPOSSCAL\_HI

#include "Opos.hi"

#include "OposScal.h"

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXScal\_MaximumWeight = 1 + PIDX\_SCAL+PIDX\_NUMBER;

const LONG PIDXScal\_WeightUnits = 2 + PIDX\_SCAL+PIDX\_NUMBER;

// Added for Release 1.2:

const LONG PIDXScal\_WeightUnit = 2 + PIDX\_SCAL+PIDX\_NUMBER;

// WeightUnit = WeightUnits: Support both, due to

// editing error in the pre-1.2 APG.

// Added for Release 1.3:

const LONG PIDXScal\_AsyncMode = 3 + PIDX\_SCAL+PIDX\_NUMBER;

const LONG PIDXScal\_MaxDisplayTextChars = 4 + PIDX\_SCAL+PIDX\_NUMBER;

const LONG PIDXScal\_TareWeight = 5 + PIDX\_SCAL+PIDX\_NUMBER;

// \* Capabilities \*

// Added for Release 1.1:

const LONG PIDXScal\_CapDisplay = 501 + PIDX\_SCAL+PIDX\_NUMBER;

// Added for Release 1.3:

const LONG PIDXScal\_CapDisplayText = 502 + PIDX\_SCAL+PIDX\_NUMBER;

const LONG PIDXScal\_CapPriceCalculating = 503 + PIDX\_SCAL+PIDX\_NUMBER;

const LONG PIDXScal\_CapTareWeight = 504 + PIDX\_SCAL+PIDX\_NUMBER;

const LONG PIDXScal\_CapZeroScale = 505 + PIDX\_SCAL+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidScalNumericPidx(LONG Pidx)

{

return ( PIDXScal\_MaximumWeight <= Pidx &&

Pidx <= PIDXScal\_WeightUnits )

? TRUE : FALSE ;

}

inline BOOL IsValidScalNumericPidx13(LONG Pidx)

{

return ( PIDXScal\_MaximumWeight <= Pidx &&

Pidx <= PIDXScal\_TareWeight )

? TRUE : FALSE ;

}

inline BOOL IsValidScalCapPidx(LONG Pidx)

{

return FALSE;

}

inline BOOL IsValidScalCapPidx12(LONG Pidx)

{

return ( PIDXScal\_CapDisplay == Pidx )

? TRUE : FALSE ;

}

inline BOOL IsValidScalCapPidx13(LONG Pidx)

{

return ( PIDXScal\_CapDisplay <= Pidx &&

Pidx <= PIDXScal\_CapZeroScale )

? TRUE : FALSE ;

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Validation Function \*

inline BOOL IsValidScalStringPidx(LONG Pidx)

{

return FALSE;

}

#endif // !defined(OPOSSCAL\_HI)

## OposChan.hi : 自動釣り銭機内部ヘッダーファイル

//////////////////////////////////////////////////////////////////////

//

// OposChan.hi

//

// Cash Changer header file for OPOS Controls and Service Objects.

//

// Modification history

// -------------------------------------------------------------------

// 97-06-04 OPOS Release 1.2 CRM

// 00-09-16 OPOS Release 1.5 BKS

//

//////////////////////////////////////////////////////////////////////

#if !defined(OPOSCHAN\_HI)

#define OPOSCHAN\_HI

#include "Opos.hi"

#include "OposChan.h"

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXChan\_AsyncMode = 1 + PIDX\_CHAN+PIDX\_NUMBER;

const LONG PIDXChan\_AsyncResultCode = 2 + PIDX\_CHAN+PIDX\_NUMBER;

const LONG PIDXChan\_AsyncResultCodeExtended

= 3 + PIDX\_CHAN+PIDX\_NUMBER;

const LONG PIDXChan\_CurrentExit = 4 + PIDX\_CHAN+PIDX\_NUMBER;

const LONG PIDXChan\_DeviceExits = 5 + PIDX\_CHAN+PIDX\_NUMBER;

const LONG PIDXChan\_DeviceStatus = 6 + PIDX\_CHAN+PIDX\_NUMBER;

const LONG PIDXChan\_FullStatus = 7 + PIDX\_CHAN+PIDX\_NUMBER;

// Added in Release 1.5

const LONG PIDXChan\_DepositAmount = 8 + PIDX\_CHAN+PIDX\_NUMBER;

const LONG PIDXChan\_DepositStatus = 9 + PIDX\_CHAN+PIDX\_NUMBER;

// \* Capabilities \*

const LONG PIDXChan\_CapDiscrepancy = 501 + PIDX\_CHAN+PIDX\_NUMBER;

const LONG PIDXChan\_CapEmptySensor = 502 + PIDX\_CHAN+PIDX\_NUMBER;

const LONG PIDXChan\_CapFullSensor = 503 + PIDX\_CHAN+PIDX\_NUMBER;

const LONG PIDXChan\_CapNearEmptySensor = 504 + PIDX\_CHAN+PIDX\_NUMBER;

const LONG PIDXChan\_CapNearFullSensor = 505 + PIDX\_CHAN+PIDX\_NUMBER;

// Added in Release 1.5

const LONG PIDXChan\_CapDeposit = 506 + PIDX\_CHAN+PIDX\_NUMBER;

const LONG PIDXChan\_CapDepositDataEvent = 507 + PIDX\_CHAN+PIDX\_NUMBER;

const LONG PIDXChan\_CapPauseDeposit = 508 + PIDX\_CHAN+PIDX\_NUMBER;

const LONG PIDXChan\_CapRepayDeposit = 509 + PIDX\_CHAN+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidChanNumericPidx(LONG Pidx)

{

return ( PIDXChan\_AsyncMode <= Pidx && Pidx <= PIDXChan\_FullStatus )

? TRUE : FALSE ;

}

inline BOOL IsValidChanNumericPidx12(LONG Pidx)

{

return IsValidChanNumericPidx(Pidx);

}

inline BOOL IsValidChanNumericPidx15(LONG Pidx)

{

return ( PIDXChan\_AsyncMode <= Pidx && Pidx <= PIDXChan\_DepositStatus )

? TRUE : FALSE ;

}

inline BOOL IsValidChanCapPidx(LONG Pidx)

{

return ( PIDXChan\_CapDiscrepancy <= Pidx &&

Pidx <= PIDXChan\_CapNearFullSensor )

? TRUE : FALSE ;

}

inline BOOL IsValidChanCapPidx12(LONG Pidx)

{

return IsValidChanCapPidx(Pidx);

}

inline BOOL IsValidChanCapPidx15(LONG Pidx)

{

return ( PIDXChan\_CapDiscrepancy <= Pidx &&

Pidx <= PIDXChan\_CapRepayDeposit )

? TRUE : FALSE ;

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

const LONG PIDXChan\_CurrencyCashList = 1 + PIDX\_CHAN+PIDX\_STRING;

const LONG PIDXChan\_CurrencyCode = 2 + PIDX\_CHAN+PIDX\_STRING;

const LONG PIDXChan\_CurrencyCodeList = 3 + PIDX\_CHAN+PIDX\_STRING;

const LONG PIDXChan\_ExitCashList = 4 + PIDX\_CHAN+PIDX\_STRING;

// Added in Release 1.5

const LONG PIDXChan\_DepositCashList = 5 + PIDX\_CHAN+PIDX\_STRING;

const LONG PIDXChan\_DepositCodeList = 6 + PIDX\_CHAN+PIDX\_STRING;

const LONG PIDXChan\_DepositCounts = 7 + PIDX\_CHAN+PIDX\_STRING;

// \* Validation Function \*

inline BOOL IsValidChanStringPidx(LONG Pidx)

{

return ( PIDXChan\_CurrencyCashList <= Pidx &&

Pidx <= PIDXChan\_ExitCashList )

? TRUE : FALSE ;

}

inline BOOL IsValidChanStringPidx12(LONG Pidx)

{

return IsValidChanStringPidx(Pidx);

}

inline BOOL IsValidChanStringPidx15(LONG Pidx)

{

return ( PIDXChan\_CurrencyCashList <= Pidx &&

Pidx <= PIDXChan\_DepositCounts )

? TRUE : FALSE ;

}

#endif // !defined(OPOSCHAN\_HI)

## OposTone.hi : トーンインジケータ内部ヘッダーファイル

//////////////////////////////////////////////////////////////////////

//

// OposTone.hi

//

// Tone Indicator header file for OPOS Controls and Service Objects.

//

// Modification history

// -------------------------------------------------------------------

// 97-06-04 OPOS Release 1.2 CRM

//

//////////////////////////////////////////////////////////////////////

#if !defined(OPOSTONE\_HI)

#define OPOSTONE\_HI

#include "Opos.hi"

#include "OposTone.h"

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXTone\_AsyncMode = 1 + PIDX\_TONE+PIDX\_NUMBER;

const LONG PIDXTone\_Tone1Pitch = 2 + PIDX\_TONE+PIDX\_NUMBER;

const LONG PIDXTone\_Tone1Volume = 3 + PIDX\_TONE+PIDX\_NUMBER;

const LONG PIDXTone\_Tone1Duration = 4 + PIDX\_TONE+PIDX\_NUMBER;

const LONG PIDXTone\_Tone2Pitch = 5 + PIDX\_TONE+PIDX\_NUMBER;

const LONG PIDXTone\_Tone2Volume = 6 + PIDX\_TONE+PIDX\_NUMBER;

const LONG PIDXTone\_Tone2Duration = 7 + PIDX\_TONE+PIDX\_NUMBER;

const LONG PIDXTone\_InterToneWait = 8 + PIDX\_TONE+PIDX\_NUMBER;

// \* Capabilities \*

const LONG PIDXTone\_CapPitch = 501 + PIDX\_TONE+PIDX\_NUMBER;

const LONG PIDXTone\_CapVolume = 502 + PIDX\_TONE+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidToneNumericPidx(LONG Pidx)

{

return ( PIDXTone\_AsyncMode <= Pidx &&

Pidx <= PIDXTone\_InterToneWait )

? TRUE : FALSE ;

}

inline BOOL IsValidToneNumericPidx12(LONG Pidx)

{

return IsValidToneNumericPidx(Pidx);

}

inline BOOL IsValidToneCapPidx(LONG Pidx)

{

return ( PIDXTone\_CapPitch <= Pidx &&

Pidx <= PIDXTone\_CapVolume )

? TRUE : FALSE ;

}

inline BOOL IsValidToneCapPidx12(LONG Pidx)

{

return IsValidToneCapPidx(Pidx);

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Validation Function \*

inline BOOL IsValidToneStringPidx(LONG Pidx)

{

return FALSE ;

}

inline BOOL IsValidToneStringPidx12(LONG Pidx)

{

return IsValidToneStringPidx(Pidx);

}

#endif // !defined(OPOSTONE\_HI)

## OposPpad.hi : PINパッド内部ヘッダーファイル

//////////////////////////////////////////////////////////////////////

//

// OposPpad.hi

//

// PIN Pad header file for OPOS Controls and Service Objects.

//

// Modification history

// -------------------------------------------------------------------

// 98-04-07 OPOS Release 1.3 JDB

// 99-12-07 OPOS Release 1.5 CRM

// Remove Amount property index Amount is of type CURRENCY,

// so it is not accessed by Get/SetPropertyXxx.

// 00-09-24 OPOS Release 1.5 BKS

// Added the Track4Data property

// (01-01-23 Corrected its value from 10 to 12)

//

//////////////////////////////////////////////////////////////////////

#if !defined(OPOSPPAD\_HI)

#define OPOSPPAD\_HI

#include "Opos.hi"

#include "OposPpad.h"

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXPpad\_MaximumPINLength = 1 + PIDX\_PPAD+PIDX\_NUMBER;

const LONG PIDXPpad\_MinimumPINLength = 2 + PIDX\_PPAD+PIDX\_NUMBER;

const LONG PIDXPpad\_PINEntryEnabled = 3 + PIDX\_PPAD+PIDX\_NUMBER;

const LONG PIDXPpad\_Prompt = 4 + PIDX\_PPAD+PIDX\_NUMBER;

const LONG PIDXPpad\_PromptLanguage = 5 + PIDX\_PPAD+PIDX\_NUMBER;

const LONG PIDXPpad\_TransactionType = 6 + PIDX\_PPAD+PIDX\_NUMBER;

// \* Capabilities \*

const LONG PIDXPpad\_CapDisplay = 501 + PIDX\_PPAD+PIDX\_NUMBER;

const LONG PIDXPpad\_CapKeyboard = 502 + PIDX\_PPAD+PIDX\_NUMBER;

const LONG PIDXPpad\_CapLanguage = 503 + PIDX\_PPAD+PIDX\_NUMBER;

const LONG PIDXPpad\_CapMACCalculation = 504 + PIDX\_PPAD+PIDX\_NUMBER;

const LONG PIDXPpad\_CapTone = 505 + PIDX\_PPAD+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidPpadNumericPidx(LONG Pidx)

{

return ( PIDXPpad\_MaximumPINLength <= Pidx &&

Pidx <= PIDXPpad\_TransactionType )

? TRUE : FALSE ;

}

inline BOOL IsValidPpadNumericPidx13(LONG Pidx)

{

return IsValidPpadNumericPidx(Pidx);

}

inline BOOL IsValidPpadCapPidx(LONG Pidx)

{

return ( PIDXPpad\_CapDisplay <= Pidx &&

Pidx <= PIDXPpad\_CapTone )

? TRUE : FALSE ;

}

inline BOOL IsValidPpadCapPidx13(LONG Pidx)

{

return IsValidPpadCapPidx(Pidx);

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXPpad\_AccountNumber = 1 + PIDX\_PPAD+PIDX\_STRING;

const LONG PIDXPpad\_AdditionalSecurityInformation

= 2 + PIDX\_PPAD+PIDX\_STRING;

const LONG PIDXPpad\_AvailableLanguagesList

= 3 + PIDX\_PPAD+PIDX\_STRING;

// (Unused) 4 + PIDX\_PPAD+PIDX\_STRING;

const LONG PIDXPpad\_AvailablePromptsList= 5 + PIDX\_PPAD+PIDX\_STRING;

const LONG PIDXPpad\_EncryptedPIN = 6 + PIDX\_PPAD+PIDX\_STRING;

const LONG PIDXPpad\_MerchantID = 7 + PIDX\_PPAD+PIDX\_STRING;

const LONG PIDXPpad\_TerminalID = 8 + PIDX\_PPAD+PIDX\_STRING;

const LONG PIDXPpad\_Track1Data = 9 + PIDX\_PPAD+PIDX\_STRING;

const LONG PIDXPpad\_Track2Data = 10 + PIDX\_PPAD+PIDX\_STRING;

const LONG PIDXPpad\_Track3Data = 11 + PIDX\_PPAD+PIDX\_STRING;

// Added in Release 1,5

const LONG PIDXPpad\_Track4Data = 12 + PIDX\_PPAD+PIDX\_STRING;

// \* Validation Functions \*

inline BOOL IsValidPpadStringPidx(LONG Pidx)

{

return ( PIDXPpad\_AccountNumber <= Pidx &&

Pidx <= PIDXPpad\_Track3Data )

? TRUE : FALSE ;

}

inline BOOL IsValidPpadStringPidx13(LONG Pidx)

{

return IsValidPpadStringPidx(Pidx);

}

inline BOOL IsValidPpadStringPidx15(LONG Pidx)

{

return ( PIDXPpad\_AccountNumber <= Pidx &&

Pidx <= PIDXPpad\_Track4Data )

? TRUE : FALSE ;

}

#endif // !defined(OPOSPPAD\_HI)

## OposCat.hi : CAT内部ヘッダーファイル

/////////////////////////////////////////////////////////////////////

//

// OposCat.hi

//

// CAT header file for OPOS Controls and Service Objects.

//

// Modification history

// ------------------------------------------------------------------

// 98-06-01 OPOS Release 1.4 OPOS-J

// 00-09-16 OPOS Release 1.5 BKS

//

/////////////////////////////////////////////////////////////////////

#if !defined(OPOSCAT\_HI)

#define OPOSCAT\_HI

#include "Opos.hi"

#include "OposCat.h"

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXCat\_AsyncMode = 1 + PIDX\_CAT+PIDX\_NUMBER;

const LONG PIDXCat\_TrainingMode = 2 + PIDX\_CAT+PIDX\_NUMBER;

// Changed in Release 1.5: In 1.4 TransactionType was incorrectly

// identified as a String property

const LONG PIDXCat\_TransactionType = 3 + PIDX\_CAT+PIDX\_NUMBER;

// Added in Release 1.5

const LONG PIDXCat\_PaymentMedia = 4 + PIDX\_CAT+PIDX\_NUMBER;

// \* Capabilities \*

const LONG PIDXCat\_CapAdditionalSecurityInformation

= 501 + PIDX\_CAT+PIDX\_NUMBER;

const LONG PIDXCat\_CapAuthorizeCompletion

= 502 + PIDX\_CAT+PIDX\_NUMBER;

const LONG PIDXCat\_CapAuthorizePreSales = 503 + PIDX\_CAT+PIDX\_NUMBER;

const LONG PIDXCat\_CapAuthorizeRefund = 504 + PIDX\_CAT+PIDX\_NUMBER;

const LONG PIDXCat\_CapAuthorizeVoid = 505 + PIDX\_CAT+PIDX\_NUMBER;

const LONG PIDXCat\_CapAuthorizeVoidPreSales

= 506 + PIDX\_CAT+PIDX\_NUMBER;

const LONG PIDXCat\_CapCenterResultCode = 507 + PIDX\_CAT+PIDX\_NUMBER;

const LONG PIDXCat\_CapCheckCard = 508 + PIDX\_CAT+PIDX\_NUMBER;

const LONG PIDXCat\_CapDailyLog = 509 + PIDX\_CAT+PIDX\_NUMBER;

const LONG PIDXCat\_CapInstallments = 510 + PIDX\_CAT+PIDX\_NUMBER;

const LONG PIDXCat\_CapPaymentDetail = 511 + PIDX\_CAT+PIDX\_NUMBER;

const LONG PIDXCat\_CapTaxOthers = 512 + PIDX\_CAT+PIDX\_NUMBER;

const LONG PIDXCat\_CapTransactionNumber = 513 + PIDX\_CAT+PIDX\_NUMBER;

const LONG PIDXCat\_CapTrainingMode = 514 + PIDX\_CAT+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidCatNumericPidx(LONG Pidx)

{

return ( PIDXCat\_AsyncMode <= Pidx &&

Pidx <= PIDXCat\_TransactionType )

? TRUE : FALSE ;

}

inline BOOL IsValidCatNumericPidx14(LONG Pidx)

{

return IsValidCatNumericPidx(Pidx);

}

inline BOOL IsValidCatNumericPidx15(LONG Pidx)

{

return ( PIDXCat\_AsyncMode <= Pidx &&

Pidx <= PIDXCat\_PaymentMedia )

? TRUE : FALSE ;

}

inline BOOL IsValidCatCapPidx(LONG Pidx)

{

return ( PIDXCat\_CapAdditionalSecurityInformation <= Pidx &&

Pidx <= PIDXCat\_CapTrainingMode )

? TRUE : FALSE ;

}

inline BOOL IsValidCatCapPidx14(LONG Pidx)

{

return IsValidCatCapPidx(Pidx);

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXCat\_AccountNumber = 1 + PIDX\_CAT+PIDX\_STRING;

const LONG PIDXCat\_AdditionalSecurityInformation

= 2 + PIDX\_CAT+PIDX\_STRING;

const LONG PIDXCat\_ApprovalCode = 3 + PIDX\_CAT+PIDX\_STRING;

const LONG PIDXCat\_CardCompanyID = 4 + PIDX\_CAT+PIDX\_STRING;

const LONG PIDXCat\_CenterResultCode = 5 + PIDX\_CAT+PIDX\_STRING;

const LONG PIDXCat\_DailyLog = 6 + PIDX\_CAT+PIDX\_STRING;

const LONG PIDXCat\_PaymentCondition = 7 + PIDX\_CAT+PIDX\_STRING;

const LONG PIDXCat\_PaymentDetail = 8 + PIDX\_CAT+PIDX\_STRING;

const LONG PIDXCat\_SequenceNumber = 9 + PIDX\_CAT+PIDX\_STRING;

const LONG PIDXCat\_SlipNumber = 10 + PIDX\_CAT+PIDX\_STRING;

const LONG PIDXCat\_TransactionNumber = 11 + PIDX\_CAT+PIDX\_STRING;

// Changed in Release 1.5: In 1.4 TransactionType was incorrectly

// identified as a String property

//const LONG PIDXCat\_TransactionType = 12 + PIDX\_CAT+PIDX\_STRING;

// \* Validation Functions \*

inline BOOL IsValidCatStringPidx(LONG Pidx)

{

return ( PIDXCat\_AccountNumber <= Pidx &&

Pidx <= PIDXCat\_TransactionNumber )

? TRUE : FALSE ;

}

inline BOOL IsValidCatStringPidx14(LONG Pidx)

{

return IsValidCatStringPidx(Pidx);

}

#endif // !defined(OPOSCAT\_HI)

## OposPcrw.hi : ポイントカード機内部ヘッダーファイル

/////////////////////////////////////////////////////////////////////

//

// OposPcrw.hi

//

// Point Card Reader Writer header file for OPOS Controls and

// Service Objects.

//

// Modification history

// ------------------------------------------------------------------

// 00-09-24 OPOS Release 1.5 BKS

// 02-08-17 OPOS Release 1.7 CRM

// Added MapCharacterSet property.

// Added CapMapCharacterSet capability.

//

/////////////////////////////////////////////////////////////////////

#if !defined(OPOSPCRW\_HI)

#define OPOSPCRW\_HI

#include "Opos.hi"

#include "OposPcrw.h"

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXPcrw\_CardState = 1 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CharacterSet = 2 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_LineChars = 3 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_LineHeight = 4 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_LineSpacing = 5 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_LineWidth = 6 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_MapMode = 7 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_MaxLine = 8 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_PrintHeight = 9 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_ReadState1 = 10 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_ReadState2 = 11 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_RecvLength1 = 12 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_RecvLength2 = 13 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_SidewaysMaxChars = 14 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_SidewaysMaxLines = 15 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_TracksToRead = 16 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_TracksToWrite = 17 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_WriteState1 = 18 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_WriteState2 = 19 + PIDX\_PCRW+PIDX\_NUMBER;

// Added in Release 1.7

const LONG PIDXPcrw\_MapCharacterSet = 20 + PIDX\_PCRW+PIDX\_NUMBER;

// \* Capabilities \*

const LONG PIDXPcrw\_CapBold = 501 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CapCardEntranceSensor = 502 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CapCharacterSet = 503 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CapCleanCard = 504 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CapClearPrint = 505 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CapDhigh = 506 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CapDwide = 507 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CapDwideDhigh = 508 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CapItalic = 509 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CapLeft90 = 510 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CapPrint = 511 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CapPrintMode = 512 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CapRight90 = 513 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CapRotate180 = 514 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CapTracksToRead = 515 + PIDX\_PCRW+PIDX\_NUMBER;

const LONG PIDXPcrw\_CapTracksToWrite = 516 + PIDX\_PCRW+PIDX\_NUMBER;

// Added in Release 1.7

const LONG PIDXPcrw\_CapMapCharacterSet = 517 + PIDX\_PCRW+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidPcrwNumericPidx(LONG Pidx)

{

return ( PIDXPcrw\_CardState <= Pidx &&

Pidx <= PIDXPcrw\_WriteState2 )

? TRUE : FALSE ;

}

inline BOOL IsValidPcrwNumericPidx15(LONG Pidx)

{

return IsValidPcrwNumericPidx(Pidx);

}

inline BOOL IsValidPcrwNumericPidx17(LONG Pidx)

{

return ( PIDXPcrw\_CardState <= Pidx &&

Pidx <= PIDXPcrw\_MapCharacterSet )

? TRUE : FALSE ;

}

inline BOOL IsValidPcrwCapPidx(LONG Pidx)

{

return ( PIDXPcrw\_CapBold <= Pidx &&

Pidx <= PIDXPcrw\_CapTracksToWrite )

? TRUE : FALSE ;

}

inline BOOL IsValidPcrwCapPidx15(LONG Pidx)

{

return IsValidPcrwCapPidx(Pidx);

}

inline BOOL IsValidPcrwCapPidx17(LONG Pidx)

{

return ( PIDXPcrw\_CapBold <= Pidx &&

Pidx <= PIDXPcrw\_CapMapCharacterSet )

? TRUE : FALSE ;

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXPcrw\_CharacterSetList = 1 + PIDX\_PCRW+PIDX\_STRING;

const LONG PIDXPcrw\_FontTypeFaceList = 2 + PIDX\_PCRW+PIDX\_STRING;

const LONG PIDXPcrw\_LineCharsList = 3 + PIDX\_PCRW+PIDX\_STRING;

const LONG PIDXPcrw\_Track1Data = 4 + PIDX\_PCRW+PIDX\_STRING;

const LONG PIDXPcrw\_Track2Data = 5 + PIDX\_PCRW+PIDX\_STRING;

const LONG PIDXPcrw\_Track3Data = 6 + PIDX\_PCRW+PIDX\_STRING;

const LONG PIDXPcrw\_Track4Data = 7 + PIDX\_PCRW+PIDX\_STRING;

const LONG PIDXPcrw\_Track5Data = 8 + PIDX\_PCRW+PIDX\_STRING;

const LONG PIDXPcrw\_Track6Data = 9 + PIDX\_PCRW+PIDX\_STRING;

const LONG PIDXPcrw\_Write1Data = 10 + PIDX\_PCRW+PIDX\_STRING;

const LONG PIDXPcrw\_Write2Data = 11 + PIDX\_PCRW+PIDX\_STRING;

const LONG PIDXPcrw\_Write3Data = 12 + PIDX\_PCRW+PIDX\_STRING;

const LONG PIDXPcrw\_Write4Data = 13 + PIDX\_PCRW+PIDX\_STRING;

const LONG PIDXPcrw\_Write5Data = 14 + PIDX\_PCRW+PIDX\_STRING;

const LONG PIDXPcrw\_Write6Data = 15 + PIDX\_PCRW+PIDX\_STRING;

// \* Validation Functions \*

inline BOOL IsValidPcrwStringPidx(LONG Pidx)

{

return ( PIDXPcrw\_CharacterSetList <= Pidx &&

Pidx <= PIDXPcrw\_Write6Data )

? TRUE : FALSE ;

}

inline BOOL IsValidPcrwStringPidx15(LONG Pidx)

{

return IsValidPcrwStringPidx(Pidx);

}

#endif // !defined(OPOSPCRW\_HI)

## OposPwr.hi : パワーマネージメント内部ヘッダファイル

/////////////////////////////////////////////////////////////////////

//

// OposPwr.hi

//

// POS Power header file for OPOS Controls and Service Objects.

//

// Modification history

// ------------------------------------------------------------------

// 00-09-16 OPOS Release 1.5 BKS

//

/////////////////////////////////////////////////////////////////////

#if !defined(OPOSPWR\_HI)

#define OPOSPWR\_HI

#include "Opos.hi"

#include "OposPwr.h"

//////////////////////////////////////////////////////////////////////

// Numeric Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Properties \*

const LONG PIDXPwr\_EnforcedShutdownDelayTime

= 1 + PIDX\_PWR+PIDX\_NUMBER;

const LONG PIDXPwr\_PowerFailDelayTime = 2 + PIDX\_PWR+PIDX\_NUMBER;

const LONG PIDXPwr\_QuickChargeMode = 3 + PIDX\_PWR+PIDX\_NUMBER;

const LONG PIDXPwr\_QuickChargeTime = 4 + PIDX\_PWR+PIDX\_NUMBER;

const LONG PIDXPwr\_UPSChargeState = 5 + PIDX\_PWR+PIDX\_NUMBER;

// \* Capabilities \*

const LONG PIDXPwr\_CapFanAlarm = 501 + PIDX\_PWR+PIDX\_NUMBER;

const LONG PIDXPwr\_CapHeatAlarm = 502 + PIDX\_PWR+PIDX\_NUMBER;

const LONG PIDXPwr\_CapQuickCharge = 503 + PIDX\_PWR+PIDX\_NUMBER;

const LONG PIDXPwr\_CapShutdownPOS = 504 + PIDX\_PWR+PIDX\_NUMBER;

const LONG PIDXPwr\_CapUPSChargeState = 505 + PIDX\_PWR+PIDX\_NUMBER;

// \* Validation Functions \*

inline BOOL IsValidPwrNumericPidx(LONG Pidx)

{

return ( PIDXPwr\_EnforcedShutdownDelayTime <= Pidx &&

Pidx <= PIDXPwr\_UPSChargeState )

? TRUE : FALSE ;

}

inline BOOL IsValidPwrNumericPidx14(LONG Pidx)

{

return IsValidPwrNumericPidx(Pidx);

}

inline BOOL IsValidPwrCapPidx(LONG Pidx)

{

return ( PIDXPwr\_CapFanAlarm <= Pidx &&

Pidx <= PIDXPwr\_CapUPSChargeState )

? TRUE : FALSE ;

}

inline BOOL IsValidPwrCapPidx15(LONG Pidx)

{

return IsValidPwrCapPidx(Pidx);

}

//////////////////////////////////////////////////////////////////////

// String Property Index Values.

//////////////////////////////////////////////////////////////////////

// \* Validation Functions \*

inline BOOL IsValidPwrStringPidx(LONG Pidx)

{

return FALSE ;

}

inline BOOL IsValidPwrStringPidx15(LONG Pidx)

{

return IsValidPwrStringPidx(Pidx);

}

#endif // !defined(OPOSPWR\_HI)

# 付録C 修正履歴

## 第1.11版

第1.11版は、米国OLE for Retail POS Committeeにおいて標準化された

“Control Programmer’s Guide Release 1.1”との整合性確保（コインディスペンサ，MICR， 秤，シグニチャ・キャプチャの4デバイス追加を含む）と第1.0版の誤記の修正を行ったものです。

以下に第1.11版にて修正・削除された項目について記述します。

|  |  |
| --- | --- |
| 章または節 | 修正内容 |
| 表紙 | 日本拡張仕様書を日本版仕様書に名称変更 |
| はしがき | 著作権の記述を修正 |
| はじめに，全体 | Application Programmer’s Guide 日本拡張仕様書をApplication Programmer’s Guide 日本版仕様書に名称変更 |
| はじめに | “Coin Dispenser” “MICR” “Scale” “Signature Capture”が含まれない説明を削除 |
| はじめに，イベント | 作業スレッドをワーカスレッドに修正 |
| 付録A | “OPOSCOIN.HI” “OPOSMICR.HI” “OPOSSCALE.HI” “OPOSSIG.HI”を新規追加 |

## 第1.2版

第1.2版は、デバイスクラスの追加とＯＰＯＳ米国，ＯＰＯＳ日本およびＯＰＯＳヨーロッパの各メンバからの各種要求をベースとするＡＰＩを追加しました。

第1.2版は、第1.11版のスーパーセットに位置付けられます。

以下に第1.2版にて修正・削除された項目について記述します。

|  |  |
| --- | --- |
| 章または節 | 修正内容 |
| はしがき | 企業名および著作権の記述を修正 |
| はじめに | “標準化に...日本のPOSシステム...” を削除  OLE for Retail POSコントロール概要の図を更新  備考の内容をアウトプロセスとインプロセスのサービスオブジェクトについての議論に差し替え |
| コントロールオブジェクトの役割とインプリメンテーション | コントロールオブジェクトとサービスオブジェクト間でバージョン不一致の場合の内部処理を追加  **SOProcessID**メソッドの追加 |
| Opos.hiヘッダーファイル | “CashChanger” “ToneIndicator”を追加  **AutoDisable** **BinaryConvertion**  **DataCount** プロパティを追加 |
| OposMsr.hiヘッダーファイル | **ErrorReportingType** プロパティを追加  **ParseDecodedData**プロパティを追加、値は**ParseDecodeData**プロパティに同じ |
| OposKbd.hiヘッダーファイル | **CapKeyUp EventTypes POSKeyEventType** プロパティを追加 |
| OposScal.hiヘッダーファイル | **CapDisplay WeightUnit** プロパティを追加 |
| OposScan.hiヘッダーファイル | **ScanDataLabel ScanDataType** プロパティを追加 |
| OposSig.hiヘッダーファイル | **CapRealTimeData RealTimeDataEnabled** プロパティを追加 |
| 付録A | “OposChan.hi” “OposTone.hi”を新規追加 |

## 第1.3版

第1.3版は、デバイスクラスの追加とわずかなAPIの追加およびいくつかの修正をしました。

第1.3版は、第1.2版のスーパーセットに位置付けられます。

以下に第1.3版にて修正・削除された項目について記述します。

|  |  |
| --- | --- |
| 章または節 | 修正内容 |
| はしがき | 企業名および著作権の記述を修正  Webサイト参照の情報を追加 |
| 全体的に | イベント”発行”という単語の使い方を一部変更  イベント発行の記述を”キューイング”と”発行”を使ってそれぞれ対応する記述に変更 |
| コントロールオブジェクトの役割とインプリメンテーション | **SOError**：コントロールがサポートしている場合入力エラーの戻り時に OPOS\_ER\_RETRYを指定可能とした |
| サービスオブジェクトの役割とインプリメンテーション | 文字列型でも４バイト数値型でもない型のプロパティ用メソッドの記述を追加 |
| Opos.hiヘッダーファイル | ”Fiscal Printer” “PIN Pad” “Remote Order Display” “Bump Bar”を追加  **CapPowerReporting** **PowerNotify**  **PowerState** プロパティを追加 |
| OposChan.hiヘッダーファイル | 文字列型プロパティがPIDX\_NUMBERで定義されていたものをPIDX\_STRINGに修正 |
| OposScal.hiヘッダーファイル | **CapDisplayText CapPriceCalculating CapTareWeight CapZeroScalw AsyncMode MaxDisplayTextChars TareWeight** プロパティを追加 |
| いくつかのヘッダーファイル | バリデーション関数を追加 |
| 付録A | “OposFptr.hi” “OposPpad.hi” “OposRod.hi” “OposBb.hi”を新規追加 |

## 第1.4版

第1.4版は、デバイスクラスの追加をしました。

第1.4版は、第1.3版のスーパーセットに位置付けられます。

以下に第1.4版にて修正・削除された項目について記述します。

|  |  |  |
| --- | --- | --- |
| 章または節 | | 修正内容 |
| 付録A | “OposCat.hi” を新規追加 |
| Opos.hiヘッダーファイル | “CAT”を追加 |

## 第1.5版

第1.5版は、2つのデバイスクラスが追加されました。

第1.5版は、第1.4版のスーパーセットに位置付けられます。

以下に第1.5版にて修正・削除された項目について記述します。

|  |  |
| --- | --- |
| 章または節 | 修正内容 |
| はしがき | 著作権の記述を修正  Webサイト参照を修正 |
| 全体的に | **Claim**と**Release**のレファレンスに**ClaimDevice**と**ReleaseDevice**情報を追加  適宜OLEをActiveXに修正  MFCの実装を一般化し、ATLの実装情報を追加 |
| コントロールオブジェクトの役割 | 詳細な実装を削除し、コモンコントロールオブジェクトの説明追加 |
| サービスオブジェクトの**GetOpenResult**メソッド | 新規メソッドを追加 |
| Opos.hiヘッダーファイル | ポイントカード機とパワーマネージメントデバイスカテゴリを追加 |
| OposCash.hiヘッダファイル | **CapMultiDrawerDetect**プロパティを追加 |
| OposCat.hiヘッダファイル | **PaymentMedia**プロパティを追加 |
| OposCash.hiヘッダファイル | **DepositAmount**、**DepositStatus**、**DeviceStatus, CapDeposit**、**CapDepositDataEvent**、**CapPauseDeposit**、**CapRepayDeposit**、**DepositCashList**、**DepositCodeList**、**DepositCounts**プロパティを追加 |
| OposMSR.hiヘッダファイル | **CapTransmitSentinels**、**Track4Data**、**TransmitSentinels**プロパティを追加 |
| OposPcrw.hiヘッダーファイル | ポイントカード機の新規ヘッダファイル |
| OposPpad.hiヘッダーファイル | 1.3版に適合するように修正  Amountプロパティインデックスを削除－文字列ではない  **Track4Data**プロパティを追加 |
| OposPtr.hiヘッダーファイル | **CapJrnCartridgeSensor**、**CapJrnColor**、**CapRecCartrdigeSensor**、**CapRecColor**、**CapRecMarkFeed**、**CapSlpBothSidesPrint**、**CapSlpCartridgeSensor**、**CapSlpColor**、**CartridgeNotify**、**JrnCartridgeState**、**JrnCurrentCartridge**、**RecCartridgeState**、**RecCurrentCartridge**、**SlpPrintSide**、**SlpCartridgeState**、**SlpCurrentCartridge**プロパティを追加 |
| OposPwr.hiヘッダーファイル | パワーマネージメントの新規ヘッダファイル |

## 第1.6版

## 第1.7版

第1.7版は、デバイスクラスの追加とわずかなAPIの追加およびいくつかの修正をしました。

第1.7版は、第1.6版のスーパーセットに位置付けられます。

以下に第1.7版にて修正・削除された項目について記述します。

|  |  |  |
| --- | --- | --- |
| 章または節 | | 修正内容 |
| はしがき | 基本となる仕様書をUnifiedPOSに修正  著作権の記述を修正 |
| はじめに | 基本となる仕様書をUnifiedPOSに修正  サポートするOSにMicrosoft Windows®XP追加 |
| コントロールオブジェクトのプロパティ | 新デバイス提案時のデータファイルについての記述追加 |
| サービスオブジェクトのClerOutputメソッド | 非同期出力のバッファリングされているデータもクリアされることの説明追加 |
| サービスオブジェクトのCOFreezeEventsメソッド | “アーキテクチャの問題：コンテナによるイベント凍結”参照を追加 |
| 付録B | Check Scannerを新規追加  Motion Sensorを新規追加  Webサイト参照を追加 |
| Opos.hiヘッダーファイル | チェックスキャナーとモーションセンサーデバイスカテゴリを追加 |
| OposDisp.hiヘッダーファイル | **BlinkRate**、**CursorType**、**CustomSSGlyphList**、**GlyphHeight**、**GlyphWidth**、**CapBlinkRate**、**CapCursorType**、**CapCustomGlyph**、**CapReadBack**、**CapReverse**プロパティ（第1.6版）を追加  **MapCharacterSet**、**MaximumX**、**MaximumY**、**ScreenMode**、**ScreenModeList**、**CapBitmap**、**CapMapCharacterSet**、**CapScreenMode**プロパティ追加 |
| OposMicr.hiヘッダーファイル | MICR（磁気インクリーダ）内部ヘッダーファイル削除 |
| OposSig.hiヘッダーファイル | シグニチャキャプチャ内部ヘッダーファイル削除 |
| OposFptr.hiヘッダーファイル | フィスカルプリンタ内部ヘッダーファイル削除 |
| OposPpad.hiヘッダーファイル | **Track4Data**プロパティ定数重複誤り修正 |
| OposRod.hiヘッダーファイル | リモートオーダディスプレイ内部ヘッダーファイル削除 |
| OposBb.hiヘッダーファイル | バンプバー内部ヘッダーファイル削除 |
| OposPcrw.hiヘッダーファイル | **MapCharacterSet**、**CapMapCharacterSet**プロパティ追加 |