OpenPOS

Application Programmer's Guide

日本版仕様書 第1.16.1版

2025年 10月 28日

第1章－第47章・付録

OpenPOS技術協議会

OpenPOS

はしがき

本仕様書は、米国OMG UnifiedPOS Technical Committeeにおいて標準化された“Unified POS Retail Peripheral Architecture Version 1.16.1”に準拠して作成されています。

本仕様書の著作権は、OpenPOS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、OpenPOS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、OpenPOS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。

“Application 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。

© 1995-2005 Fujitsu Transaction Solutions Inc. All rights reserved.
© 1995-2005 Microsoft Corporation. All rights reserved.
© 1995-2005 NCR Corporation. All rights reserved.
© 1995-2005 Seiko Epson Corporation. All rights reserved.

*Microsoft, Windows, Windows-NT, Windows 2000, WindowsXPおよびMicrosoft WindowsVista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。

目次

第 1 章 （ OLE 版 ） はじめに 15

OPENPOSコントロール 15
OpenPOSとは？ 15
マニュアル対象者 16
OPENPOSコントロール概要 17
OPOS関連用語の説明 18
OPOSコントロール使用方法 20
初期化とエラー通知 第1.11版にて追加 22
メソッドとプロパティの使い方 26
ステータス、結果コード、状態モデル 28
ステータスモデル 29
結果コードモデル 29
状態モデル 第1.7版にて更新 30
デバイス共有モデルについて 32
排他使用デバイス 32
共有可能デバイス 33
イベント 34
入力モデルについて 第1.13版にて更新 37
入力データの順序について 40
出力モデルについて 第1.13版にて更新 41
同期出力 41
非同期出力 41
デバイス電源通知モデルについて 42
モデル 42
プロパティ 44
DeviceEnabledプロパティと電源通知機能 45
デバイスインフォメーションレポーティングモデルについて 46
スタティスティクスレポーティングプロパティとメソッド 47
POSデバイススタティスティクスのXML定義 48
アップデートファームウェアデバイスモデルについて 51
廃止予定の扱い 第1.11版にて追加 52
ハイドラデバイスの検討 第1.1 2版にて更新 53
リテールコミュニケーションサービスデバイス 第1.16版にて追加 58
OPOSコントロールの解説 60
章の構成 60
データの型 62

第 1 章 （ .NET 版 ） はじめに 64

POS FOR .NETとは? 第1.11版にて更新 64
マニュアル対象者 65
POS FOR .NETの概説 66
POS FOR .NET関連用語の説明 68

デバイスクラス 68
サービスオブジェクト（SO） 68
POS FOR .NETの主要な機能 68
POS周辺機器の.NETインターフェイス 68
サービスオブジェクトのBase class 68
サービスオブジェクトのBasic class 69
プラグアンドプレイ 70
標準化されたセットアップ 70
デバイスの列挙 70
ソフトウェアベースのデバイススタティスティクス 70
OPOS（COMベースの）サービスオブジェクトのサポート 70
サービスオブジェクト検証プログラム 71
プログラミング構築におけるOPOSとの主な差異 71
命名規則 71
列挙体 72
構造体 91
提供されるクラスライブラリの一覧 96
戻り値 96
返されるプロパティ 97
返されるリスト 97
パラメータの主な違い 99
プロパティシグネチャの主な違い 100
関連情報 101
POSEXPLORER API 101
PosExplorerプロパティ 102
PosExplorerメソッド 102
PosExplorerイベント 105
グローバル設定 107
サービスオブジェクトレジストリ 107
サービスオブジェクトの利用 107
OPOS 107
POS for .NET 109
サービスオブジェクトの制作 109
POS for .NET 109
ステータス、状態モデル、および例外 110
StatusUpdateEvent 110
ControlState 110
例外 110
デバイス共有モデルについて 111
排他使用デバイス 112
共有可能デバイス 112
イベント 112
入力モデルについて 114
出力モデルについて 116
同期出力 116
非同期出力 116
デバイス電源通知モデルについて 117

モデル.....*118*
電源通知プロパティ.....119
DeviceEnabledプロパティと電源通知機能.....*120*
デバイスインフォメーションレポーティングモデルについて.....120
スタティスティクスレポーティングプロパティとメソッド.....*121*
POS FOR .NETコンポーネントの解説.....122
POS for .NETのデータの型.....*122*
POS for .NETの共通プロパティ、メソッド、イベント、スタティスティクス、定数.....*123*

第 2 章 （ ＯＬＥ版 ） 共通プロパティ、メソッド、イベント.....126

一覧.....126
概説.....128
プロパティ.....129
メソッド.....155
イベント.....175

第 2 章 （ .ＮＥＴ版 ） 共通プロパティ、メソッド、イベント.....183

プロパティ.....183
メソッド.....193
イベント.....210
POS FOR .NETのメンバとUNIFIEDPOSのメンバ.....211
レガシーなOPOSサービスのために使用可能な暫定的な手順... SHIM コードの使用方法 第1.11版にて更新.....212
アーキテクチャ図 第1.11版にて追加.....*214*
実装の方法.....*215*
管理方法.....*218*
クラス図.....*222*

第 3 章 ドロワー.....225

一覧.....225
概説.....227
プロパティ.....228
メソッド.....230
イベント.....232

第 4 章 ラインディスプレイ.....233

一覧.....233
概説.....237
プロパティ.....244
メソッド.....278
補足説明.....302
ウィンドウとビューポートについて.....*302*
マーキースクロールについて.....*308*

カーソル、文字列表示とスクロールの関係について 312

第 5 章 ハードトータル 315

一覧 315
概説 319
プロパティ 324
メソッド 327

第 6 章 キーロック 346

一覧 346
概説 348
プロパティ 350
メソッド 353
イベント 355

第 7 章 磁気ストライプリーダ 357

一覧 357
概説 361
プロパティ 369
メソッド 408
イベント 417

第 8 章 POSプリンタ 422

一覧 422
概説 428
プロパティ 450
メソッド 526
イベント 593

第 9 章 スキャナ (バーコードリーダ) 601

一覧 601
概説 603
プロパティ 605
イベント 615

第 10 章 POSキーボード 617

一覧 617
概説 619
プロパティ 622
イベント 624

第 11 章 コインディスペンサ 627

一覧 627
概説 629
プロパティ 630
メソッド 632
イベント 634

第 12 章 MICR (磁気インクリーダ) 635

一覧 635
概説 638

第 13 章 秤 639

一覧 639
概説 642
プロパティ 646
メソッド 656
イベント 660

第 14 章 シグニチャ・キャプチャ 665

一覧 665
概説 667

第 15 章 自動釣り銭機 669

一覧 669
概説 672
プロパティ 679
メソッド 695
イベント 705

第 16 章 トーンインジケータ 707

一覧 707
概説 710
プロパティ 715
メソッド 723

第 17 章 フィスカルプリンタ 725

一覧 725
概説 733

第 18 章 PINパッド 735

一覧 735
概説 738
プロパティ 742
メソッド 760
イベント 767

第 19 章 リモートオーダディスプレイ 771

一覧 771
概説 775

第 20 章 バンプバー 777

一覧 777
概説 780

第　２１章　　CAT (CREDIT AUTHORIZATION TERMINAL)　781

一覧　781
概説　784
プロパティ　794
メソッド　820
イベント　833

第　２２章　　ポイントカード機　836

一覧　836
概説　840
印字データとエスケープシーケンス　847
プロパティ　850
メソッド　880
イベント　893

第　２３章　　パワーマネージメント　897

一覧　897
概説　900
プロパティ　903
メソッド　916
イベント　920

第　２４章　　スマートカードリーダライタ　923

一覧　923
概説　926
プロパティ　935
メソッド　944
イベント　954

第　２５章　　バイオメトリクス　959

一覧　959
概説　962
プロパティ　966
メソッド　977
イベント　988

第　２６章　　電子ジャーナル　993

一覧　993
概説　996
プロパティ　999
メソッド　1010
イベント　1026

第　２７章　　モーションセンサ　1033

一覧　1033
概説　1037

第 28 章 紙幣入金機 1039

一覧 1039
概説 1042
プロパティ 1045
メソッド 1050
イベント 1056

第 29 章 硬貨入金機 1057

一覧 1057
概説 1060
プロパティ 1063
メソッド 1071
イベント 1078

第 30 章 紙幣出金機 1079

一覧 1079
概説 1082
プロパティ 1085
メソッド 1091
イベント 1095

第 31 章 イメージスキャナ 1097

一覧 1097
概説 1100
プロパティ 1103
メソッド 1114
イベント 1115

第 32 章 電子バリューリーダライタ 1117

一覧 1117
概説 1122
プロパティ 1174
メソッド 1221
イベント 1284

第 33 章 RFIDスキャナ 1295

一覧 1295
概説 1298
プロパティ 1302
メソッド 1309
イベント 1322

第 34 章 チェックスキャナ 1325

一覧 1325
概説 1328

第 35 章 ベルト 1329

一覧 1329
概説 1332

第　３６章　　ゲート 1333

一覧 1333
概説 1335

第　３７章　　アイテムディスペンサ 1337

一覧 1337
概説 1339

第　３８章　　ライト 1341

一覧 1341
プロパティ 1344
メソッド 1347

第　３９章　　ビデオキャプチャ 1353

一覧 1353
概説 1357
メソッド 1387
イベント 1391

第　４０章　　個体認識 1395

一覧 1395
概説 1397
プロパティ 1400

第　４１章　　サウンドレコーダ 1409

一覧 1409
概説 1412
プロパティ 1417
メソッド 1425
イベント 1427

第　４２章　　音声認識 1431

一覧 1431
概説 1434
プロパティ 1438
メソッド 1442
イベント 1447

第　４３章　　サウンドプレーヤ 1451

一覧 1451
概説 1454
プロパティ 1457
メソッド 1461
イベント 1463

第 44 章 音声合成 1467

一覧 1467
概説 1470
プロパティ 1473
メソッド 1478
イベント 1481

第 45 章 ジェスチャーコントロール 1484

一覧 1484
概説 1487
プロパティ 1490
メソッド 1496
イベント 1503

第 46 章 デバイスモニタ 1506

一覧 1506
概説 1508
プロパティ 1510
メソッド 1513
イベント 1517

第 47 章 グラフィックディスプレイ 1521

一覧 1521
概説 1524
プロパティ 1529
メソッド 1539
イベント 1545

付 録 A OPOSレジストリの利用法 1549

付 録 B OPOSで定義するヘッダーファイル 1555

OPOS.H： OPOS基本ヘッダーファイル 1557
OPOSBACC.H： 紙幣入金機ヘッダーファイル 1560
OPOSBD.H： バンプバーヘッダーファイル 1561
OPOSBDSP.H： 紙幣出金機ヘッダーファイル 1562
OPOSBELT.H： ベルトヘッダーファイル 1563
OPOSBIO.H： バイオメトリクスヘッダーファイル 1564
OPOSCACC.H： 硬貨入金機ヘッダーファイル 1566
OPOSCASH.H： ドロワーヘッダーファイル 1567
OPOSCAT.H： CATヘッダーファイル 1568
OPOSCHAN.H： 自動釣り銭機ヘッダーファイル 1570
OPOSCHK.H： チェックスキャナヘッダーファイル 1572
OPOSCOIN.H： コインディスペンサヘッダーファイル 1575
OPOSDISP.H： ラインディスプレイヘッダーファイル 1576
OPOSDMON.H： デバイスモニタヘッダーファイル 1579
OPOSEJ.H： 電子ジャーナルヘッダーファイル 1580

OPOSEVRW.H： 電子バリューリーダライタヘッダーファイル 1581
OPOSFPTR.H： フィスカルプリンタヘッダーファイル 1586
OPOSGATE.H： ゲートヘッダーファイル..... 1593
OPOSGCTL.H： ジェスチャーコントロールヘッダーファイル 1594
OPOSGDSP.H： グラフィックディスプレイヘッダーファイル 1595
OPOSIMG.H： イメージスキャナヘッダーファイル 1597
OPOSIRCG.H： 個体認識ヘッダーファイル 1598
OPOSITEM.H： アイテムディスペンサヘッダーファイル 1599
OPOSKBD.H： POSキーボードヘッダーファイル..... 1600
OPOSLGT.H： ライトヘッダーファイル 1601
OPOSLOCK.H： キーロックヘッダーファイル 1603
OPOSMICR.H： MICR(磁気インクリーダ)ヘッダーファイル 1604
OPOSMOTION.H： モーションセンサヘッダーファイル 1605
OPOSMSR.H： 磁気ストライプリーダヘッダーファイル 1606
OPOSPPAD.H： PINパッドヘッダーファイル 1609
OPOSPCRW.H： ポイントカード機ヘッダーファイル 1611
OPOSPTR.H： POSプリンタヘッダーファイル 1613
OPOSPWR.H： パワーマネージメントヘッダーファイル 1620
OPOSROD.H：リモートオーダーディスプレイヘッダーファイル 1622
OPOSRFID.H： RFIDスキャナヘッダーファイル..... 1625
OPOSSCAL.H： 秤ヘッダーファイル 1626
OPOSSCAN.H： スキャナヘッダーファイル 1628
OPOSSCRW.H： スマートカードリーダライタヘッダーファイル 1631
OPOSSIG.H： シグニチャ・キャプチャヘッダーファイル..... 1633
OPOSSPCH.H： 音声合成ヘッダーファイル 1634
OPOSSPLY.H： サウンドプレーヤヘッダーファイル 1635
OPOSSREC.H： サウンドレコーダヘッダーファイル..... 1636
OPOSSTAT.H： スタティスティック名ヘッダーファイル 1637
OPOSTONE.H： トーンインジケータヘッダーファイル..... 1640
OPOSTOT.H： ハードトータルヘッダーファイル 1641
OPOSVCAP.H： ビデオキャプチャヘッダーファイル..... 1642
OPOSVRCG.H： 音声認識ヘッダーファイル..... 1643

付　録　C　　技術的詳細説明 1645

システム文字列(BSTR)..... 1645
キャラクターセットのマッピング　　第1.10版で変更 1648

付　録　D　　第1.5版でのAPI変更：CLAIMDEVICEと RELEASEDEVICE..... 1651

付　録　E　　修正履歴 1653

第1.1.1版 1653
第1.2版 1658
第1.3版 1662
第1.4版 1668
第1.5版 1669

第1.6版 1673
第1.7版 1677
第1.8版 1681
第1.9版 1685
第1.10版 1690
第1.11版 1693
第1.12版 1697
第1.13版 1700
第1.14版 1706
第1.14.1版 1710
第1.15版 1714
第1.16版 1715
第1.16.1版 1718

付 録 F XMLPOS XML POS MAPPING REFERENCE 1721

概要 第１．１４版で更新 1722
XMLPOS要件項目 1722
仕様の範囲外 1722
参照資料 1723
OPOSからXMLに移行に関して 第1.14.1版で更新 1723
XMLPOSへの変更 第1.13版で更新 1724
XMLPOS構成の概要 第1.14版で更新 1724
OPOSのXML要件 1724
ＯＰＯＳメソッドとイベントのXMLPOSへの変換 1725
OPOS同期XML通信 1727
OPOS非同期XML通信 1728
OPOS XMLエラー 1732
デバイスエラーコードとメッセージセベリティコード 1732
メッセージセベリティコード 1732
セベリティエラー対応表 1734
セベリティ状態対応表 1736
OPOS同期XMLエラー 1738
OPOS非同期XMLエラー 1738
XMLPOS共通イベント 1739
OPOS同期XMLイベント 1740
OPOS非同期XMLイベント 1740
ＸＭＬＰＯＳ共通プロパティ 1741
ＸＭＬＰＯＳ共通データ 1742
ＡＲＴＳ共通データ 1743
ＯＰＯＳデバイス 1744
Bill Acceptor 1745
Bill Dispenser 1751
Biometrics 1756
Cash Changer 1762
Cash Drawer 1769
CAT 1773
Check Scanner 1780

Coin Acceptor *1788*
Coin Dispenser *1794*
Electronic Journal *1799*
Electronic Value Reader / Writer *1805*
Hard Totals *1812*
Image Scanner *1819*
Keylock *1825*
Line Display *1830*
MSR *1837*
PIN Pad *1844*
Point Card Reader/Writer *1850*
POS Keyboard *1859*
POS Power *1863*
POS Printer *1869*
RFID Scanner *1884*
Scanner Device *1891*
Smart Card Reader *1896*
Tone Indicator *1904*
NAFEM Protocol *1909*
Distributed Files *1919*

付　録　G　削除予定履歴 1921

付　録　H　システムマネジメントインフォメーション 1923

第 1 章 （ OLE 版 ）
はじめに

OpenPOSコントロール

OpenPOSとは？

OpenPOS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。

OpenPOS（略して"OPOS"）の目的は次のとおりです。

Win32ベースPOSデバイスアクセス用アーキテクチャの定義

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。

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（本マニュアル）

Control Programmer's Guide 日本版仕様書：ハードウェアプロバイダ向け

OPOSで定義するヘッダファイル

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

必要最小限の機能を組込んだコモンコントロールオブジェクトが入手できる。これらのコントロールオブジェクト及び他の有用な情報についてはウェブサイト：http://www.nrf-arts.org 参照。

マニュアル対象者

本マニュアルの対象読者は、主としてPOS周辺装置にアクセスするアプリケーション開発者およびハードプロバイダです。OPOSコントロールを作成するシステム開発者も対象になります。

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。

POS周辺装置の一般的特徴

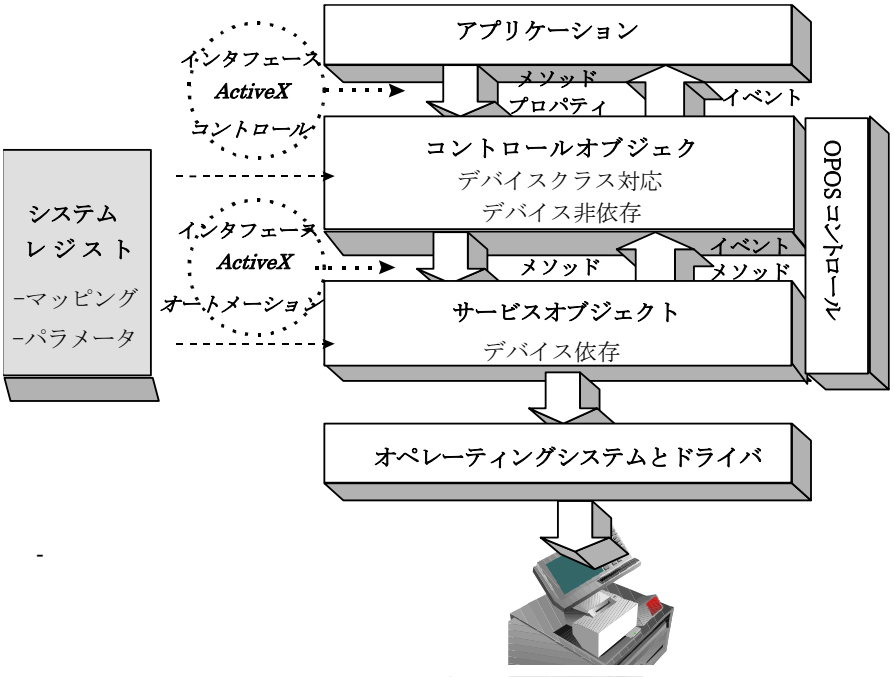
ActiveXコントロールおよびOLEオートメーションの用語とアーキテクチャ

Microsoft Visual Basic 、Microsoft Visual C++ など、ActiveXコントロールコンテナ開発環境に精通していることが望ましい

OpenPOSコントロール概要

OpenPOS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。このマニュアルではこれらのAPIについて説明します。

COは標準ActiveX(OLEの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、アウトプロセスサーバとして構築する場合には注意が必要です。詳細については、「Contorol Programmer’s Guide日本版仕様書」を参照願います。

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。

このマニュアルでは、通常、単に「コントロール」と言います。場合によっては、COが実行する機能とSOが実行する機能を区別しなければなりません。その場合には、レイヤを明示します。

OPOSコントロール使用方法

アプリケーションが最初に行わなければならないことは、**Open**メソッドを呼び出すことです。このメソッドのパラメータで、コントロールと関連するデバイス名を選択します。**Open**メソッドは以下のステップを実行します。

デバイス名でリンクを確立します。

OpenResult,Claimed、DeviceEnabled、DataEventEnabled、FreezeEvents、AutoDisable、DataCount、BinaryConversion プロパティを、OPOSコントロールレイヤのコピーライト記述およびバージョン番号と同様に初期化します。その他のクラス固有プロパティも初期化されます。

いくつかのアプリケーションが、同時にOPOSコントロールをオープンします。従って、デバイスをオープンした後、一般に、デバイスに排他アクセスするため、アプリケーションが**ClaimDevice**メソッドを呼び出さなければなりません。多くの場合、まずデバイスの排他アクセス権を獲得しなければなりません。**ClaimDevice**メソッドの後、コントロールはメソッドやプロパティへのアクセスを許可します。排他アクセス権を獲得すると、ほかのアプリケーションがそのデバイスへアクセスすることはできません。

アプリケーションは、ほかのアプリケーションとデバイスを共有できる場合、たとえばトランザクション終了時、排他アクセス権を**ReleaseDevice**メソッドにより、解放することもできます。

アプリケーションがデバイスを使用するには、まず、**DeviceEnabled**プロパティをTRUEに設定しなければなりません。この値でデバイスはイネーブルになりますが、FALSEではデバイスをディセーブルにします。たとえば、スキャナコントロールをディセーブルにすると、（可能ならば）デバイスが物理的にディセーブルにされます。デバイスを物理的にディセーブルにするかしないかに関わらず、イネーブルにするまで、デバイスからの入力は受け付けられません（無視されます）。

アプリケーションがデバイスの使用を終了した後、**Close**メソッドを呼び出してデバイスと関連のある資源を解放する必要があります。**DeviceEnabled**プロパティがTRUEであれば、**Close**メソッドは、デバイスをディセーブルにします。**Claimed**プロパティがTRUEの場合、**Close**は排他アクセスのロックを解除します。アプリケーションは、終了する前に、オープンしているOPOSコントロールのすべてをクローズする必要があります。

要約すると、アプリケーションは以下の一般的シーケンスに従います。

Openメソッド：コントロールオブジェクトをサービスオブジェクトにリンクするために呼び出します。

ClaimDeviceメソッド：デバイスに排他アクセスできるようにするために呼び出します。排他使用するデバイスでは必須であり、共有可能デバイスには任意のメソッドです。（詳細については、「デバイス共有モデルについて」を参照してください。）

DeviceEnabledプロパティ：デバイスを動作させる場合TRUEに設定します。（共有可能デバイスの場合、デバイスを先にClaimDeviceメソッドで排他アクセスとせずに、イネーブルしてもかまいません。）

デバイスを使用します。

DeviceEnabledプロパティ：FALSEに設定してデバイスをディセーブルにします。

ReleaseDeviceメソッド:デバイスの排他アクセスを解除するために呼び出します。

Closeメソッド：サービスオブジェクトをコントロールオブジェクトから解放するために呼び出します。

初期化とエラー通知　　*第1.11版にて追加*

エラー状態は、一つまたはそれ以上の初期化API、つまり**Open**メソッドや**ClaimDevice**メソッドの呼び出し、および**DeviceEnabled**=TRUEの処理に失敗した場合、必要になります。以下を、サービスオブジェクトの実装者による初期化期間中のエラー処理として推奨します。しかしながら、いくつかのハードウェアデバイスの幅広いバリエーションと、それに対する様々な初期化要求および、すでにリリースされた様々なサービスオブジェクトが存在するので、これらのガイドラインには必ずしも従う必要がありません。

Openメソッド

主要目的：サービスオブジェクトの生成と、サポートしているソフトウェアコンポーネントの初期化を含む、ソフトウェアスタックの初期化をすること。

１．ソフトウェアの初期化に失敗した場合、サービスオブジェクトはOpenメソッドの呼び出しに失敗しなければいけません。

例：サポートしているソフトウェアコンポーネントはインストールされていない、または利用ができません。従ってOpenメソッドの呼び出しに失敗します。

２．サービスオブジェクトが**Open**メソッド実行時に初期化すべきプロパティ（例えば性能を示すプロパティ）を正確に設定するためにデバイスを調べなければならない場合、デバイスにアクセスできないならば、**Open**メソッドの呼び出しに失敗します。

例：一つのサービスオブジェクトがいくつかのラインディスプレイモデルをサポートし、デバイスと通信後、UnifiedPOSの機能を設定します。デバイスのポートが利用できない、または応答がない場合、サービスオブジェクトが**Open**メソッド処理を完了できず、APIの呼び出しが失敗する必要があります。

３．他のケースでは、サービスオブジェクトが**Open**メソッドの呼び出しに成功し、その後（必要ならば）失敗を通知します。

例：サービスオブジェクトは、RS232ポートを開けません。上記のケース２が適用されない場合でポートがまだ利用できないなら、サービスオブジェクトは**Open**メソッドに成功してから、**ClaimDevic**eメソッド実行時にポートオープンの失敗を通知します。

ClaimDeviceメソッド

主要目的：排他使用デバイスのために、デバイスの排他アクセス権を取得する。

１．別のプロセスがデバイスの排他アクセス権をすでに取得していることによりタイムアウトが発生し、サービスオブジェクトは**ClaimDevice**メソッドの呼び出しに失敗しなければなりません。

２．デバイスにアクセスできない場合、サービスオブジェクトは**ClaimDevice**メソッドの呼び出しを失敗します。

例：必要な通信ポート、またはI/Oポートのオープンおよび、排他アクセス権の取得ができない。サービスオブジェクトはデバイスが存在していないか、またはオフラインになっているかを識別します。これらのそれぞれの場合、サービスオブジェクトは**ClaimDevice**メソッドの呼び出しに失敗します。

３．その他の場合、サービスオブジェクトは**ClaimDevice**メソッドの呼び出しに成功します。このことを厳密に言うならば、実行時にエラーがあった場合を含んでいます。

例：POSプリンタ、レシートステーションの用紙切れ、または紙ジャムを検出した場合。これらは操作の時々に起こる、実行時のエラーで、ユーザーにより修理可能です。サービスオブジェクトが**ClaimDevice**メソッドの呼び出しを成功します。POSプリンタの実行時のエラーは、（**DeviceEnabled**=TRUEになった後）**StatusUpdateEvent**と**PrintNormal**などのメソッドからの戻り値によって通知されます。

DeviceEnabled=TRUE

主要目的：オペレーションとアプリケーションを使用するための最終準備。

１．デバイスとの通信ができない場合、サービスオブジェクトは**DeviceEnabled**= TRUEの呼び出しに失敗します。（デバイスは**ClaimDevice**メソッド実行時に通信可能であっても、現在は通信できないのでご注意ください。）

例：サービスオブジェクトはデバイスが存在してないか、またはオフラインになっているかを識別し、サービスオブジェクトはメソッドの呼び出しに失敗します。

２．その他の場合、サービスオブジェクトは**DeviceEnabled**=TRUEの呼び出しに成功します。このことを厳密に言うならば、実行時にエラーがあった場合を含んでいます。

例：上記の排他アクセス権を取得するClaimDeviceメソッドのケース３をご覧下さい。

アプリケーション開発者は、どのような初期化時のエラーに対しても備えておかなければなりません。ハードウェアデバイスと、それらサービスオブジェクトの実装のバリエーションがある中で、熟慮されたアプリケーションは、対応可能な最大限のエラー状態とそれらの可能な限りの通知を思い通りに行うでしょう。

Retailデバイスは、RS232、RS485、Parallel、USB、Ethernet、Wirelessを含んだ、多種のポートを用いてPOS端末と通信しています。加えて、デバイスはPOS端末または外部電源によって（直接）電源を供給されます。これらのガイドラインはすべてのデバイスに適用されます。典型的な初期化の2つの例を以下にあげます。

例１：携帯型スキャナがPOS端末の電源供給されたRS232ポートに接続されています。

Open： ソフトウェアの初期化が成功した場合、**Open**メソッドは成功します。

ClaimDevice：**Open**メソッドが成功し、デバイスとの通信に成功した場合、**ClaimDevice**メソッドは成功します。

DeviceEnabled=TRUE : **ClaimDevice**メソッドが成功し、デバイスとの通信に成功した場合、**DeviceEnabled**=TRUEの処理は成功します。

DeviceEnabled=TRUEの間 ： デバイスが電源供給されたRS232ポートから引き抜かれた場合、電源状態の変化を検出し、アプリケーションへ通知します。デバイスが後から再び接続された場合、電源状態の変化を検出し、アプリケーションへ通知します。多くのデバイスは、電源状態の変化をRS232のDSR信号の監視によって取得できます。（このポートタイプ(RS232)の電源が入ったままの抜き差しは、恐らくハードウェアベンダーに推奨されないのでご注意ください。）

例２ ： 専用外部電源によって電源供給されたデッキタイプのスキャナ／秤がPOS端末のUSBポートに接続されています。

Open : ソフトウェアの初期化が成功した場合、**Open**メソッドは成功します。

ClaimDevice : **Open**メソッドが成功し、デバイスとの通信に成功した場合、**ClaimDevice**メソッドは成功します。

DeviceEnabled=TRUE : **ClaimDevice**メソッドが成功し、デバイスとの通信に成功した場合、**DeviceEnabled**=TRUEの処理は成功します。

DeviceEnabled=TRUEの間 ： デバイスがUSBポートから引き抜かれた、または電源が抜かれた場合、電源状態の変化を検出し、アプリケーションへ通知します。デバイスが後から再び接続された場合、電源状態の変化を検出し、アプリケーションへ通知します。OS固有のメカニズムが、ポートのopen、writeまたはread実行時に、固有のエラー状態として電源状態の変化を検出します。

一般的な初期化の処理は非常に似ていますが、例２であってもデバイス接続の監視と再初期化するためのサービスオブジェクト内での標準的なある程度のロジックが必要なのでご注意ください。

メソッドとプロパティの使い方

メソッド

Openが成功する前に、ほかのメソッドを呼び出してはなりません。呼び出した場合、OPOS_E_CLOSEDステータスが戻るのみです。

排他使用デバイスの場合、アプリケーションが**ClaimDevice**メソッドを呼び出し、**DeviceEnabled**プロパティをTRUEに設定してから、ほかのメソッドを呼び出す必要があります。

共有可能デバイスの場合、**DeviceEnabled**プロパティをTRUEに設定してからほかのメソッドを呼び出します。

各メソッドに特有の条件については、各デバイスクラスの章の「**一覧**」の節を参照してください。

プロパティ

Openに成功しないと、大半のプロパティの値は初期化されません。書込み可能プロパティを設定しようとしても無視されます。以下のプロパティは必ず初期化されます。

プロパティ	値
State	OPOS_S_CLOSED
ResultCode	OPOS_E_CLOSED
ControlObjectDescription	コントロールオブジェクトコピーライト記述
ControlObjectVersion	コントロールオブジェクトバージョン番号

機能の有無を示すプロパティは、**Open**が呼び出された後に初期化されます。

排他使用デバイスの場合、アプリケーションが**ClaimDevice**メソッドを呼び出し、**DeviceEnabled**プロパティをTRUEに設定してから、ほかのプロパティの初期化や書込みを行う必要があります。

共有可能デバイスの場合、**DeviceEnabled**プロパティをTRUEに設定してからほかのプロパティの初期化や書込みを行います。

プロパティがいつ初期化される、もしくは書き込み可能になるかについては、各デバイスクラスの「**一覧**」およびプロパティの説明を参照してください。

これらの条件を満たす前に書込み可能プロパティを設定しても、書込みは無視され、**ResultCode**プロパティをOPOS_E_NOTCLAIMEDまたはOPOS_E_DISABLEDに設定します。

初期化されていないプロパティを読み取ると、特に指定されていない限り、以下の値が戻されます。

プロパティタイプ	値
Boolean	FALSE
Long	0
String	“[Error]”　– かぎ括弧を含む。

プロパティを初期化した後は、排他アクセス要求やイネーブルにしてもプロパティを再初期化することはありません。**Close**メソッドが呼び出されるまでは初期化されたままです。

ステータス、結果コード、状態モデル

次の表に示すように、共通なプロパティ、イベント、メソッドの中には、ステータス、結果コード、状態モデルが設定されているものがあり、クラス固有コンポーネントによってサポートされます。

名前	意味
State	コントロールの現在の状態を含むプロパティです。 OPOS_S_CLOSED OPOS_S_IDLE OPOS_S_BUSY OPOS_S_ERROR
ResultCode	直前のメソッドか直前で変更した書込み可能プロパティのステータスを示すプロパティです。 OPOS_SUCCESS OPOS_E_CLOSED OPOS_E_CLAIMED OPOS_E_NOTCLAIMED OPOS_E_NOSERVICE OPOS_E_DISABLED OPOS_E_ILLEGAL OPOS_E_NOHARDWARE OPOS_E_OFFLINE OPOS_E_NOEXIST OPOS_E_EXISTS OPOS_E_FAILURE OPOS_E_TIMEOUT OPOS_E_BUSY OPOS_E_EXTENDED
ResultCodeExtended	直前のメソッドか直前で変更した書込み可能プロパティの拡張ステータスを示すプロパティです。値は**ResultCode**の値とデバイスクラスで変わります。
StatusUpdateEvent	クラス固有の状態かステータス変数が変化したときに発生するイベントです。 第1.3版以降：すべてのデバイスに対し、デバイスの電源状態を通知できるAPIが用意されました。（詳細については、「**デバイス電源通知モデルについて**」の項目を参照して下さい。）
ErrorEvent	**State**がOPOS_S_ERRORに変化したときに発生するイベントです。

ステータスモデル

ステータスモデルの規則は次のとおりです。

ステータスモデルがすべてのデバイスクラスに共通にもつ唯一の機能は、アプリケーションに警報を出すことで、StatusUpdateEventの通知によって行います。

各デバイスクラスは、デバイスで発生する固有のステータス変化を決めています。デバイスクラス固有のステータス変化の例を次に示します。

- ドロワー位置の変化（たとえば、オープンからクローズへの遷移。）
- POSプリンタセンサの変化（たとえば、スリップが挿入されたことを示す「用紙有り」センサのアクティブ化。）

結果コードモデル

結果コードモデルの規則は次のとおりです。

どのメソッドも結果コードを戻します。このコードはResultCodeプロパティにも格納されます。

書込み可能プロパティを設定すると、結果コードがResultCodeプロパティに格納されます。

ResultCodeがOPOS_SUCCESSであれば値0が割り当てられます。ゼロ以外の値はエラーか警告を意味します。

コントロールは結果コードに、一覧に示す値のどれか1つを選択します。報告された結果をこれらのコードで表すことができない場合、ResultCodeにOPOS_E_EXTENDEDに設定し、ResultCodeExtendedに、各デバイスクラスの章で記述されている値を設定します。

> コントロールがResultCodeをOPOS_E_EXTENDED以外の値に設定した場合、サービスオブジェクトはResultCodeExtendedプロパティをサービスオブジェクトに特有の値に設定できます。アプリケーションがこれらの値を使用する場合、サービスオブジェクトに特有の値を追加する必要があります。（アプリケーションがそのようなコードを追加する必要がある場合、ServiceObjectDescription、DeviceDescriptionまたはDeviceNameプロパティに問い合わせて、扱っているサービスオブジェクトを確認することができます。）

状態モデル　　*第1.7版にて更新*

状態モデルの規則は次のとおりです。

コントロールのStateプロパティの初期値はOPOS_S_CLOSEDです。

Openメソッドを呼び出して、その結果がOPOS_SUCCESSの場合、Stateプロパティは OPOS_S_IDLEに変化します。

OPOSが出力処理中のとき、StateプロパティはOPOS_S_BUSYに設定されます。正常に終了すると、OPOS_S_IDLEに戻ります。

以下の場合、OPOS_S_ERRORに変化します。

- 非同期出力がエラー状態になりました。
- イベント駆動入力の収集または処理時にエラーが発生しました。

OPOSは、StateプロパティをOPOS_S_ERRORに変更した後、ErrorEventを通知します。このイベントのパラメータは、結果コード、拡張結果コード、エラーの発生場所、およびアプリケーションがエラーに応答するポインタです。発生場所としては、次の3つのエラー位置が定義されています。

- Output – 以前にキューされた出力を処理するときにエラーが発生しました。
- InputWithData –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アプリケーションで利用できます。このエラー位置の場合、InputNoData値で次のErrorEventを受信するまで、アプリケーションは入力の処理を継続できます。あるいは、入力をクリアできます。
- InputNoData –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、入力済のデータをすべて処理したか、利用できる入力データがないことを示します。

アプリケーションは、ErrorEventから戻るとき、応答パラメータを変更できます。以下の応答値があります。

- Retry – エラー位置の値がOutputの場合：　非同期出力を再試行し、エラー状態を終了します。再試行中にエラーが発生すると、別のErrorEventが生成されます。
  エラー位置の値がInputの場合：　SOによる再試行が可能ならば、デバイスによっては入力を再試行します。
  このRetryはエラー位置の値がOutputのときのデフォルト応答です。
- Clear – バッファリングされている全出力データ（全ての非同期出力を含む）かバッファリングされている入力データをクリアしエラー状態を終了します。このClearはエラー位置の値がInputNoDataのときのデフォルト応答です。

- Continue – エラー位置の値がInputWithDataの場合だけ使用できます。この応答はエラーを容認し、コントロールに処理を継続させる場合に指定します。コントロールはエラー状態のままで、DataEventEnabledプロパティの指示に従ってデータイベントを発生します。すべての入力がイベント処理された後、DataEventEnabledプロパティを再度TRUEに設定すると、エラー位置の値がInputNoDataで別のErrorEventが通知されます。このContinueはエラー位置の値がInputWithDataのときのデフォルト応答です。

コントロールは、アプリケーションがErrorEventを処理している間、別のErrorEventを通知しないようにします。

デバイス共有モデルについて

OpenPOSのデバイス共有モデルは、複数のアプリケーションが部分的または全面的に共有できるデバイスのほか、同時にはひとつのアプリケーション[1] が排他的に使用するデバイスをサポートします。（詳細については、「**メソッドとプロパティの使用方法**」を参照してください。）全てのOPOSコントロールは、一つ以上のアプリケーションにオープンされます。しかし、コントロールで実行できるアプリケーションの動作の中には、デバイスへのアクセス権を獲得するアプリケーションを制限するものがあります。

排他使用デバイス

もっとも一般的なデバイスタイプは「排他使用デバイス」で、一例がPOSプリンタです。物理的特性、動作上の特性により、このデバイスは同時には1つのアプリケーションでしか利用できません。アプリケーションは、大半のメソッド、プロパティ、イベントが有効になる前に、**ClaimDevice**メソッドを呼び出して排他アクセスできるようにしなければなりません。排他アクセス権を獲得するまえに、メソッドを呼び出したり、プロパティを設定したりすると、OPOS_E_NOTCLAIMEDエラーが発生し、イベントはアプリケーションに通知されません。

密接に関連する2つのアプリケーションが共有方式で排他使用デバイスを使用したい場合、一方のアプリケーションは一部の時間だけデバイスを排他アクセスして、その後、他方のアプリケーションも使用できるように、排他アクセスを解除するという方法もあります。

ClaimDeviceメソッドが再び呼び出されると、設定可能なデバイスの特性を示すプロパティは**ReleaseDevice**メソッド時の状態に復元されます。復元される例としては、ラインディスプレイの輝度、磁気ストライプリーダの読み取りトラック、プリンタの行あたりの文字などがあります。プリンタのセンサプロパティのような状態を示すものは復元されません。その代わり、現在の状態に設定されます。

1. アプリケーションは1プロセスだけからなるものとしています。複数のプロセスからなるアプリケーションを生成することもできますが、一般的ではありません。技術的にはプロセス単位でデバイスを共有します。しかし、単一プロセスアプリケーションでは、共有をアプリケーションレベルとして見ることができます。

共有可能デバイス

いくつかのデバイスは共有デバイスです。一例がキーロックです。共有可能デバイスの場合、複数のアプリケーションでそのメソッドを呼び出して、プロパティにアクセスすることが可能となります。また、デバイスをオープンしたすべてのアプリケーションにイベントが通知されます。ただし、共有可能デバイスを排他アクセスしたアプリケーションにより、一部のメソッドやプロパティのアクセスを制限することや、そのアプリケーションだけにイベントを通知させることもできます。

備考　アプリケーションをできる限り柔軟にするためには、すべてのデバイスを共有可能にすべきであると考えるかもしれません。しかし、実用上、このような柔軟性は有効ではありません。下位層が、かなり複雑で、正確さに欠けるインプリメントになります。

アプリケーションを開発するのに十分、堅牢であり、ハードウェア製造者が実装可能な仕様としており、本仕様では大半のデバイスを排他使用デバイスと規定し、複数のアプリケーションによる同時使用に効果的であるデバイスのみを共有可能デバイスとしました。

イベント

OpenPOSはイベントを使用してOPOSコントロールでの様々な動作や変化をアプリケーションに通知します。イベントには以下の5つがあります。後の章でその定義を示します。

DataEvent：入力データがデバイスクラス固有のプロパティに格納されています。

ErrorEvent：イベント駆動入力中または非同期出力中にエラーが発生しました。

StatusUpdateEvent：デバイスの状態変化を報告します。

OutputCompleteEvent：非同期出力が正常に完了しました。

DirectIOEvent：このイベントは、サービスオブジェクトプロバイダによって定義され、本仕様でカバーできないものを提供します。

サービスオブジェクトはイベントが発生するとイベントをキューイング(Enqueue)します。これらのイベントは、アプリケーションのスレッドではなくしばしばワーカースレッドによってキューイング(Enqueue)されます。キューイング(Enqueue)されたイベントは通知可能な状態になるとアプリケーションに通知(Deliver)されます。イベントの引き渡しを遅らせる要因には以下のものがあります。

アプリケーションスレッドが他のメッセージ処理のためビジーである場合。OPOSコントロールはOLE Apartment Threadingモデルに従わなければなりません。OLE Apartment Threadingの規定によると、イベントはCOMオブジェクトを作成したスレッドで通知(Deliver)されなければならず、それは通常アプリケーションのメインスレッドとなります。アプリケーションが他のメッセージを処理している場合は、その処理が終了するまで、イベント引き渡し(Delivery)を待たなければなりません。

アプリケーションがFreezeEventsプロパティをTRUEに設定している場合。（ FreezeEventsプロパティを参照。）

イベントタイプがDataEventまたは入力ErrorEventで、DataEventEnabledプロパティがFALSEである場合。（「入力モデルについて」参照。）

以前にキューイング(Enqueue)されたイベントが上記のいずれかの理由により待たされているときは、それ以降のすべてのイベントもまた待たされる場合もあります。すなわち、キューイング(Enqueue)されたイベントの通知(Delivery)は一般的にFIFOに従います。キュー内のいかなる種類のイベントも特に優先順位は持ちません。

“Application Programmer’s Guide Release 1.3”（英語版）において、イベントの通知に関する用語の解説が下記のとおり追加され、仕様書内で一貫して使い分けられています。日本版仕様書においては、本項「イベント」以外ではこれらの使い分けは行っていませんが、参考のため掲載しています。

備考ー用語

下記のイベントに関する用語はこの仕様書の中で一貫して使用されるものです。実装によってはここで説明するモデルと若干異なるかもしれませんが、実質的には同様です。

Enqueue :　SOがアプリケーションに対しイベントの通知の必要性を決定したとき、SOが内部のイベントキューにイベントを詰め込みます。一般的にイベントの詰め込みは、SO内部の1つまたはそれ以上のワーカースレッドによって行われます。

Deliver :　イベントキューが空でなく、キューの先頭のイベントについてのすべての条件が揃ったとき、このイベントはキューから外され、アプリケーションに対するイベント通知要求が実行されます。一般的にイベント通知要求は、SO内部の専用スレッドにより管理されます。このスレッドは、アパートメント・スレッディング・モデルに従うため、コントロールを生成したスレッドのコンテキストにおいてイベント通知要求する必要があります。

Fire :　キューイングと通知要求の組み合わせと言えます。時にこの用語は大まかに使用され、これらのステップの片方だけしか意味しない場合もあります。読者は前後関係からこれらを識別しなければなりません。

イベントのキューの管理に関する規定は以下の通りです。

コントロールは、デバイスがイネーブルである間は新しいイベントをキューイング(Enqueue)するのみです。

コントロールは、アプリケーションがReleaseDeviceメソッド（排他使用デバイス）またはCloseメソッド（その他のデバイス）を呼び出すまで、キューイング(Enqueue)されたイベントの通知(Deliver)を行う可能性があります。また、前記メソッド実行時には残存しているイベントは削除されます。

入力デバイスでは、ClearInputメソッドはデータとエラーイベントをクリアします。

イベントハンドラ内で、アプリケーションはプロパティにアクセスしたり、メソッドの呼び出しを行えます。しかし、 アプリケーションは**ReleaseDevice**メソッドや**Close**メソッドをイベントハンドラ内から呼び出してはなりません。 **ReleaseDevice**メソッドはイベントハンドリング（イベントを通知(Deliver)させるスレッドを含んでいる）を閉鎖してしまい、**Close**は戻る前にイベントハンドリングを閉鎖しなければならないためです。

入力モデルについて *第1.13版にて更新*

第 1.10 版にて修整

OpenPOSの入力はイベント駆動入力をサポートします。入力データは**DeviceEnabled**がTRUEに設定された後に受信できるようになります。受信されたデータは、条件が満たされている場合に、アプリケーションに引き渡される**DataEvent**としてキューイングされます。データ受信時に**AutoDisable**プロパティがTRUEの場合、コントロールは**DeviceEnabled**をFALSEに設定して自動的にデバイスをディセーブルにします。これはコントロールが更に入力をキューイングするのを妨げ、可能な場合は物理的にデバイスをディセーブルにします。

アプリケーションは、デバイスからの入力を受信する準備ができたら**DataEventEnabled**プロパティをTRUEに設定します。入力を受信すると（通常、ハードウェア割込みの結果として）、コントロールは**DataEvent**を生成します。（入力がすでにキューイングされている場合は、**DataEvent**が通知されます．）このイベントには数値パラメータによって入力ステータス情報を持つことがあります。コントロールは、イベントを通知する直前に、入力データと、必要に応じてその他の情報をデバイス固有プロパティに格納します。

イベントを通知する直前に、OPOSは**DataEventEnabled**プロパティをFALSEに設定することによって、それ以上のデータイベントをディセーブルにします。これによりアプリケーションが現在の入力と関連のプロパティを処理している間、それ以上の入力データはOPOSによってキューイングされます。現在の入力の処理を終了し、次のデータを受け付ける準備ができたとき、アプリケーションは**DataEventEnabled**をTRUEに設定してイベントの通知を再開するようにします。

アプリケーションがデバイスを無効にする場合（**DeviceEnabled**がFALSEまたは、**AutoDisable**がTRUEで入力イベントが待機状態になったとき）、デバイスを再び有効にするまで、以降の入力データを無視するロジックが必要になります。特に、既に受信し待機している入力データは、通知できるまで待ちます。（ただし、未通知の入力データは、**Release**または**Close**メソッドが呼ばれた時に廃棄されます。）アプリケーションは、**DataEventEnabled**プロパティまたは FreezeEventsプロパティを使うことでこの通知をコントロールできます。

入力デバイスが排他使用デバイスの場合、アプリケーションは、入力の読取りを開始する前に、そのデバイスの排他アクセス権の獲得とイネーブルの両方を行わなければなりません。

共有可能な入力デバイスの場合、アプリケーションは、入力の読取りを開始する前に、そのデバイスをオープンして、イネーブルにしなければなりません。アプリケーションはOPOSが**DataEvent**を使用して

データを送る前に、**ClaimDevice**メソッドを呼び出してデバイスの排他アクセスを要求しなければなりません。イベント駆動入力を受信したとき、アプリケーションがデバイスを排他アクセスしていない場合、アプリケーションがデバイスの排他アクセス権を獲得する（かつ**DataEventEnabled**プロパティがTRUEになる）まで、入力はキューイングされます。これによって、複数のアプリケーションがデバイスを順番に共有でき、アプリケーション間で入力フォーカスを効果的に受け渡すことができます。

イベント駆動入力によりデータの入力処理をしているときにOPOSがエラーを見つけると、 OPOSはその状態をErrorに変更し、**ErrorEvent**を通知してアプリケーションにエラー状態の警報を出します。**DataEventEnabled**プロパティがTRUEになるまで、このイベントは通知しないので、アプリケーションは順序よく処理することができます。エラーイベントは、次のエラー位置を示すデータを有して通知されます。

- InputWithData (OPOS_EL_INPUT_DATA) –入力データが前もってバッファリングされている間にエラーが発生した場合に通知します。このイベントはすべての入力データより先にキューイングされます。（一般的な実装では、このイベントをイベントキューの先頭にセットします。）このイベントにより、アプリケーションは入力を直ちにクリアするか、オプションでユーザにエラーの警報を出して、バッファリングされた入力済のデータを処理することができます。

  後者のケースはスキャナコントロールで有効です。ユーザに直ちにエラー警報を出すことができるので、エラーが解除されるまで後続のアイテムはスキャンされません。エラー復旧を実行する前に、以前スキャンしたアイテムを正常に処理できます。

- InputNoData (OPOS_EL_INPUT)－エラーが発生し、利用できるデータがないときに通知します。（一般的な実装では、このイベントをイベントキューの末尾にセットします。）エラーが発生したときに入力データの一部をすでにキューイングした場合、エラー位置がInputWithDataのErrorEventがまず発生し、このエラーイベントはすべてのキューイングデータが処理されたことを示します。（ “InputWithData”のイベント通知によるアプリケーションのイベントハンドラが、”Clear”応答した場合、本“InputNoData”を有するイベントは通知されません。）

コントロールは以下のいずれかが行われた場合に、Error状態を終了します。

アプリケーションのイベントハンドラがInputNoDataのErrorEventを終了する場合。

アプリケーションのイベントハンドラがInputWithDataのErrorEventをOPOS_ER_CLEARで終了する場合。

アプリケーションがClearInputメソッドを呼び出した場合。

コントロールによっては、イベント駆動入力を開始するためにメソッドを呼び出す必要があります。コントロールが入力データを受け取った後、一般的には入力を再初期化するためにメソッドを再度呼び出すまでは、以降の入力データを受け付けません。例として、ＭＩＣＲとシグニチャキャプチャがあります。このタイプのイベント駆動入力は「非同期入力」と呼ばれることもあります。

DataCountプロパティはコントロールでキューイングされた**DataEvent**の数を得るために参照されます。

コントロールがキューイングしたすべての入力データは、**ClearInput**メソッドを呼び出せば削除できます。 **ClearInput**は、共有可能デバイスでは**Open**後に、排他使用デバイスでは**ClaimDevice**後に呼び出せます。

DataEventや**ErrorEvent**の結果通知された全てのデータプロパティは、**ClearInputProperties**メソッドを呼び出せば初期値に戻ります。この呼び出しでは、**DataCount**と**State**プロパティはリセットしません。

一般的なイベント駆動入力モデルは、入力データを直接戻すメソッドやプロパティを含むデバイスクラスの定義を特には定めていません。デバイスクラスの中には、より直覚的または柔軟に動作するよう、メソッドやプロパティを定義しているものがあります。

例として、キーロックがあります。このタイプの入力は「同期入力」と呼ばれることもあります。

入力データの順序について

複数の入力デバイスからの入力順の保証は、アプリケーションの処理に一任されます。したがって、**DataEvent**を受け取ったアプリケーションは、すみやかに、次の**DataEvent**を受け取れるように処理をしなければなりません。この順序性は、PC標準キーボード入力も含めて、アプリケーションで考慮する必要があります。

たとえば、商品コードをスキャナより入力し、商品が複数ならば、スキャナからの入力の後でキーボードより個数を入力するシステムを想定し、実際の操作が次の順序で行われたとき、

　　商品1の読み取り、商品2の読み取り、個数

商品1の読み取りによる**DataEvent**の通知の後、商品2の読み取りによる**DataEvent**通知が保留されている間に、個数のキーボード入力をアプリケーションが受け取ると、

　　商品1の読み取り、個数、商品2の読み取り

の操作が行われたことになり、個数は商品1の数として処理されてしまいます。このようなことになるのは、商品1の読み取りによるスキャナからの**DataEventEnabled**プロパティをTRUEに設定していないときに発生します。

このように複数の入力デバイスからの入力データが関連する場合には、入力デバイスの**DataEvent**のイベント処理は、可能な限り短時間に終了し、次の**DataEvent**をすみやかに受け取り可能にする必要があります。また、場合によっては、データ入力用の専用スレッドかタスクが必要になるかもしれません。

出力モデルについて　*第1.13版にて更新*

OpenPOSの出力は、非同期と同期の2つの出力タイプからなります。各デバイスタイプは一方または両方のタイプをサポートすることも、両方ともサポートしないこともあります。

同期出力

デバイス出力を迅速に実行したい場合にこの出力タイプが有効です。このメリットは単純さにあります。

アプリケーションはクラス固有メソッドを呼び出して出力を実行します。出力が完了するまで、制御はアプリケーションに戻りません。

非同期出力

デバイス出力が低速のハードウェアで、会話的に処理する必要がある場合に、この出力タイプが有効です。デバイスが出力を実行しているときに、アプリケーションが別の処理を実行できることから、このメリットは確実な応答性にあります。

アプリケーションはクラス固有メソッドを呼び出して出力を開始します。コントロールはプログラムメモリー内のリクエストをバッファリングし、このリクエストの一意な整数の識別子を**OutputID**プロパティに設定して、できる限り迅速に戻ります。デバイスがリクエストを正常に終了すると、OPOSが**OutputCompleteEvent**を通知します。このイベントのパラメータは完了したリクエストの**OutputID**です。アプリケーションは、デバイスへのデータ送信に成功したことを追跡するために、非同期処理メソッドの呼び出しによって設定された**OutputID**と、**OutputCompleteEvent**の**OutputID**プロパティを比較します。

非同期リクエストを実行しているときにエラーが発生すると、**ErrorEvent**が通知されます。アプリケーションのイベントハンドラは、未処理の出力を再試行するかまたは、それをクリアできます。**ErrorEvent**が処理中は、コントロールはErrorの状態です。

（備考：エラーを引き起こす原因が除去されていない場合は、コントロールは直ちにError状態に再びなり、別の**ErrorEvent**を通知します。）

非同期出力はFIFOベースで実行されます。

コントロールがバッファリングした出力はすべて（非同期出力をすべて含む）、**ClearOutput**メソッドを呼び出すことによって削除できます。クリアした出力に対して**OutputCompleteEvent**は通知されません。このメソッドは（可能ならば）処理中の出力も停止します。

デバイス電源通知モデルについて

第 1.8 版にて更新

アプリケーションでは、使用しているデバイスの電源状態を知る必要がある場合があります。OPOSの以前のバージョンでは、この情報を通知する方法がありませんでした。
備考：このモデルは、PCやPOS端末本体の電源状態（「バッテリ・オン」や「バッテリ・ロー」のような）を通知することを目的にしていません。これらの情報の通知は、現在POSパワーのデバイスカテゴリで管理しています。

モデル

OPOSでは、デバイスの電源を３つの状態に分類します。

ONLINE：　デバイスは電源オン状態でかつレディ状態です。これは「使用可能」状態を示します。

OFF：　デバイスは電源オフ状態または本体に未接続状態です。これは「使用不可能」状態を示します。

OFFLINE：　デバイスは電源オン状態ですが、ノットレディ状態か応答不可状態です。ボタンを押してオンライン状態にする必要があるかもしれません。また、本体からの要求に応答できない状態かもしれません。これは「使用不可能」状態を示します。

上記の状態の組み合わせとして、下記の状態が定義されています。

OFF_OFFLINE：　デバイスは電源オフ状態、またはオフライン状態です。SOはこれらの２つの状態を識別することができません。

電源通知は、デバイスがオープン、排他アクセス権の獲得（排他デバイスの場合）を経てイネーブル状態となっている場合にのみ機能します。

備考－イネーブル／ディセーブル　vs.　電源状態

これら２つの状態は、全く別のもので通常は関連がありません。OPOSでは、「イネーブル」／「ディセーブル」は論理的な状態であるのに対し、電源状態は物理的な状態であると定義しています。デバイスによっては、論理的には「イネーブル」であっても、物理的には「オフライン」かもしれません。または、論理的には「ディセーブル」であっても、物理的には「オンライン」かもしれません。物理的な電源状態にかかわらず、OPOSは、デバイスがイネーブル状態の場合のみ状態を通知します。（SOは一般的にイネーブル状態の間しかデバイスと通信できないので、この制限が必要です。）

デバイスが「オフライン」の場合、SOはデバイスを「イネーブル」にしようとはしません。しかし、一度「イネーブル」になると、通常、SOは電源状態が変わっても（「オフライン」や「オフ」になっても）デバイスを「ディセーブル」にはしません。

プロパティ

OPOSのデバイス電源通知モデルは、すべてのデバイスクラスに対して下記の共通APIを追加しています：

CapPowerReportingプロパティ： デバイスの通知可能レベルを識別します。このプロパティの値は下記のうちのいずれかです。：

- OPOS_PR_NONE： SOはデバイスの電源状態の判断ができません。よって、電源通知機能は機能しません。
- OPOS_PR_STANDARD： SOは、２種類の電源状態を判断し通知が可能です。―― OFF_OFFLINE（電源オフまたはオフライン）とONLINE。
- OPOS_PR_ADVANCED： SOは、３種類のすべての電源状態を判断し通知が可能です。―― ONLINE、OFFLINE と OFF。

PowerStateプロパティ： 判断が可能ならば、SOによって現在の電源状態にセットされます。このプロパティの値は下記のうちのいずれかです。：

- OPOS_PS_UNKNOWN
- OPOS_PS_ONLINE
- OPOS_PS_OFF
- OPOS_PS_OFFLINE
- OPOS_PS_OFF_OFFLINE

PowerNotifyプロパティ： StatusUpdateEventsやPowerStateプロパティを使用した電源通知機能をイネーブルにするために、アプリケーションがこのプロパティをセットします。このプロパティはデバイスがイネーブルになる前にのみセットが可能です。（すなわちDeviceEnabledがTRUEにセットされる前）
この制限により、アプリケーションに対する悪影響を発生させることなく、電源通知機能の単純な実装が可能となっています。アプリケーションは、電源通知を受けるか否かのいずれかですが、これらの場合について切り換えの仕組みは必要ありません。このプロパティの値は下記のうちのいずれかです。：

- OPOS_PN_DISABLED
- OPOS_PN_ENABLED

DeviceEnabledプロパティと電源通知機能

CapPowerReportingが OPOS_PR_NONE以外の値で、かつ **PowerNotify**が OPOS_PN_ENABLEDの場合、下記の記述内容が、**DeviceEnabled**に追加されます。

コントロールがDeviceEnabledをFALSEからTRUEに変更すると、 電源状態の監視が開始されます。

- デバイスがONLINEの場合：
  - PowerState は OPOS_PS_ONLINE に設定されます。
  - StatusUpdateEvent が通知され、その *Status* パラメータは
- デバイスの電源状態がOFF、OFFLINE、OFF_OFFLINEの場合、コントロールはイネーブルはできないと判断し、ResultCodeにOPOS_E_NOHARDWARE、または OPOS_E_OFFLINE を設定します。

  しかし、デバイスをイネーブルにできない他の条件がなく、またコントロールがイネーブル要求に対し正常終了を返す場合：
  - PowerState は OPOS_PS_OFF、OPOS_PS_OFFLINE、または OPOS_PS_OFF_OFFLINE に設定されます。
  - StatusUpdateEvent が通知され、その *Status* パラメータは OPOS_SUE_POWER_OFF、OPOS_SUE_POWER_OFFLINE、または OPOS_SUE_POWER_OFF_OFFLINE が設定されます。

コントロールが DeviceEnabled を TRUE から FALSE に変更すると、OPOSとしては、コントロールがもはや電源状態の監視は行わないもの、と定義されています。したがって：

PowerStateは、OPOS_PS_UNKNOWNに設定されます。

デバイスインフォメーションレポーティングモデルについて

第 1.8 版にて追加

システム管理エージェントのようなアプリケーションでは、POS端末に取り付けられた様々なPOSデバイスの現在の構成や使用状態を監視する必要がある場合があります。

構成情報は、デバイスのシリアル番号、ファームウェア版数や接続方法などです。POSプリンタ使用状態の例は、印刷行数、稼働時間、用紙カット回数などです。スキャナ使用状態の例は、スキャン回数、稼動時間などです。MSR使用状態の例は、読み取り成功回数、読み取り失敗回数、稼働時間などです。後述のPOSデバイスカテゴリ毎に収集されるデバイススタティスティクスのXML定義例を参照してください。

情報はデバイス自身に収集格納される場合や、サービスオブジェクトで収集され、POS端末やストアコントローラに格納される場合があります。

複数のアプリケーション（例えばＰＯＳアプリケーションとシステム管理アプリケーション）が同一のデバイスからスタティスティクスを獲得したい場合、それぞれが必要な時に整然とデバイスにアクセスできるように、両方のアプリケーションが適切な注意を払わなければなりません。これはデバイスへのアクセスが必要になった時に、**ClaimDevice**メソッドの使用と**DeviceEnabled**プロパティをTRUEにセットし、その後デバイスへのアクセスが不要となった時に**DeviceEnabled**プロパティへのFALSEのセットと**ReleaseDevice**メソッドの使用によって行います。この方式によるデバイスアクセスの調整については、各アプリケーション自身の責任となります。

スタティスティクスレポーティングプロパティとメソッド

OPOSデバイスインフォメーションレポーティングモデルはすべてのデバイスクラスに下記の共通プロパティとメソッドを追加します。

CapStatisticsReportingプロパティ：　デバイスのスタティスティクスレポーティング能力を示します。FALSEの場合、そのデバイスに関する有効なスタティスティクスは有りません。これは以前の版数の仕様と互換なサービスと同じです。TRUEの場合、デバイスのどれかのスタティスティクスが有効です。

CapUpdateStatisticsプロパティ：　収集しているスタティスティクス（またはその一部）をアプリケーションがリセット/変更できるかどうかを示します。このプロパティはCapStatisticsReportingがTRUEの時のみ有効です。CapUpdateStatisticsがFALSEの場合、アプリケーションがリセット/変更できるスタティスティクスはありません。TRUEの場合、スタティスティクス（のどれか）をアプリケーションがリセット/変更できます。

ResetStatisticsメソッド：　CapStatisticsReportingとCapUpdateStatisticsの両方がTRUEの時のみ呼び出し可能です。このメソッドはリセット可能なデバイススタティスティクスの１つ、複数、あるいはすべてを０にリセットします。

RetrieveStatisticsメソッド：　CapStatisticsReportingがTRUEの時のみ呼び出し可能です。このメソッドは収集されているデバイススタティスティクスの１つ、複数、あるいはすべてを取得します。

UpdateStatisticsメソッド：　CapStatisticsReportingとCapUpdateStatisticsの両方がTRUEの時のみ呼び出し可能です。このメソッドはリセット可能なデバイススタティスティクスの１つ、複数、あるいはすべてを指定の値に変更します。

POSデバイススタティスティクスのXML定義

OPOS(UnifiedPOS)で定義された各デバイスカテゴリのスタティスティクスを含むXMLファイルは、この仕様書と同じWebサイトからダウンロードできます。これらのスタティスティクスは名前や “U_” の文字列を使ったグループをメソッドのパラメータ（の一部）に指定することで個別に参照することができます。

製造者/サービス提供者はベンダ固有のスタティスティクスを用意された “ManufacturerSpecific” セクションに追加できます。これらのスタティスティクスは名前や “M_” の文字列を使ったグループをメソッドのパラメータ（の一部）に指定することで個別に参照することができます。

以下の表はUnifiedPOSに定義された、すべてのデバイスカテゴリに共通な”DeviceInformation”セクションの情報の定義を示しています。

<DeviceInformation>

XML定義名称	定義内容説明
UnifiedPOSVersion	**サポートしている UnifiedPOS仕様書の版数**
DeviceCategory	**デバイスクラスカテゴリ (例えばPOSPrinter)**
ManufacturerName	**製造者名**
ModelName	**モデル名**
SerialNumber	**シリアル番号**
ManufactureDate	**製造日付**

MechanicalRevision

ハードウェア版数

FirmwareRevision

ファームウェア版数

Interface

接続方法 (例えばRS232やUSB)

InstallationDate

設置日付

以下は “UnifiedPOS” に定義されたキャッシュドロワデバイスカテゴリのスタティスティクスで記述されたXMLファイルの例です。

```
<?xml version='1.0' ?>
<UPOSStat version="1.8.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nrf-arts.org/IXRetail/namespace/"
xsi:schemaLocation="http://www.nrf-
arts.org/IXRetail/namespace/UPOSStat.xsd">
  <Event>
    <Parameter>
      <Name>DrawerGoodOpenCount</Name>
      <Value>1353</Value>
    </Parameter>
    <Parameter>
      <Name>DrawerFailedOpenCount</Name>
      <Value>2</Value>
    </Parameter>
    <ManufacturerSpecific>
      <Name>MyPersonalStat</Name>
      <Value>14.32</Value>
      <unitofmeasure>meters</unitofmeasure>
    </ManufacturerSpecific>
  </Event>
  <Equipment>
    <UnifiedPOSVersion>1.8</UnifiedPOSVersion>
    <DeviceCategory UPOS="CashDrawer"/>
    <ManufacturerName>Cashdrawers R
Us</ManufacturerName>
    <SerialNumber>12345</SerialNumber>
    <ModelName>CD-123</ModelName>
    <FirmwareRevision>1.0 Rev. B</FirmwareRevision>
    <Interface>RS232</Interface>
    <InstallationDate>2000-03-01</InstallationDate>
  </Equipment>
</UPOSStat>
```

すべてのデバイスカテゴリのスタティスティクスのXMLタグ名とスキーマ例を定義する最新のファイルは、NRF-ARTSのウェブサイト http://www.nrf-arts.org/ からダウンロードすることができます。

アップデートファームウェアデバイスモデルについて

第 1.9 版にて追加

ＰＯＳアプリケーションは、ＰＯＳ端末に接続されたさまざまなデバイスのファームウェアの更新機能をしばしば必要とします。このモデルは、ＯＰＯＳコントロールにより制御されるデバイスのファームウェアを更新するための、統一したアプリケーションインターフェースを定義します。

このモデルは以下の機能を有します。

CapUpdateFirmwareプロパティは、デバイスがファームウェアを更新する機能を有することを示します。

CapCompareFirmwareVersionプロパティは、ファームウェアファイルのバージョンがデバイスのファームウェアバージョンと比較できることをあらわします。

UpdateFirmwareメソッドは、デバイスのファームウェアの更新を非同期で実行します。

CompareFirmwareVersionメソッドは、デバイスのファームウェアバージョンとファームウェアファイルのバージョンを比較します。

StatusUpdateEventのStatusに、ファームウェア非同期更新の経過通知の値を追加します。

ファームウェア更新処理は、その進行を**StatusUpdateEvent**で通知することで非同期に処理されます。このファームウェア更新処理は、OPOSで定義される全てのデバイスに適用されます。

サービスオブジェクトがデバイスのファームウェアを実際に更新する手段は、OPOS仕様に含まれません。ファームウェア更新処理の開始と、更新の進捗状況の通知手段のみが含まれます。

廃止予定の扱い　*第1.11版にて追加*

仕様の曖昧さおよび誤解を修正可能にするため、廃止予定の手続きは最終的にこれらの項目を削除するために必要です。

廃止予定の処理はプロパティやメソッドおよび、パラメータ、定数、列挙体に対して同様に適用できます。

ある要素が廃止予定としてマークされているならば、サービスプロバイダは標準仕様の継続する二回までのマイナーリリースでその要素の機能をサポートする必要があります。ある要素が廃止予定としてマークされた後、標準仕様の3回目のリリース開始後、その要素の使用がE_DEPRECATEDの結果をもたらします。

ある要素が廃止予定としてマークされているならば、その要素のサポートは、廃止予定としてマークされた後の標準規格のメジャーリリースで標準規格から削除されます。

すべての廃止予定の要素と、初めに要素が廃止予定としてマークされたときに関連のあるバージョンは“付録F 廃止予定履歴”に記載されています。

ハイドラデバイスの検討 *第1.12版にて更新*

初期接続モデル

POSの周辺機器の標準規格の開発が始まった時、最もフレキシブルな方法は２層のプロセスを経由して周辺機器と通信できるアプリケーションを持つことであると決めました。Microsoft製COMプラットフォームが初めてサポートされたアーキテクチャなので、コントロールオブジェクトとサービスオブジェクトの名称が選ばれました。後にJavaが定義された時、その技術がWindowsワールドで定義された"object"を使わずにJava技術を用いました。（２層のプロセスの）名称はデバイスコントロールやデバイスサービスといったWindowsと緊密に連携したものであります。しかしながら、より高い概念レベルでの機能は同じままです。

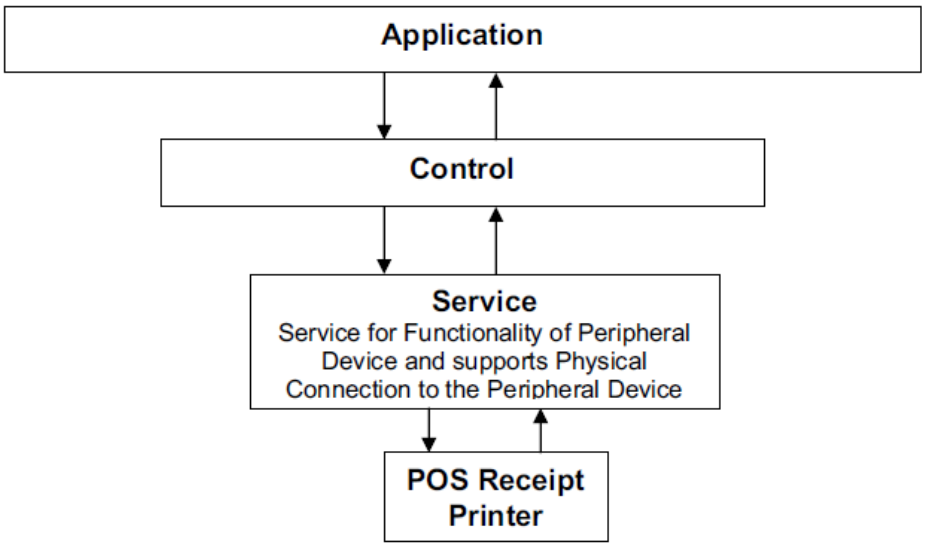
コントロールオブジェクト、またはデバイスコントロール

アプリケーションがデバイス名を使って周辺機器と通信できるように、通信ポートが開かれ、デバイス名が割り当てられるように、通常“パイプ”と呼ばれるような、ソフトウェアの薄い層が定義された。

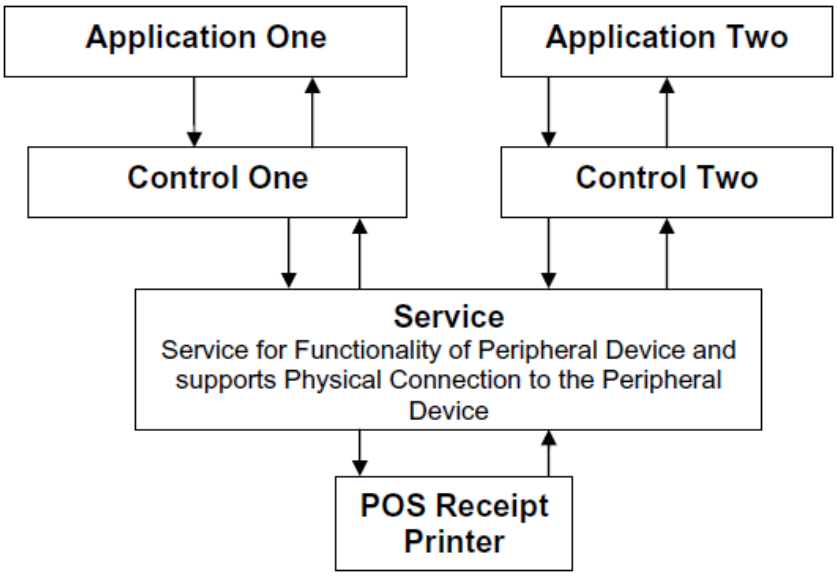
サービスオブジェクトまたはデバイスオブジェクト

アクセス処理、監視処理、プロセス処理を可能にするインターフェースを行うベンダー仕様コードは、周辺機器のすべての機能を表します。また、アプリケーションが周辺機器と接続するために必要な、プロパティ、メソッド、イベントの共通設定を明示します。

単一機能の周辺機器デバイスのプロセスは、とても明瞭です。最も単純化したシステムで、コントロールオブジェクトの１つのインスタンスは、サービスオブジェクトと接続するために作成されます。単純なPOSプリンタの例として：



単純ではありますが、別のレベルの複雑性が加わった、1つ以上のアプリケーションがデバイスを使わなければならない場合です。この場合、別のコントロールオブジェクトはインスタンス化され、周辺機器のサービスオブジェクトに対応するため生成されます。また、すべてのアプリケーションは周辺機器を共有することができる能力があることがわかっており（本例は、共有可能デバイスを仮定）、標準が提供する**ClaimDevice**と**ReleaseDevice**のメソッドを利用する必要があります。POSプリンタの例、下図参照。



本例では、RS-232 シリアル接続のため、周辺機器と接続されていている、個々のアプリケーションに関しては、デバイスに対して唯一の物理的な接続となっている点に、ご注意ください。協調するアプリケーションが利用された場合、デバイス共用の必要性が生じます。

マルチファンクション(ハイドラ)周辺機器デバイス



モデルは単一ユニット中で複数のデバイスクラス機能を含む周辺機器を対象とするために拡張する必要がありました。そのようなデバイスの一例は、ラインディスプレイ、ドロワー、MICR、あるいは他のデバイスをコントロールすることができる付加的な機能をもったPOSプリンタです。これらの周辺機器は、一つの体に繋がった多数の頭を持つギリシャ神話の動物にちなんで、“ハイドラ”周辺機器と呼ばれています。

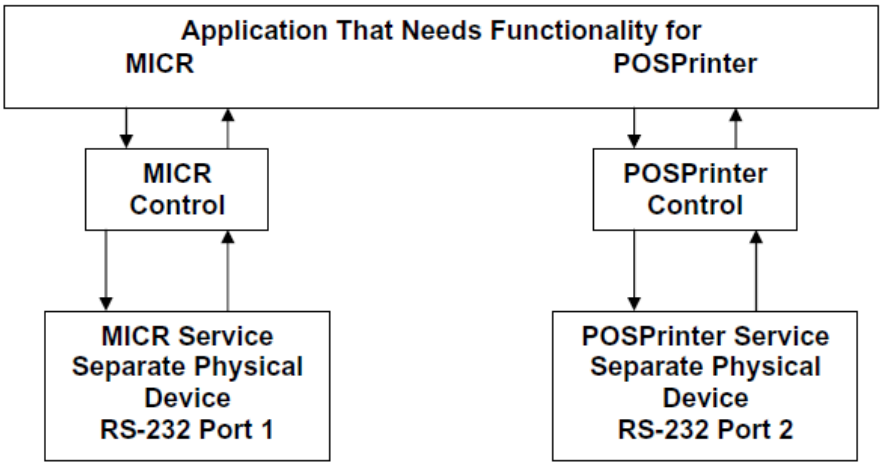
POS周辺機器の相互通信で、アプリケーションへのインターフェースは以下のいずれの場合も知る必要がありません。一方は物理的に複数の周辺機器が使われています。他方は複数の物理的周辺機器デバイスの機能を持ち合わせた、一つの物理的周辺機器が使われています。

複数の物理的周辺機器が存在する場所においては、複数の“パイプ”(例えばRS-232 シリアルポート)が必要です・・・一つ一つの物理的周辺機器のために。

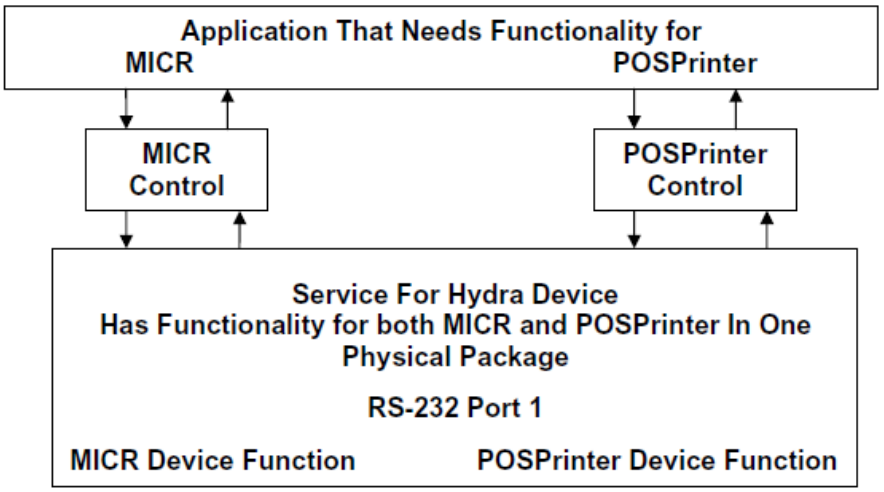
ハイドラ周辺機器においては、1つの"パイプ"が必要であり、そのパイプに接続された周辺機器の多岐にわたる全ての機能と通信するために使われています。

例えば、個別のPOSプリンタと個別のMICRデバイスが存在する場合と、別の例として、MICRリーダが組み込まれているPOSプリンタ（ハイドラデバイス）の場合を考えてみます。アプリケーション視点ではわかりません。アプリケーションは、物理的に存在するハードウェアデバイスがどんなタイプなのかを知ったり、考慮したりすることはありません。理想的には、デバイスは2つの実装のうちどちらであっても双方向通信をするために、同じアプリケーションコードを使うことが出来るべきです。

周辺機器の機能に基づいた実装に影響されて、異なるコードを作るべきものでもない。例えば：



この場合、MICRとPOSプリンタを作動させるアプリケーションは２つの分離したポートを使うが、2つのポートが使われている事実とは無関係に、MICRとPOSプリンタの機能と交信することにご注意ください。



この場合、MICRとPOSプリンタを作動させるアプリケーションは１つのポートのみを使うが、１つのポートのみが使われている事実とは無関係に、MICRとPOSプリンタの機能と交信することにご注意ください。ポートを制御したり、適正なインターフェースとの往復通信機能経路を決定したりするのは、ハイドラサービスオブジェクト次第です。

検討

アプリケーションのインターフェースに関して同じに動作する2つの相互接続技術を熱望する一方で、現実には問題が生じている。ハイドラの場合、特定のデバイスの機能のうちの一つであるエラー状態が他の機能の使用を妨げてしまいます。ハイドラでなく各々の周辺機器がまさに独立している場合、これは起きません。プリンタとMICRがハイドラになっている場合、例えば、用紙切れで作動しているプリンタが、MICRコードの読み込みを妨げる状態があります。“用紙切れ”のエラー状態はPOSプリンタのインターフェースを経由して通知されますが、MICRのインターフェースを経由する手段がありません。

ハイドラの場合、MICR読み込みを要求するアプリケーションに、個別のMICRの場合には得られなかった、エラーまたはステータスの状態が提示されます。これは、”ハングアップ”状態または、未解決のエラー状態が起こりえることを示します。

明らかに、エラー状態は、問題を警告し、解決させることができるように、MICR機能を使用しているアプリケーションに通知される必要があります。MICRアプリケーションに対して無意味な"用紙切れ"エラー通知するよりむしろ、一般的なE_FAILUREエラーが、問題について警告するためにアプリケーションへ返されるべきです。

MICRアプリケーションは、そして、状況の回復のためにエラー復帰の手順をおこないます。エラーメッセージは、MICRデバイス使用時に、差し迫った問題をオペレータに知らせるコンソールメッセージにより、エラー復帰操作を実現します。特定デバイスに関するオペレータの知識は、問題を訂正するのに使われます。

この場合、MICRはプリンタの一部であるという知識により、オペレーターの注意は“紙なし”状態に向かうでしょう。その解決策は、紙の交換であり、そして、プリンタ同様にMICRのエラー状態を解消することにもなります。どのようなサービスオブジェクトを作っても、個別周辺機器またはハイドラ周辺機器との相互通信の際に、アプリケーションに対しては"同じに見える"のでご注意ください。これを確実に達成するため、サービスオブジェクトは慎重に設計する必要があります。デバイスベンダーは、そのようなハイドラデバイスにアクセスがあったときに、存在しうるすべての制限事項および、独立デバイスでは起こりえないハイドラデバイス特有な非定常状態をユーザードキュメント中で明らかにすべきです。従って、ソフトを設計する時と同様にエラーの解決策を計画する場合、アプリケーション開発者は、個別とハイドラPOSデバイスの可能性を考えるべきです。

リテールコミュニケーションサービスデバイス　*第1.16版にて追加*

リテールコミュニケーションサービス(RCSD)デバイスについて

OpenPOSのデバイスに、リテールコミュニケーションサービスデバイス（以下、RSCDと省略する場合あり）を追加します。リテールコミュニケーションサービスデバイスとは、ユーザーとのコミュニケーションを行うためのデバイスを指します。

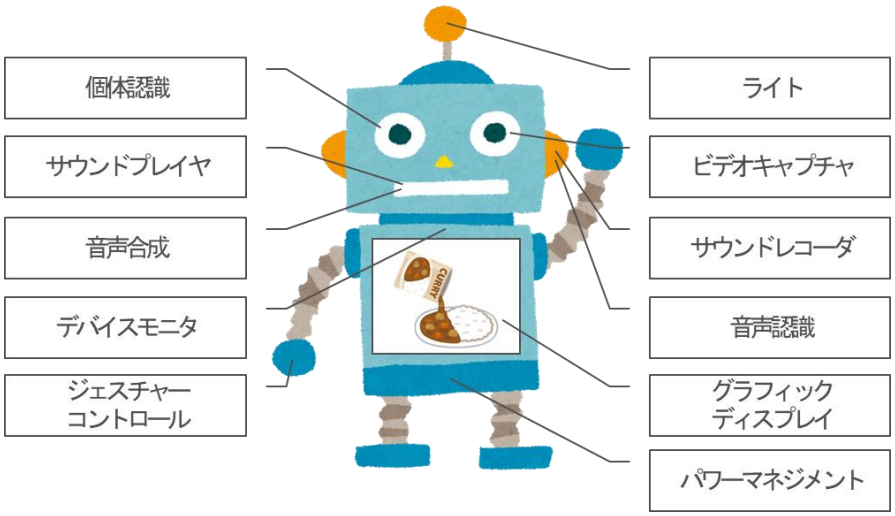
リテールコミュニケーションサービスデバイスは複数のデバイスで構成することができます。

例えばヒューマノイド型のロボットを指し、主に店舗の店頭や、ＰＯＳレジの傍に設置され、店舗システムやＰＯＳシステムと連携動作することができます。

リテールコミュニケーションサービスデバイスは様々な機能を有しています。また、リテールコミュニケーションサービスデバイスのすべてに、その様々な機能をすべて有しているわけではなく、リテールコミュニケーションサービスデバイスごとに有する機能は異なります。

ここで定義する機能は、リテイル分野での利用を想定したものである。デバイスの持つ専門的な機能は標準化範囲外とします。（例えば多言語発声や、クラウドサービス連携機能、無人化を想定したロボットを自走させる技術など）

デバイス	No	OPOS デバイス		主な機能	関連 OPOS デバイス
カメラ	1	ビデオキャプチャ	新設	・キャプチャ ・写真撮影 ・動画撮影	・ スキャナ ・ オブジェクト認識 ・ ハードトータル
	2	個体認識	新設	・個体認識（顔認識等）	
マイク	3	サウンドレコーダ	新設	・録音	・ ハードトータル
	4	音声認識	新設	・音声認識	
スピーカー	5	サウンドプレーヤ	新設	・音声再生	・ ハードトータル
	6	音声合成	新設	・音声合成	
アクチュエータ	7	ジェスチャーコントロール	新設	・ジョイント動作制御 ・ジョイント自動動作制御 ・モーション/ポーズ制御	・ ハードトータル
センサ	8	デバイスモニタ	新設	・デバイス監視	
ディスプレイ	9	グラフィックディスプレイ	新設	・画像表示 ・動画再生 ・ブラウザー表示	・ ハードトータル
LED	10	ライト	既存 拡張	・LED 制御	
バッテリー	11	パワーマネジメント	既存 拡張	・バッテリー監視	



OPOSコントロールの解説

章の構成

第2章では、共通のプロパティ、イベント、メソッドのインタフェースについて説明します。

後続の章では、以下のOpenPOSのActiveXコントロールについての、インタフェースの説明を行います。

ドロワー

ラインディスプレイ

ハードトータル

キーロック

磁気ストライプリーダ(MSR)

POSプリンタ

スキャナ（バーコードリーダ）

POSキーボード

コインディスペンサ（第1.2版にて自動釣り銭機に拡張）

秤

シグニチャ・キャプチャ

自動釣り銭機（第1.2版にて追加）

トーンインジケータ（第1.2版にて追加）

フィスカルプリンタ（第1.3版にて追加）

PINパッド（第1.3版にて追加）

リモートオーダディスプレイ（第1.3版にて追加）

バンプバー（第1.3版にて追加）

CAT（第1.4版にて追加）

ポイントカード機（第1.5版にて追加）

パワーマネージメント（第1.5版にて追加）

モーションセンサ（第1.7版にて追加）

チェックスキャナ（第1.7版にて追加）

スマートカードリーダライタ（第1.8版にて追加）

バイオメトリクス（第1.10版にて追加）

電子ジャーナル（第1.10版にて追加）

紙幣入金機（第1.11版にて追加）

硬貨入金機（第1.11版にて追加）

紙幣出金機（第1.11版にて追加）

イメージスキャナ（第1.11版にて追加）
電子バリューリーダライタ（第1.12版にて追加）
RFIDスキャナ（第1.12版にて追加）
ベルト（第1.12版にて追加）
ゲート（第1.12版にて追加）
アイテムディスペンサ（第1.12版にて追加）
ライト（第1.12版にて追加）
ビデオキャプチャ（第1.16版にて追加）
個体認識（第1.16版にて追加）
サウンドレコーダ（第1.16版にて追加）
音声認識（第1.16版にて追加）
サウンドフレーヤ（第1.16版にて追加）
音声合成（第1.16版にて追加）
ジェスチャーコントロール（第1.16版にて追加）
デバイスモニタ（第1.16版にて追加）
グラフィックディスプレイ（第1.16版にて追加）

付録A. システムレジストリのOPOS利用について詳細に説明します。

付録B. OPOSヘッダファイルがあります。

付録C. 技術的な詳細説明があります。

付録D. **ClaimDevice**および**ReleaseDevice**について

付録E. 本書の履歴

付録F XMLPOS　XML POS MAPPING REFERENCE

付録G. 削除予定履歴

付録H. システムマネジメントインフォメーション

データの型

以降の説明で用いるパラメータの型と戻り値の型は次のとおりです。

型	意味
BOOL	TRUE（ゼロ以外）とFALSE（ゼロ）を有効値とする整数。 COM IDL型：VARIANT_BOOL(short)。値はVARIANT_TRUE(-1)とVARIANT_FALSE(0)を取ります。 VARIANT型：VT_BOOL
BSTR	文字ストリング。文字列、NUL値（文字列の終了を示す）からなります。 COM IDL型：BSTR（unsigened short*）
VARIANT型：VT_BSTR	
BSTR*	文字ストリングのポインタ。 COM IDL型：BSTR*（unsigened short**）
VARIANT型：VT_BYREF | VT BSTR	
LONG	32ビット長の整数。 COM IDL型：long
VARIANT型：VT_14	
LONG*	32ビット長のポインタ。 COM IDL型：long*
VARIANT型：VT_BYREF | VT_14	
CURRENCY	第1.3版以降 通貨値。64ビット長の整数。小数点以下を4桁固定で定義。例えば、整数“1234567”は“123.4567”を意味する。 COM IDL型：CURRENCY(union tagCY)

“union tagCY”は、下記のような宣言方法

```
{
    struct{ long Hi; long Lo; };
__    int64 int64;
};
```

VARIANT型：VT_CY

CURRENCY*	第1.3版以降 通貨値のポインタ。 COM IDL型：CURRENCY*(union tagCY*)

VARIANT型：VT_BYREF | VT CY

第　１章　（　.ＮＥＴ版　）
はじめに

POS for .NETとは?　　*第1.11版にて更新*

POS for .NETは、Microsoft Windows Embedded for Point of Service（WEPOS）と呼ばれる、Microsoftのオペレーティングシステムであるwindowsベースのposシステムに、ハードウェアを容易に統合可能にするオープンなデバイスドライバアーキテクチャを提供するクラスライブラリです。.NET Frameworkアーキテクチャを利用した、Microsoftオペレーティングシステムを基盤とするUnifiedPOS標準の実装です。**注：この版のPOS for .NETの実装は、UnifiedPOS仕様書の1.8版に準拠しています。UnifiedPOS仕様書の1.10版のリリース後、POS for .NET仕様はUnifiedPOS仕様書の最新リリースに準拠、同期していく予定です。**

POS for .NETの目的は次のとおりです。

UnifiedPOS仕様書と既存のOPOS実装との密接な関連を維持しながら、.NET Framework上でWin32ベースPOSデバイスへのアクセス用アーキテクチャを定義すること

抽象化したUnifiedPOSデバイスと繋がる、様々なPOSアプリケーションのサポートが十分に行える、POSデバイスインターフェイスセットを定義すること。.NET Framework拡張のメリットによって、これらデバイスの管理が簡単になります

.NET Frameworkが提供するすべての豊富な機能ではないが、既存の（レガシーな）OPOSデバイスサービスが.NET Framework上で機能するマイグレーションパスを提供すること

POS for .NETの本リリースは次のものからなります。

UnifiedPOS Programmer's Guide：アプリケーション開発者およびハードウェア業者向け（本マニュアル）

POS for .NET SDK（完全なクラスライブラリを含む）：下記で入手可能です
www.microsoft.com/downloads/details.aspx?FamilyID=ADAA1129-5CB1-415E-B339-E508FCA55CA0&displaylang=en
SDKには、サンプルコードも含まれます

.NETアプリケーションでOPOSサービスを使用するための追加リソース：

「Shim」と呼ばれる、ソフトウェア/ミドルウェアの文書とコードのセットが入手可能です。これにより、POSアプリケーションが、.NETFrameworkのもとで、既存のOPOS命名規則を用いて実行することができます。「Shim」は、Microsoftがサポートする製品ではなく、.NET Frameworkの利点をすべて利用できるわけではありませんが、.最小限のコード変更で.NETPlatformを作成するためのものです。簡単な説明がこの付録にあります

マニュアル対象者

この章の対象読者は、POS周辺装置へのアクセスが必要な、Microsoft Windows Embedded for Point of Service（WEPOS）プラットホームにUnifiedPOS標準を実装したいアプリケーション開発者です。また、POS for .NETのサービスオブジェクトを作成したいプログラマ（デバイスメーカー）や、POS for .NETでのインターフェイスの方法の理解を深めたいアプリケーション開発者も対象になります。

この章は、読者が以下に精通していることを前提としています。

本マニュアルのUnifiedPOSデバイスの各章

POS周辺装置の一般的特徴

Microsoft .NET Frameworkの用語とアーキテクチャ

付録AのOPOSに関する実用的な知識。これによって、UnifiedPOSで実装された周辺機器の、Windows上での微妙な違いについて、特に詳しく知ることができます

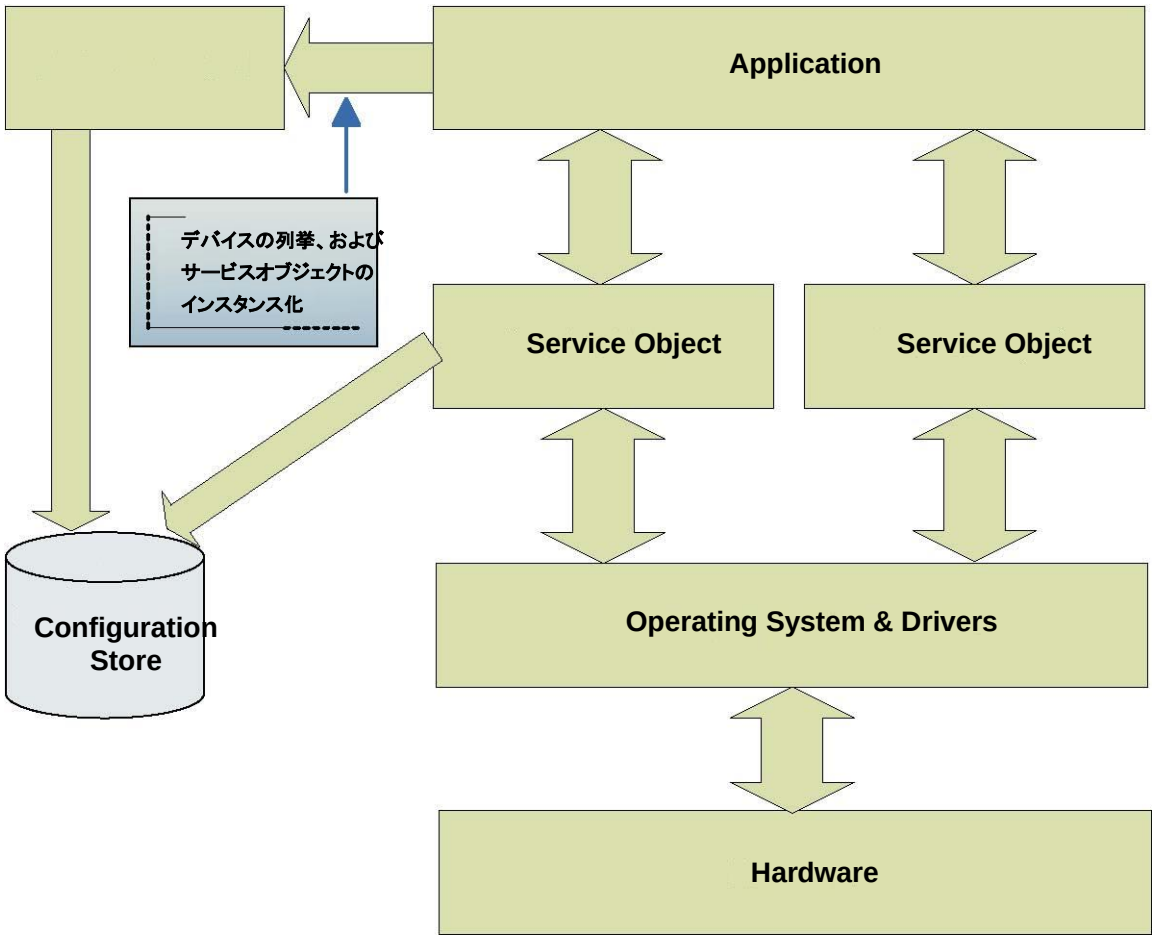
Visual Studio for .NETを含むMicrosoftの統合開発ツール、および.NETアプリケーション開発言語（例えば、Microsoft Visual Basic for .NET、C# for .NET、Visual C++ for .NETなど）のうち少なくとも1つに精通していること。注意すべき点ですが、POS for .NET向けのControl Programmer's Guide（CPG）はありません。コードのサンプルは、下記リンクのPOS for .NET SDKに含まれています
www.microsoft.com/downloads/details.aspx?FamilyID=ADAA1129-5CB1-415E-B339-E508FCA55CA0&displaylang=en

注：この付録の例では、メソッドのシグネチャを記載するとき、Visual C# .NETの構文を使用しています。

POS for .NETの概説

次の図は、POS for .NETの上位レベルのアーキテクチャを表します。アプリケーションは、**PosExplorer** APIを呼び出すことで、利用可能なPOS周辺機器を列挙し、それらのサービスオブジェクトをインスタンス化します。一旦PosExplorer APIによってサービスオブジェクトがインスタンス化されると、アプリケーションはオブジェクトと直接通信することができます。デバイス依存のサービスオブジェクトは、プロパティ、メソッド、およびイベントを通じて、周辺機器の状態と動作を提示します。

OPOS実装での動作とは異なり、POS for .NETでは、コントロールオブジェクトの概念はありません。その代わり、PosExplorer APIは、ある意味で、すべてのデバイスクラスに対する唯一のコントロールオブジェクトとして動作します。POS for .NETの設定が保持する、グローバルなconfiguration storeが存在します。PosExplorer APIは、システムに定義されている論理デバイスとその他の関連する情報を記憶域から読み込みます。また、物理デバイスとサービスオブジェクトの設定もconfiguration storeに保持されます。サービスオブジェクトは、そのプロパティ値をconfiguration storeとの間で読み書きできます。


Application
デバイスの列挙、および
サービスオブジェクトの
インスタンス化
Service Object
Service Object
Configuration
Store
Operating System & Drivers
Hardware

注目すべき重要な点は、既存の（レガシーな）OPOS CO/SOのソフトウェアコードと新しい.NETベースのクラスに依存するソフトウェアコードの両方を利用できることです。ただし、OPOSの既存の（レガシーな）サービスオブジェクトを使用する場合は、.NETベースのサービスの豊富な機能のすべてを使用することはできません。.NETの全ての機能を活用したサービスを持つ、.NETの全ての機能を活用するデバイスに対して、.NET POSアプリケーションを実装することが今後期待されます。

OPOSコントロールと同様、.NET SOベースクラスは、アプリケーションに対してプロパティ、メソッド、およびイベントを公開します。サービスオブジェクトは、POS for .NETによって定義されたデバイスクラスインターフェイスを実装するクラスです。Microsoftから供給されるインターフェイスは、アプリケーションが、UnifiedPOS標準で定義されたプロパティ、メソッド、およびイベントを利用して、POS周辺デバイスと相互通信するクラスインターフェイスを提供します。通信の応答は、メソッドの戻り値によって、また、パラメータ、プロパティ、およびイベントによってアプリケーションに渡されます。

POS for .NET関連用語の説明

デバイスクラス

デバイスクラスは、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。例えば、CashDrawer、POSPrinter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。例えば、POSプリンタには、キャッシュドロワー・キックアウト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、一体化された計量器が組み込まれているものもあります。

サービスオブジェクト（SO）

サービスオブジェクトは、POS for .NETによって定義されるデバイスクラスのインターフェイスを実装するクラスです。プロパティ、およびアプリケーションから呼び出されるメソッドを公開します

POS for .NETの主要な機能

POS周辺機器の.NETインターフェイス

POS for .NETは、UnifiedPOSの1.8版で定義されたデバイスについてのインターフェイスを定義します。1.9版、および1.10版（この版）で追加されたデバイスは、UnifiedPOSの将来の版で定義されます。

サービスオブジェクトのBase class

Base classは、デバイスの種類ごとにサービスオブジェクトの典型的な機能を実装します。このことが、SOの開発を単純化し、総合的な品質と一貫性を向上し、開発期間を削減するのに寄与します。

サービスオブジェクトのBasic class

Basic classは、サービスオブジェクトに共通の動作を実装します。共通の動作の例として、以下が挙げられます。

Open();
Claim();
DeviceEnabled();

プラグアンドプレイ

POS for .NETは、POS用周辺機器が、標準的なPCデスクトップ周辺機器と同等レベルのプラグアンドプレイ（PnP）Windowsアーキテクチャを利用できるようにします。PnPは、ハードウェア周辺機器がPCに接続されたときに、対応するドライバを（ほとんど）ユーザの介在なしに自動的にインストールするWindowsの機能です。現時点では、PnPはUnifiedPOSが実装する機能ではありませんが、PnPデバイスは、UnifiedPOSデバイス仕様に沿ってサポートされています。PnPサポートの詳細については、
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwue/html/ch11j.asp
をご覧ください。

標準化されたセットアップ

POS for .NETサービスオブジェクトの標準的なインストールとアンインストールの手順が提供されています。これにより、OPOSで必要とされるようなサービスをロードする特別なインストールプログラムは必要なくなります。

デバイスの列挙

POS for .NETサービスによって、システムにインストールされたすべてのPOS周辺機器のデバイスを列挙することが可能になります。

ソフトウェアベースのデバイススタティスティクス

UnifiedPOSで提供されるデバイススタティスティクスに加えて、ハードウェア独自のデバイススタティスティクスのための付加的なサポートが利用可能です。

OPOS（COMベースの）サービスオブジェクトのサポート

POS for .NETは、COMベースのサービスオブジェクトへの投資を無駄にしないためのライブラリの一部として、.NET、COM間の完全な相互運用性を提供します。ただし、POS for .NET V1.0では、下記のデバイスクラスについてのみ、1.10版リリースで相互運用機能が提供されます。

CashDrawer
(http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ccl/html/C_Microsoft_PointOfService_CashDrawer_ctor.asp)

CheckScanner
(http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ccl/html/C_Microsoft_PointOfService_CheckScanner_ctor.asp)

LineDisplay
(http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ccl/html/C_Microsoft_PointOfService_LineDisplay_ctor.asp)

MSR
(http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ccl/html/C_Microsoft_PointOfService_MSR_ctor.asp)

PINPad
(http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ccl/html/C_Microsoft_PointOfService_PINPad_ctor.asp)

POSKeyboard
(http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ccl/html/C_Microsoft_PointOfService_POSKeyboard_ctor.asp)

POSPrinter
(http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ccl/html/C_Microsoft_PointOfService_POSPrinter_ctor.asp)

Scanner
(http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ccl/html/C_Microsoft_PointOfService_Scanner_ctor.asp)

POS for .NETの後継バージョンにおけるOPOSサポートの詳細については、
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/xpehelp/html/startpage.asp
をご覧ください。

サービスオブジェクト検証プログラム

OPOSと比較した場合のPOS for .NETの進歩は、サードパーティの検証プログラムが利用可能なことです。このプログラムは、POS for .NETサービスオブジェクトのための一定レベルのテスト機能を提供します。現時点では、この相互運用性プログラムは、Microsoftによって管理されています。

プログラミング構築におけるOPOSとの主な差異

命名規則

ライブラリは、.NETクラスにPascal式命名規則を使用しており、メソッドのパラメータはキャメルケース（単語を大文字始まりにして記号や空白なしで連結する表記規則）になっています。これらの規則は、.NETの「クラスライブラリ開発者向けのデザインガイドライン」と一致しています。.NETの「クラスライブラリ開発者向けのデザインガイドライン」の詳細については、下記をご覧ください。

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpgenref/html/cpconnetframeworkdesignguidelines.asp

列挙体

POS for .NETは、広範囲にわたって列挙体を利用します。これには、いくつかの目的があります。列挙体は、アプリケーションとデバイスサービスオブジェクトの両方にインバウンドパラメータの使用を強要します。この方法の型チェックは、パラメータの領域オーバーや戻り値に起因するバグ防止に役立ちます。

さらに、列挙体の使用により、名前空間における大きな定数リストを使用しなくて済みます。ライブラリの開発では、定数データ型に対して範囲の有効性チェックを実施することが必要ですが、これは列挙体を使用することで自動的に行われます。

注意すべき点は、使用可能な列挙体の値には限りがあります。しかし、個々の数値の指定範囲が非常に大きくなる可能性があります。例として、*timeout*パラメータが挙げられます。指定可能な値は-1からInt32のサイズまでです。値−1は「無限に待つ」と解釈され、0からInt32のサイズまでのすべての値は、タイムアウトエラーが発生するまでの時間（ミリ秒単位）を表します。この場合の最善の方法は、定数（-1など）を「無限に待つ」と定義し、Int32の値を「無限に待たない条件」として利用することです。

以下の数ページは、現在のOPOSの定数とそれに対応するPOS for .NETの列挙体を示す表です。

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
OPOS_S_CLOSED	ControlState	enum	Closed
OPOS_S_IDLE	ControlState	enum	Idle
OPOS_S_BUSY	ControlState	enum	Busy
OPOS_S_ERROR	ControlState	enum	Error
OPOS_SUCCESS	ErrorCode	enum	Success
	ErrorCode	enum	Unspecified
OPOS_E_CLOSED	ErrorCode	enum	Closed
OPOS_E_CLAIMED	ErrorCode	enum	Claimed
OPOS_E_NOTCLAIMED	ErrorCode	enum	NotClaimed
OPOS_E_NOSERVICE	ErrorCode	enum	NoService
OPOS_E_DISABLED	ErrorCode	enum	Disabled
OPOS_E_ILLEGAL	ErrorCode	enum	Illegal
OPOS_E_NOHARDWARE	ErrorCode	enum	NoHardware
OPOS_E_OFFLINE	ErrorCode	enum	Offline
OPOS_E_NOEXIST	ErrorCode	enum	NoExist
OPOS_E_EXISTS	ErrorCode	enum	Exists
OPOS_E_FAILURE	ErrorCode	enum	Failure
OPOS_E_TIMEOUT	ErrorCode	enum	Timeout
OPOS_E_BUSY	ErrorCode	enum	Busy
OPOS_E_EXTENDED	ErrorCode	enum	Extended
OPOS_ESTATS_ERROR	PosCommon	Int32	ExtendedErrorStatistics
OPOS_CH_INTERNAL	HealthCheckLevel	enum	Internal
OPOS_CH_EXTERNAL	HealthCheckLevel	enum	External
OPOS_CH_INTERACTIVE	HealthCheckLevel	enum	Interactive
OPOS_PR_NONE	PowerReporting	enum	None
OPOS_PR_STANDARD	PowerReporting	enum	Standard
OPOS_PR_ADVANCED	PowerReporting	enum	Advanced
OPOS_PN_DISABLED	PowerNotification	enum	Disabled
OPOS_PN_ENABLED	PowerNotification	enum	Enabled
OPOS_PS_UNKNOWN	PowerState	enum	Unknown
OPOS_PS_ONLINE	PowerState	enum	Online
OPOS_PS_OFF	PowerState	enum	Off
OPOS_PS_OFFLINE	PowerState	enum	Offline
OPOS_PS_OFF_OFFLINE	PowerState	enum	OffOffline
OPOS_EL_OUTPUT	ErrorLocus	enum	Output
OPOS_EL_INPUT	ErrorLocus	enum	Input
OPOS_EL_INPUT_DATA	ErrorLocus	enum	InputData
OPOS_ER_RETRY	ErrorResponse	enum	Retry
OPOS_ER_CLEAR	ErrorResponse	enum	Clear
OPOS_ER_CONTINUEINPUT	ErrorResponse	enum	ContinueInput
OPOS_SUE_POWER_ONLINE	PosCommon	Int32	StatusPowerOnline
OPOS_SUE_POWER_OFF	PosCommon	Int32	StatusPowerOff
OPOS_SUE_POWER_OFFLINE	PosCommon	Int32	StatusPowerOffline
OPOS_SUE_POWER_OFF_OFFLINE	PosCommon	Int32	StatusPowerOffOffline
CFV_FIRMWARE_DIFFERENT	CompareFirmwareResult	enum	Different
CFV_FIRMWARE_NEWER	CompareFirmwareResult	enum	Newer
CFV_FIRMWARE_OLDER	CompareFirmwareResult	enum	Older
CFV_FIRMWARE_SAME	CompareFirmwareResult	enum	Same
CFV_FIRMWARE_UNKNOWN	CompareFirmwareResult	enum	Unknown
OPOS_SUE_UF_FAILED_DEV_OK	PosCommon	Int32	StatusUpdateFirmwareFailedDeviceOk
OPOS_SUE_UF_FAILED_DEV_UNRECOVERABLE	PosCommon	Int32	StatusUpdateFirmwareFailedDeviceUnrecoverable
OPOS_SUE_UF_FAILED_DEV_NEEDS_FIRMWARE	PosCommon	Int32	StatusUpdateFirmwareFailedDeviceNeedsFirmware
OPOS_SUE_UF_FAILED_DEV_UNKNOWN	PosCommon	Int32	StatusUpdateFirmwareFailedDeviceUnknown
OPOS_SUE_UF_COMPLETE	PosCommon	Int32	StatusUpdateFirmwareComplete

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
OPOS_SUE_UF_COMPLETE_DEV_NOT_RESTORED	PosCommon	Int32	StatusUpdateFirmwareCompleteDeviceNotRestored
OPOS_SUE_UF_PROGRESS + 1 to 100	PosCommon	Int32	StatusUpdateFirmwareProgress
OPOS_FOREVER	PosCommon	Int32	WaitForever
BB_UID_1	DeviceUnits	enum	Unit1
BB_UID_2	DeviceUnits	enum	Unit2
BB_UID_3	DeviceUnits	enum	Unit3
BB_UID_4	DeviceUnits	enum	Unit4
BB_UID_5	DeviceUnits	enum	Unit5
BB_UID_6	DeviceUnits	enum	Unit6
BB_UID_7	DeviceUnits	enum	Unit7
BB_UID_8	DeviceUnits	enum	Unit8
BB_UID_9	DeviceUnits	enum	Unit9
BB_UID_10	DeviceUnits	enum	Unit10
BB_UID_11	DeviceUnits	enum	Unit11
BB_UID_12	DeviceUnits	enum	Unit12
BB_UID_13	DeviceUnits	enum	Unit13
BB_UID_14	DeviceUnits	enum	Unit14
BB_UID_15	DeviceUnits	enum	Unit15
BB_UID_16	DeviceUnits	enum	Unit16
BB_UID_17	DeviceUnits	enum	Unit17
BB_UID_18	DeviceUnits	enum	Unit18
BB_UID_19	DeviceUnits	enum	Unit19
BB_UID_20	DeviceUnits	enum	Unit20
BB_UID_21	DeviceUnits	enum	Unit21
BB_UID_22	DeviceUnits	enum	Unit22
BB_UID_23	DeviceUnits	enum	Unit23
BB_UID_24	DeviceUnits	enum	Unit24
BB_UID_25	DeviceUnits	enum	Unit25
BB_UID_26	DeviceUnits	enum	Unit26
BB_UID_27	DeviceUnits	enum	Unit27
BB_UID_28	DeviceUnits	enum	Unit28
BB_UID_29	DeviceUnits	enum	Unit29
BB_UID_30	DeviceUnits	enum	Unit30
BB_UID_31	DeviceUnits	enum	Unit31
BB_UID_32	DeviceUnits	enum	Unit32
BB_DE_KEY	BumpBar	Int32	DataEventKey
CASH_SUE_DRAWERCLOSED	CashDrawerStatus	enum	Closed
CASH_SUE_DRAWEROPEN	CashDrawerStatus	enum	Open
CAT_PAYMENT_LUMP	PaymentCondition	enum	Lump
CAT_PAYMENT_BONUS_1	PaymentCondition	enum	Bonus1
CAT_PAYMENT_BONUS_2	PaymentCondition	enum	Bonus2
CAT_PAYMENT_BONUS_3	PaymentCondition	enum	Bonus3
CAT_PAYMENT_BONUS_4	PaymentCondition	enum	Bonus4
CAT_PAYMENT_BONUS_5	PaymentCondition	enum	Bonus5
CAT_PAYMENT_INSTALLMENT_1	PaymentCondition	enum	Installment1
CAT_PAYMENT_INSTALLMENT_2	PaymentCondition	enum	Installment2
CAT_PAYMENT_INSTALLMENT_3	PaymentCondition	enum	Installment3
CAT_PAYMENT_BONUS_COMBINATION_1	PaymentCondition	enum	BonusCombination1
CAT_PAYMENT_BONUS_COMBINATION_2	PaymentCondition	enum	BonusCombination2
CAT_PAYMENT_BONUS_COMBINATION_3	PaymentCondition	enum	BonusCombination3
CAT_PAYMENT_BONUS_COMBINATION_4	PaymentCondition	enum	BonusCombination4
CAT_PAYMENT_REVOLVING	PaymentCondition	enum	Revolving
CAT_PAYMENT_DEBIT	PaymentCondition	enum	Debit
CAT_TRANSACTION_SALES	CreditTransactionType	enum	Sales
CAT_TRANSACTION_VOID	CreditTransactionType	enum	Void
CAT_TRANSACTION_REFUND	CreditTransactionType	enum	Refund
CAT_TRANSACTION_VOIDPRESALES	CreditTransactionType	enum	VoidPreSales
CAT_TRANSACTION_COMPLETION	CreditTransactionType	enum	Completion
CAT_TRANSACTION_PRESALES	CreditTransactionType	enum	PreSales
CAT_TRANSACTION_CHECKCARD	CreditTransactionType	enum	CheckCard
CAT_MEDIA_UNSPECIFIED	PaymentMedia	enum	Unspecified

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
CAT_MEDIA_NONDEFINE	PaymentMedia		
CAT_MEDIA_CREDIT	PaymentMedia	enum	Credit
CAT_MEDIA_DEBIT	PaymentMedia	enum	Debit
OPOS_PS_UNKNOWNECAT_CENTERERROR	Cat	Int32	ExtendedErrorCenterError
OPOS_PS_UNKNOWNECAT_COMMANDERROR	Cat	Int32	ExtendedErrorCommandError
OPOS_PS_UNKNOWNECAT_RESET	Cat	Int32	ExtendedErrorReset
OPOS_PS_UNKNOWNECAT_COMMUNICATIONERRO R	Cat	Int32	ExtendedErrorCommunicationError
OPOS_PS_UNKNOWNECAT_DAILYLOGOVERFLOW	Cat	Int32	ExtendedErrorDailyLogOverflow
CAT_DL_NONE	CatLogs	enum	None
CAT_DL_REPORTING	CatLogs	enum	Reporting
CAT_DL_SETTLEMENT	CatLogs	enum	Settlement
CAT_DL_REPORTING_SETTLEMENT	CatLogs	enum	ReportingAndSettlement
CHAN_STATUS_OK	CashChangerStatus	enum	OK
CHAN_STATUS_EMPTY	CashChangerStatus	enum	Empty
CHAN_STATUS_NEAREMPTY	CashChangerStatus	enum	NearEmpty
CHAN_STATUS_EMPTYOK	CashChangerStatus		
	CashChangerFullStatus	enum	OK
CHAN_STATUS_FULL	CashChangerFullStatus	enum	Full
CHAN_STATUS_NEARFULL	CashChangerFullStatus	enum	NearFull
CHAN_STATUS_FULLOK	CashChangerFullStatus		
CHAN_STATUS_JAM	CashChangerStatus	enum	Jam
CHAN_STATUS_JAMOK	CashChangerStatus		
CHAN_STATUS_ASYNC	CashChanger	Int32	StatusAsync
CHAN_STATUS_DEPOSIT_START	CashDepositStatus	enum	Start
CHAN_STATUS_DEPOSIT_END	CashDepositStatus	enum	End
CHAN_STATUS_DEPOSIT_NONE	CashDepositStatus	enum	None
CHAN_STATUS_DEPOSIT_COUNT	CashDepositStatus	enum	Count
CHAN_STATUS_DEPOSIT_JAM	CashDepositStatus	enum	Jam
CHAN_DEPOSIT_CHANGE	CashDepositAction	enum	Change
CHAN_DEPOSIT_NOCHANGE	CashDepositAction	enum	NoChange
CHAN_DEPOSIT_REPAY	CashDepositAction	enum	Repay
CHAN_DEPOSIT_PAUSE	CashDepositPause	enum	Pause
CHAN_DEPOSIT_RESTART	CashDepositPause	enum	Restart
OPOS_ECHAN_OVERDISPENSE	CashChanger	Int32	ExtendedErrorOverDispense
CHK_CCL_MONO	CheckColors	enum	Mono
CHK_CCL_GRAYSCALE	CheckColors	enum	GrayScale
CHK_CCL_16	CheckColors	enum	Color16
CHK_CCL_256	CheckColors	enum	Color256
CHK_CCL_FULL	CheckColors	enum	Full
CHK_CIF_NATIVE	CheckImageFormats	enum	Native
CHK_CIF_TIFF	CheckImageFormats	enum	Tiff
CHK_CIF_BMP	CheckImageFormats	enum	Bmp
CHK_CIF_JPEG	CheckImageFormats	enum	Jpeg
CHK_CIF_GIF	CheckImageFormats	enum	Gif
CHK_CL_MONO	CheckColors	enum	Mono
CHK_CL_GRAYSCALE	CheckColors	enum	GrayScale
CHK_CL_16	CheckColors	enum	Color16
CHK_CL_256	CheckColors	enum	Color256
CHK_CL_FULL	CheckColors	enum	Full
CHK_IF_NATIVE	CheckImageFormats	enum	Native
CHK_IF_TIFF	CheckImageFormats	enum	Tiff
CHK_IF_BMP	CheckImageFormats	enum	Bmp
CHK_IF_JPEG	CheckImageFormats	enum	Jpeg
CHK_IF_GIF	CheckImageFormats	enum	Gif

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
CHK_IMS_EMPTY	ImageMemoryStatus	enum	Empty
CHK_IMS_OK	ImageMemoryStatus	enum	OK
CHK_IMS_FULL	ImageMemoryStatus	enum	Full
CHK_MM_DOTS	MapMode	enum	Dots
CHK_MM_TWIPS	MapMode	enum	Twips
CHK_MM_ENGLISH	MapMode	enum	English
CHK_MM_METRIC	MapMode	enum	Metric
CHK_CLR_ALL	CheckImageClear	enum	All
CHK_CLR_BY_FILEID	CheckImageClear	enum	FileId
CHK_CLR_BY_FILEINDEX	CheckImageClear	enum	FileIndex
CHK_CLR_BY_IMAGETAGDATA	CheckImageClear	enum	ImageTagData
CHK_CROP_AREA_ENTIRE_IMAGE	CheckScanner	Int32	CropEntireImage
CHK_CROP_AREA_RESET_ALL	CheckScanner	Int32	CropResetAll
CHK_CROP_AREA_RIGHT	CheckScanner	Int32	CropRight
CHK_CROP_AREA_BOTTOM	CheckScanner	Int32	CropBottom
CHK_LOCATE_BY_FILEID	CheckImageLocate	enum	FileId
CHK_LOCATE_BY_FILEINDEX	CheckImageLocate	enum	FileIndex
CHK_LOCATE_BY_IMAGETAGDATA	CheckImageLocate	enum	ImageTagData
CHK_SUE_SCANCOMPLETE	CheckScannerStatus	enum	ScanComplete
OPOS_ECHK_NOCHECK	CheckScanner	Int32	ExtendedErrorNoCheck
OPOS_ECHK_CHECK	CheckScanner	Int32	ExtendedErrorCheck
OPOS_ECHK_NOROOM	CheckScanner	Int32	ExtendedErrorNoRoom
COIN_STATUS_OK	CoinDispenserStatus	enum	OK
COIN_STATUS_EMPTY	CoinDispenserStatus	enum	Empty
COIN_STATUS_NEAREMPTY	CoinDispenserStatus	enum	NearEmpty
COIN_STATUS_JAM	CoinDispenserStatus	enum	Jam
DISP_CB_NOBLINK	DisplayBlink	enum	None
DISP_CB_BLINKALL	DisplayBlink	enum	All
DISP_CB_BLINKEACH	DisplayBlink	enum	Each
DISP_CCS_NUMERIC	CharacterSetCapability	enum	Numeric
DISP_CCS_ALPHA	CharacterSetCapability	enum	Alpha
DISP_CCS_ASCII	CharacterSetCapability	enum	Ascii
DISP_CCS_KANA	CharacterSetCapability	enum	Kana
DISP_CCS_KANJI	CharacterSetCapability	enum	Kanji
DISP_CCS_UNICODE	CharacterSetCapability	enum	Unicode
DISP_CCT_NONE	DisplayCursors	enum	None
DISP_CCT_FIXED	DisplayCursors	enum	Fixed
DISP_CCT_BLOCK	DisplayCursors	enum	Block
DISP_CCT_HALFBLOCK	DisplayCursors	enum	HalfBlock
DISP_CCT_UNDERLINE	DisplayCursors	enum	Underline
DISP_CCT_REVERSE	DisplayCursors	enum	Reverse
DISP_CCT_OTHER	DisplayCursors	enum	Other
DISP_CCT_BLINK	DisplayCursors	enum	Blink
DISP_CRB_NONE	DisplayReadBack	enum	None
DISP_CRB_SINGLE	DisplayReadBack	enum	Single
DISP_CR_NONE	DisplayReverse	enum	None
DISP_CR_REVERSEALL	DisplayReverse	enum	All
DISP_CR_REVERSEEACH	DisplayReverse	enum	Each
DISP_CS_UNICODE	PosCommon	Int32	CharacterSetUnicode
DISP_CS_ASCII	PosCommon	Int32	CharacterSetAscii
DISP_CS_WINDOWS	PosCommon	Int32	
DISP_CS_ANSI	PosCommon	Int32	CharacterSetAnsi
DISP_CT_NONE	DisplayCursors	enum	None
DISP_CT_FIXED	DisplayCursors	enum	Fixed

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
DISP_CT_BLOCK	DisplayCursors	enum	Block
DISP_CT_HALFBLOCK	DisplayCursors	enum	HalfBlock
DISP_CT_UNDERLINE	DisplayCursors	enum	Underline
DISP_CT_REVERSE	DisplayCursors	enum	Reverse
DISP_CT_OTHER	DisplayCursors	enum	Other
DISP_CT_BLINK	DisplayCursors	enum	Blink
DISP_MT_NONE	DisplayMarqueeType	enum	None
DISP_MT_UP	DisplayMarqueeType	enum	Up
DISP_MT_DOWN	DisplayMarqueeType	enum	Down
DISP_MT_LEFT	DisplayMarqueeType	enum	Left
DISP_MT_RIGHT	DisplayMarqueeType	enum	Right
DISP_MT_INIT	DisplayMarqueeType	enum	Init
DISP_MF_WALK	DisplayMarqueeFormat	enum	Walk
DISP_MF_PLACE	DisplayMarqueeFormat	enum	Place
DISP_DT_NORMAL	DisplayTextMode	enum	Normal
DISP_DT_BLINK	DisplayTextMode	enum	Blink
DISP_DT_REVERSE	DisplayTextMode	enum	Reverse
DISP_DT_BLINK_REVERSE	DisplayTextMode	enum	BlinkReverse
DISP_ST_UP	DisplayScrollText	enum	Up
DISP_ST_DOWN	DisplayScrollText	enum	Down
DISP_ST_LEFT	DisplayScrollText	enum	Left
DISP_ST_RIGHT	DisplayScrollText	enum	Right
DISP_SD_OFF	DisplaySetDescriptor	enum	Off
DISP_SD_ON	DisplaySetDescriptor	enum	On
DISP_SD_BLINK	DisplaySetDescriptor	enum	Blink
DISP_BM_ASIS	LineDisplay	Int32	DisplayBitmapAsIs
DISP_BM_LEFT	LineDisplay	Int32	DisplayBitmapLeft
DISP_BM_CENTER	LineDisplay	Int32	DisplayBitmapCenter
DISP_BM_RIGHT	LineDisplay	Int32	DisplayBitmapRight
DISP_BM_TOP	LineDisplay	Int32	DisplayBitmapTop
DISP_BM_BOTTOM	LineDisplay	Int32	DisplayBitmapBottom
OPOS_EDISP_TOOBIG	LineDisplay	Int32	ExtendedErrorTooBig
OPOS_EDISP_BADFORMAT	LineDisplay	Int32	ExtendedErrorBadFormat
FPTR_S_JOURNAL	FiscalPrinterStations	enum	Journal
FPTR_S_RECEIPT	FiscalPrinterStations	enum	Receipt
FPTR_S_SLIP	FiscalPrinterStations	enum	Slip
FPTR_S_JOURNAL_RECEIPT	FiscalPrinterStations	enum	JournalReceipt
	FiscalPrinterStations	enum	JournalSlip
	FiscalPrinterStations	enum	ReceiptSlip
FPTR_AC_BRC	FiscalCurrency	enum	BrazilianCruceiro
FPTR_AC_BGL	FiscalCurrency	enum	BulgarianLev
FPTR_AC_EUR	FiscalCurrency	enum	Euro
FPTR_AC_GRD	FiscalCurrency	enum	GreekDrachma
FPTR_AC_HUF	FiscalCurrency	enum	HungarianForint
FPTR_AC_ITL	FiscalCurrency	enum	ItalianLira
FPTR_AC_PLZ	FiscalCurrency	enum	PolishZloty
FPTR_AC_ROL	FiscalCurrency	enum	RomanianLeu
FPTR_AC_RUR	FiscalCurrency	enum	RussianRouble
FPTR_AC_TRL	FiscalCurrency	enum	TurkishLira
FPTR_CID_FIRST	FiscalContractorId	enum	First
FPTR_CID_SECOND	FiscalContractorId	enum	Second
FPTR_CID_SINGLE	FiscalContractorId	enum	Single
FPTR_CC_BRAZIL	FiscalCountryCodes	enum	Brazil
FPTR_CC_GREECE	FiscalCountryCodes	enum	Greece
FPTR_CC_HUNGARY	FiscalCountryCodes	enum	Hungary
FPTR_CC_ITALY	FiscalCountryCodes	enum	Italy
FPTR_CC_POLAND	FiscalCountryCodes	enum	Poland
FPTR_CC_TURKEY	FiscalCountryCodes	enum	Turkey
FPTR_CC_RUSSIA	FiscalCountryCodes	enum	Russia

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
FPTR_CC_BULGARIA	FiscalCountryCodes	enum	Bulgaria
FPTR_CC_ROMANIA	FiscalCountryCodes	enum	Romania
FPTR_DT_CONF	FiscalDateType	enum	Configuration
FPTR_DT_EOD	FiscalDateType	enum	EndOfDay
FPTR_DT_RESET	FiscalDateType	enum	Reset
FPTR_DT_RTC	FiscalDateType	enum	RealTimeClock
FPTR_DT_VAT	FiscalDateType	enum	VatChange
FPTR_EL_NONE	FiscalErrorLevel	enum	None
FPTR_EL_RECOVERABLE	FiscalErrorLevel	enum	Recoverable
FPTR_EL_FATAL	FiscalErrorLevel	enum	Fatal
FPTR_EL_BLOCKED	FiscalErrorLevel	enum	Blocked
FPTR_PS_MONITOR	FiscalPrinterState	enum	Monitor
FPTR_PS_FISCAL_RECEIPT	FiscalPrinterState	enum	FiscalReceipt
FPTR_PS_FISCAL_RECEIPT_TOTAL	FiscalPrinterState	enum	FiscalReceiptTotal
FPTR_PS_FISCAL_RECEIPT_ENDING	FiscalPrinterState	enum	FiscalReceiptEnding
FPTR_PS_FISCAL_DOCUMENT	FiscalPrinterState	enum	FiscalDocument
FPTR_PS_FIXED_OUTPUT	FiscalPrinterState	enum	FixedOutput
FPTR_PS_ITEM_LIST	FiscalPrinterState	enum	ItemList
FPTR_PS_LOCKED	FiscalPrinterState	enum	Locked
FPTR_PS_NONFISCAL	FiscalPrinterState	enum	NonFiscal
FPTR_PS_REPORT	FiscalPrinterState	enum	Report
FPTR_RS_RECEIPT	FiscalReceiptStation	enum	Receipt
FPTR_RS_SLIP	FiscalReceiptStation	enum	Slip
FPTR_RT_CASH_IN	FiscalReceiptType	enum	CashIn
FPTR_RT_CASH_OUT	FiscalReceiptType	enum	CashOut
FPTR_RT_GENERIC	FiscalReceiptType	enum	Generic
FPTR_RT_SALES	FiscalReceiptType	enum	Sales
FPTR_RT_SERVICE	FiscalReceiptType	enum	Service
FPTR_RT_SIMPLE_INVOICE	FiscalReceiptType	enum	SimpleInvoice
FPTR_MT_ADVANCE	FiscalMessageType	enum	Advance
FPTR_MT_ADVANCE_PAID	FiscalMessageType	enum	AdvancePaid
FPTR_MT_AMOUNT_TO_BE_PAID	FiscalMessageType	enum	AmountToBePaid
FPTR_MT_AMOUNT_TO_BE_PAID_BACK	FiscalMessageType	enum	AmountToBePaidBack
FPTR_MT_CARD	FiscalMessageType	enum	Card
FPTR_MT_CARD_NUMBER	FiscalMessageType	enum	CardNumber
FPTR_MT_CARD_TYPE	FiscalMessageType	enum	CardType
FPTR_MT_CASH	FiscalMessageType	enum	Cash
FPTR_MT_CASHIER	FiscalMessageType	enum	Cashier
FPTR_MT_CASH_REGISTER_NUMBER	FiscalMessageType	enum	CashRegisterNumber
FPTR_MT_CHANGE	FiscalMessageType	enum	Change
FPTR_MT_CHEQUE	FiscalMessageType	enum	Cheque
FPTR_MT_CLIENT_NUMBER	FiscalMessageType	enum	ClientNumber
FPTR_MT_CLIENT_SIGNATURE	FiscalMessageType	enum	ClientSignature
FPTR_MT_COUNTER_STATE	FiscalMessageType	enum	CounterState
FPTR_MT_CREDIT_CARD	FiscalMessageType	enum	CreditCard
FPTR_MT_CURRENCY	FiscalMessageType	enum	Currency
FPTR_MT_CURRENCY_VALUE	FiscalMessageType	enum	CurrencyValue
FPTR_MT_DEPOSIT	FiscalMessageType	enum	Deposit
FPTR_MT_DEPOSIT_RETURNED	FiscalMessageType	enum	DepositReturned
FPTR_MT_DOT_LINE	FiscalMessageType	enum	DotLine
FPTR_MT_DRIVER_NUMB	FiscalMessageType	enum	DriverNumber
FPTR_MT_EMPTY_LINE	FiscalMessageType	enum	EmptyLine
FPTR_MT_FREE_TEXT	FiscalMessageType	enum	FreeText
FPTR_MT_FREE_TEXT_WITH_DAY_LIMIT	FiscalMessageType	enum	FreeTextWithDayLimit
FPTR_MT_GIVEN_DISCOUNT	FiscalMessageType	enum	GivenDiscount
FPTR_MT_LOCAL_CREDIT	FiscalMessageType	enum	LocalCredit
FPTR_MT_MILEAGE_KM	FiscalMessageType	enum	MileageKilometers
FPTR_MT_NOTE	FiscalMessageType	enum	Note
FPTR_MT_PAID	FiscalMessageType	enum	Paid
FPTR_MT_PAY_IN	FiscalMessageType	enum	PayIn
FPTR_MT_POINT_GRANTED	FiscalMessageType	enum	PointGranted
FPTR_MT_POINTS_BONUS	FiscalMessageType	enum	PointsBonus
FPTR_MT_POINTS_RECEIPT	FiscalMessageType	enum	PointsReceipt

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
FPTR_MT_POINTS_TOTAL	FiscalMessageType	enum	PointsTotal
FPTR_MT_PROFITED	FiscalMessageType	enum	Profited
FPTR_MT_RATE	FiscalMessageType	enum	Rate
FPTR_MT_REGISTER_NUMB	FiscalMessageType	enum	RegisterNumber
FPTR_MT_SHIFT_NUMBER	FiscalMessageType	enum	ShiftNumber
FPTR_MT_STATE_OF_AN_ACCOUNT	FiscalMessageType	enum	StateOfAnAccount
FPTR_MT_SUBSCRIPTION	FiscalMessageType	enum	Subscription
FPTR_MT_TABLE	FiscalMessageType	enum	Table
FPTR_MT_THANK_YOU_FOR_LOYALTY	FiscalMessageType	enum	ThankYouForLoyalty
FPTR_MT_TRANSACTION_NUMB	FiscalMessageType	enum	TransactionNumber
FPTR_MT_VALID_TO	FiscalMessageType	enum	ValidTo
FPTR_MT_VOUCHER	FiscalMessageType	enum	Voucher
FPTR_MT_VOUCHER_PAID	FiscalMessageType	enum	VoucherPaid
FPTR_MT_VOUCHER_VALUE	FiscalMessageType	enum	VoucherValue
FPTR_MT_WITH_DISCOUNT	FiscalMessageType	enum	WithDiscount
FPTR_MT_WITHOUT_UPLIFT	FiscalMessageType	enum	WithoutUplift
FPTR_SS_FULL_LENGTH	FiscalSlipSelection	enum	FullLength
FPTR_SS_VALIDATION	FiscalSlipSelection	enum	Validation
FPTR_TT_DOCUMENT	FiscalTotalizerType	enum	Document
FPTR_TT_DAY	FiscalTotalizerType	enum	Day
FPTR_TT_RECEIPT	FiscalTotalizerType	enum	Receipt
FPTR_TT_GRAND	FiscalTotalizerType	enum	Grand
FPTR_GD_CURRENT_TOTAL	FiscalData	enum	CurrentTotal
FPTR_GD_DAILY_TOTAL	FiscalData	enum	DailyTotal
FPTR_GD_RECEIPT_NUMBER	FiscalData	enum	ReceiptNumber
FPTR_GD_REFUND	FiscalData	enum	Refund
FPTR_GD_NOT_PAID	FiscalData	enum	NotPaid
FPTR_GD_MID_VOID	FiscalData	enum	NumberOfVoidedReceipts
FPTR_GD_Z_REPORT	FiscalData	enum	ZReport
FPTR_GD_GRAND_TOTAL	FiscalData	enum	GrandTotal
FPTR_GD_PRINTER_ID	FiscalData	enum	PrinterId
FPTR_GD_FIRMWARE	FiscalData	enum	Firmware
FPTR_GD_RESTART	FiscalData	enum	Restart
FPTR_GD_REFUND_VOID	FiscalData	enum	RefundVoid
FPTR_GD_NUMB_CONFIG_BLOCK	FiscalData	enum	NumberOfConfigurationBlocks
FPTR_GD_NUMB_CURRENCY_BLOCK	FiscalData	enum	NumberOfCurrencyBlocks
FPTR_GD_NUMB_HDR_BLOCK	FiscalData	enum	NumberOfHeaderBlocks
FPTR_GD_NUMB_RESET_BLOCK	FiscalData	enum	NumberOfResetBlocks
FPTR_GD_NUMB_VAT_BLOCK	FiscalData	enum	NumberOfVatBlocks
FPTR_GD_FISCAL_DOC	FiscalData	enum	FiscalDocument
FPTR_GD_FISCAL_DOC_VOID	FiscalData	enum	FiscalDocumentVoid
FPTR_GD_FISCAL_REC	FiscalData	enum	FiscalReceipt
FPTR_GD_FISCAL_REC_VOID	FiscalData	enum	FiscalReceiptVoid
FPTR_GD_NONFISCAL_DOC	FiscalData	enum	NonFiscalDocument
FPTR_GD_NONFISCAL_DOC_VOID	FiscalData	enum	NonFiscalDocumentVoid
FPTR_GD_NONFISCAL_REC	FiscalData	enum	NonFiscalReceipt
FPTR_GD_SIMP_INVOICE	FiscalData	enum	SimplifiedInvoice
FPTR_GD_TENDER	FiscalData	enum	Tender
FPTR_GD_LINECOUNT	FiscalData	enum	LineCount
FPTR_GD_DESCRIPTION_LENGTH	FiscalData	enum	DescriptionLength
FPTR_PDL_CASH	FiscalPrinter	Int32	PaymentDescriptionCash
FPTR_PDL_CHEQUE	FiscalPrinter	Int32	PaymentDescriptionCheque
FPTR_PDL_CHITTY	FiscalPrinter	Int32	PaymentDescriptionChitty
FPTR_PDL_COUPON	FiscalPrinter	Int32	PaymentDescriptionCoupon
FPTR_PDL_CURRENCY	FiscalPrinter	Int32	PaymentDescriptionCurrency
FPTR_PDL_DRIVEN_OFF	FiscalPrinter	Int32	PaymentDescriptionDrivenOff
FPTR_PDL_EFT_IMPRINTER	FiscalPrinter	Int32	PaymentDescriptionEftImprinter
FPTR_PDL_EFT_TERMINAL	FiscalPrinter	Int32	PaymentDescriptionEftTerminal
FPTR_PDL_TERMINAL_IMPRINTER	FiscalPrinter	Int32	PaymentDescriptionTerminalImprinter
FPTR_PDL_FREE_GIFT	FiscalPrinter	Int32	PaymentDescriptionFreeGift
FPTR_PDL_GIRO	FiscalPrinter	Int32	PaymentDescriptionGiro
FPTR_PDL_HOME	FiscalPrinter	Int32	PaymentDescriptionHome
FPTR_PDL_IMPRINTER_WITH_ISSUER	FiscalPrinter	Int32	PaymentDescriptionImprinterWithIssuer
FPTR_PDL_LOCAL_ACCOUNT	FiscalPrinter	Int32	PaymentDescriptionLocalAccount
FPTR_PDL_LOCAL_ACCOUNT_CARD	FiscalPrinter	Int32	PaymentDescriptionLocalAccountCard

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
FPTR_PDL_PAY_CARD	FiscalPrinter	Int32	PaymentDescriptionPayCard
FPTR_PDL_PAY_CARD_MANUAL	FiscalPrinter	Int32	PaymentDescriptionPayCardManual
FPTR_PDL_PREPAY	FiscalPrinter	Int32	PaymentDescriptionPrepay
FPTR_PDL_PUMP_TEST	FiscalPrinter	Int32	PaymentDescriptionPumpTest
FPTR_PDL_SHORT_CREDIT	FiscalPrinter	Int32	PaymentDescriptionShortCredit
FPTR_PDL_STAFF	FiscalPrinter	Int32	PaymentDescriptionStaff
FPTR_PDL_VOUCHER	FiscalPrinter	Int32	PaymentDescriptionVoucher
FPTR_LC_ITEM	FiscalPrinter	Int32	LineCountItem
FPTR_LC_ITEM_VOID	FiscalPrinter	Int32	LineCountItemVoid
FPTR_LC_DISCOUNT	FiscalPrinter	Int32	LineCountDiscount
FPTR_LC_DISCOUNT_VOID	FiscalPrinter	Int32	LineCountDiscountVoid
FPTR_LC_SURCHARGE	FiscalPrinter	Int32	LineCountSurcharge
FPTR_LC_SURCHARGE_VOID	FiscalPrinter	Int32	LineCountSurchargeVoid
FPTR_LC_REFUND	FiscalPrinter	Int32	LineCountRefund
FPTR_LC_REFUND_VOID	FiscalPrinter	Int32	LineCountRefundVoid
FPTR_LC_SUBTOTAL_DISCOUNT	FiscalPrinter	Int32	LineCountSubtotalDiscount
FPTR_LC_SUBTOTAL_DISCOUNT_VOID	FiscalPrinter	Int32	LineCountSubtotalDiscountVoid
FPTR_LC_SUBTOTAL_SURCHARGE	FiscalPrinter	Int32	LineCountSubtotalSurcharge
FPTR_LC_SUBTOTAL_SURCHARGE_VOID	FiscalPrinter	Int32	LineCountSubtotalSurchargeVoid
FPTR_LC_COMMENT	FiscalPrinter	Int32	LineCountComment
FPTR_LC_SUBTOTAL	FiscalPrinter	Int32	LineCountSubtotal
FPTR_LC_TOTAL	FiscalPrinter	Int32	LineCountTotal
FPTR_DL_ITEM	FiscalPrinter	Int32	DescriptionLengthItem
FPTR_DL_ITEM_ADJUSTMENT	FiscalPrinter	Int32	DescriptionLengthItemAdjustment
FPTR_DL_ITEM_FUEL	FiscalPrinter	Int32	DescriptionLengthItemFuel
FPTR_DL_ITEM_FUEL_VOID	FiscalPrinter	Int32	DescriptionLengthItemFuelVoid
FPTR_DL_NOT_PAID	FiscalPrinter	Int32	DescriptionLengthNotPaid
FPTR_DL_PACKAGE_ADJUSTMENT	FiscalPrinter	Int32	DescriptionLengthPackageAdjustment
FPTR_DL_REFUND	FiscalPrinter	Int32	DescriptionLengthRefund
FPTR_DL_REFUND_VOID	FiscalPrinter	Int32	DescriptionLengthRefundVoid
FPTR_DL_SUBTOTAL_ADJUSTMENT	FiscalPrinter	Int32	DescriptionLengthSubtotalAdjustment
FPTR_DL_TOTAL	FiscalPrinter	Int32	DescriptionLengthTotal
FPTR_DL_VOID	FiscalPrinter	Int32	DescriptionLengthVoid
FPTR_DL_VOID_ITEM	FiscalPrinter	Int32	DescriptionLengthVoidItem
FPTR_GT_GROSS	FiscalTotalizer	enum	Gross
FPTR_GT_NET	FiscalTotalizer	enum	Net
FPTR_GT_DISCOUNT	FiscalTotalizer	enum	Discount
FPTR_GT_DISCOUNT_VOID	FiscalTotalizer	enum	DiscountVoid
FPTR_GT_ITEM	FiscalTotalizer	enum	Item
FPTR_GT_ITEM_VOID	FiscalTotalizer	enum	ItemVoid
FPTR_GT_NOT_PAID	FiscalTotalizer	enum	NotPaid
FPTR_GT_REFUND	FiscalTotalizer	enum	Refund
FPTR_GT_REFUND_VOID	FiscalTotalizer	enum	RefundVoid
FPTR_GT_SUBTOTAL_DISCOUNT	FiscalTotalizer	enum	SubtotalDiscount
FPTR_GT_SUBTOTAL_DISCOUNT_VOID	FiscalTotalizer	enum	SubtotalDiscountVoid
FPTR_GT_SUBTOTAL_SURCHARGES	FiscalTotalizer	enum	SubtotalSurcharges
FPTR_GT_SUBTOTAL_SURCHARGES_VOID	FiscalTotalizer	enum	SubtotalSurchargesVoid
FPTR_GT_SURCHARGE	FiscalTotalizer	enum	Surcharge
FPTR_GT_SURCHARGE_VOID	FiscalTotalizer	enum	SurchargeVoid
FPTR_GT_VAT	FiscalTotalizer	enum	Vat
FPTR_GT_VAT_CATEGORY	FiscalTotalizer	enum	VatCategory
FPTR_AT_AMOUNT_DISCOUNT	FiscalAdjustment	enum	AmountDiscount
FPTR_AT_AMOUNT_SURCHARGE	FiscalAdjustment	enum	AmountSurcharge
FPTR_AT_PERCENTAGE_DISCOUNT	FiscalAdjustment	enum	PercentageDiscount
FPTR_AT_PERCENTAGE_SURCHARGE	FiscalAdjustment	enum	PercentageSurcharge
FPTR_RT_ORDINAL	FiscalReport	enum	Ordinal
FPTR_RT_DATE	FiscalReport	enum	Date
FPTR_SC_EURO	FiscalCurrency	enum	Euro
FPTR_SUE_COVER_OPEN	PrinterStatus	enum	CoverOpen
FPTR_SUE_COVER_OK	PrinterStatus	enum	CoverOK
FPTR_SUE_JRN_COVER_OPEN	PrinterStatus	enum	JournalCoverOpen
FPTR_SUE_JRN_COVER_OK	PrinterStatus	enum	JournalCoverOK

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
FPTR_SUE_REC_COVER_OPEN	PrinterStatus	enum	ReceiptCoverOpen
FPTR_SUE_REC_COVER_OK	PrinterStatus	enum	ReceiptCoverOK
FPTR_SUE_SLP_COVER_OPEN	PrinterStatus	enum	SlipCoverOpen
FPTR_SUE_SLP_COVER_OK	PrinterStatus	enum	SlipCoverOK
FPTR_SUE_JRN_EMPTY	PrinterStatus	enum	JournalEmpty
FPTR_SUE_JRN_NEAREMPTY	PrinterStatus	enum	JournalNearEmpty
FPTR_SUE_JRN_PAPEROK	PrinterStatus	enum	JournalPaperOK
FPTR_SUE_REC_EMPTY	PrinterStatus	enum	ReceiptEmpty
FPTR_SUE_REC_NEAREMPTY	PrinterStatus	enum	ReceiptNearEmpty
FPTR_SUE_REC_PAPEROK	PrinterStatus	enum	ReceiptPaperOK
FPTR_SUE_SLP_EMPTY	PrinterStatus	enum	SlipEmpty
FPTR_SUE_SLP_NEAREMPTY	PrinterStatus	enum	SlipNearEmpty
FPTR_SUE_SLP_PAPEROK	PrinterStatus	enum	SlipPaperOK
FPTR_SUE_IDLE	PrinterStatus	enum	Idle
OPOS_EFPTR_COVER_OPEN	FiscalPrinter	Int32	ExtendedErrorCoverOpen
OPOS_EFPTR_JRN_EMPTY	FiscalPrinter	Int32	ExtendedErrorJournalEmpty
OPOS_EFPTR_REC_EMPTY	FiscalPrinter	Int32	ExtendedErrorReceiptEmpty
OPOS_EFPTR_SLP_EMPTY	FiscalPrinter	Int32	ExtendedErrorSlipEmpty
OPOS_EFPTR_SLP_FORM	FiscalPrinter	Int32	ExtendedErrorSlipForm
OPOS_EFPTR_MISSING_DEVICES	FiscalPrinter	Int32	ExtendedErrorMissingDevices
OPOS_EFPTR_WRONG_STATE	FiscalPrinter	Int32	ExtendedErrorWrongState
OPOS_EFPTR_TECHNICAL_ASSISTANCE	FiscalPrinter	Int32	ExtendedErrorTechnicalAssistance
OPOS_EFPTR_CLOCK_ERROR	FiscalPrinter	Int32	ExtendedErrorClockError
OPOS_EFPTR_FISCAL_MEMORY_DISCONNECTED	FiscalPrinter	Int32	ExtendedErrorMemoryDisconnected
OPOS_EFPTR_FISCAL_MEMORY_FULL	FiscalPrinter	Int32	ExtendedErrorMemoryFull
OPOS_EFPTR_FISCAL_TOTALS_ERROR	FiscalPrinter	Int32	ExtendedErrorTotalsError
OPOS_EFPTR_BAD_ITEM_QUANTITY	FiscalPrinter	Int32	ExtendedErrorBadItemQuantity
OPOS_EFPTR_BAD_ITEM_AMOUNT	FiscalPrinter	Int32	ExtendedErrorBadItemAmount
OPOS_EFPTR_BAD_ITEM_DESCRIPTION	FiscalPrinter	Int32	ExtendedErrorBadItemDescription
OPOS_EFPTR_RECEIPT_TOTAL_OVERFLOW	FiscalPrinter	Int32	ExtendedErrorReceiptTotalOverflow
OPOS_EFPTR_BAD_VAT	FiscalPrinter	Int32	ExtendedErrorBadVat
OPOS_EFPTR_BAD_PRICE	FiscalPrinter	Int32	ExtendedErrorBadPrice
OPOS_EFPTR_BAD_DATE	FiscalPrinter	Int32	ExtendedErrorBadDate
OPOS_EFPTR_NEGATIVE_TOTAL	FiscalPrinter	Int32	ExtendedErrorNegativeTotal
OPOS_EFPTR_WORD_NOT_ALLOWED	FiscalPrinter	Int32	ExtendedErrorWordNotAllowed
OPOS_EFPTR_BAD_LENGTH	FiscalPrinter	Int32	ExtendedErrorBadLength
OPOS_EFPTR_MISSING_SET_CURRENCY	FiscalPrinter	Int32	ExtendedErrorMissingSetCurrency
KBD_ET_DOWN	KeyboardEventType	enum	Down
KBD_ET_DOWN_UP	KeyboardEventType	enum	DownUp
KBD_KET_KEYDOWN	KeyEvent	enum	Down
KBD_KET_KEYUP	KeyEvent	enum	Up
LOCK_KP_ANY	Keylock	Int32	PositionAny
LOCK_KP_LOCK	Keylock	Int32	PositionLocked
LOCK_KP_NORM	Keylock	Int32	PositionNormal
LOCK_KP_SUPR	Keylock	Int32	PositionSupervisor
MICR_CT_PERSONAL	CheckType	enum	Personal
MICR_CT_BUSINESS	CheckType	enum	Business
MICR_CT_UNKNOWN	CheckType	enum	Unknown
MICR_CC_USA	CheckCountryCode	enum	Usa
MICR_CC_CANADA	CheckCountryCode	enum	Canada
MICR_CC_MEXICO	CheckCountryCode	enum	Mexico
MICR_CC_UNKNOWN	CheckCountryCode	enum	Unknown Check Font E-13B
MICR_CC_CMC7	CheckCountryCode	enum	Unknown Check Font CMC-7
MICR_CC_OTHER	CheckCountryCode	enum	Unknown Check Font OCR-A or OCR_B
OPOS_EMICR_NOCHECK	Micr	Int32	ExtendedErrorNoCheck
OPOS_EMICR_CHECK	Micr	Int32	ExtendedErrorCheck
OPOS_EMICR_BADDATA	Micr	Int32	ExtendedErrorBadData
OPOS_EMICR_NODATA	Micr	Int32	ExtendedErrorNoData

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
OPOS_EMICR_BADSIZE	Micr	Int32	ExtendedErrorBadSize
OPOS_EMICR_JAM	Micr	Int32	ExtendedErrorJam
OPOS_EMICR_CHECKDIGIT	Micr	Int32	ExtendedErrorCheckDigit
OPOS_EMICR_COVEROPEN	Micr	Int32	ExtendedErrorCoverOpen
MOTION_M_PRESENT	MotionSensor	Int32	StatusMotionPresent
MOTION_M_ABSENT	MotionSensor	Int32	StatusMotionAbsent
MSR_TR_1	MsrTracks	enum	Track1
MSR_TR_2	MsrTracks	enum	Track2
MSR_TR_3	MsrTracks	enum	Track3
MSR_TR_4	MsrTracks	enum	Track4
MSR_TR_1_2	MsrTracks	enum	Tracks12
MSR_TR_1_3	MsrTracks	enum	Tracks13
MSR_TR_1_4	MsrTracks	enum	Tracks14
MSR_TR_2_3	MsrTracks	enum	Tracks23
MSR_TR_2_4	MsrTracks	enum	Tracks24
MSR_TR_3_4	MsrTracks	enum	Tracks34
MSR_TR_1_2_3	MsrTracks	enum	Tracks123
MSR_TR_1_2_4	MsrTracks	enum	Tracks124
MSR_TR_1_3_4	MsrTracks	enum	Tracks134
MSR_TR_2_3_4	MsrTracks	enum	Tracks234
MSR_TR_1_2_3_4	MsrTracks	enum	Tracks1234
MSR_ERT_CARD	MsrErrorReporting	enum	Card
MSR_ERT_TRACK	MsrErrorReporting	enum	Track
	Msr	Int32	ExtendedErrorSuccess
	Msr	Int32	ExtendedErrorFailure
OPOS_EMSR_START	Msr	Int32	ExtendedErrorStart
OPOS_EMSR_END	Msr	Int32	ExtendedErrorEnd
OPOS_EMSR_PARITY	Msr	Int32	ExtendedErrorParity
OPOS_EMSR_LRC	Msr	Int32	ExtendedErrorLrc
	CharacterSetCapability	enum	Numeric
PCRW_CCS_ALPHA	CharacterSetCapability	enum	Alpha
PCRW_CCS_ASCII	CharacterSetCapability	enum	Ascii
PCRW_CCS_KANA	CharacterSetCapability	enum	Kana
PCRW_CCS_KANJI	CharacterSetCapability	enum	Kanji
PCRW_CCS_UNICODE	CharacterSetCapability	enum	Unicode
PCRW_STATE_NOCARD	PointCardState	enum	NoCard
PCRW_STATE_REMAINING	PointCardState	enum	Remaining
PCRW_STATE_INRW	PointCardState	enum	Inserted
PCRW_TRACK1	PointCardRWTracks	enum	Track1
PCRW_TRACK2	PointCardRWTracks	enum	Track2
PCRW_TRACK3	PointCardRWTracks	enum	Track3
PCRW_TRACK4	PointCardRWTracks	enum	Track4
PCRW_TRACK5	PointCardRWTracks	enum	Track5
PCRW_TRACK6	PointCardRWTracks	enum	Track6
PCRW_CS_UNICODE	PosCommon	Int32	CharacterSetUnicode
PCRW_CS_ASCII	PosCommon	Int32	CharacterSetAscii
PCRW_CS_WINDOWS	PosCommon	Int32	
PCRW_CS_ANSI	PosCommon	Int32	CharacterSetAnsi
PCRW_MM_DOTS	MapMode	enum	Dots
PCRW_MM_TWIPS	MapMode	enum	Twips
PCRW_MM_ENGLISH	MapMode	enum	English
PCRW_MM_METRIC	MapMode	enum	Metric
OPOS_EPCRW_READ	PointCardRW	Int32	ExtendedErrorRead
OPOS_EPCRW_WRITE	PointCardRW	Int32	ExtendedErrorWrite
OPOS_EPCRW_JAM	PointCardRW	Int32	ExtendedErrorJam
OPOS_EPCRW_MOTOR	PointCardRW	Int32	ExtendedErrorMotor
OPOS_EPCRW_COVER	PointCardRW	Int32	ExtendedErrorCover
OPOS_EPCRW_PRINTER	PointCardRW	Int32	ExtendedErrorPrinter
OPOS_EPCRW_RELEASE	PointCardRW	Int32	ExtendedErrorRelease

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
OPOS_EPCRW_DISPLAY	PointCardRW	Int32	ExtendedErrorDisplay
OPOS_EPCRW_NOCARD	PointCardRW	Int32	ExtendedErrorNoCard
	PointCardReadWriteState	enum	Success
OPOS_EPCRW_START	PointCardReadWriteState	enum	Start
OPOS_EPCRW_END	PointCardReadWriteState	enum	End
OPOS_EPCRW_PARITY	PointCardReadWriteState	enum	Parity
OPOS_EPCRW_ENCODE	PointCardReadWriteState	enum	Encode
OPOS_EPCRW_LRC	PointCardReadWriteState	enum	LrcError
OPOS_EPCRW_VERIFY	PointCardReadWriteState	enum	Verify
	PointCardReadWriteState	enum	Failure
PCRW_RP_NORMAL	PrintRotation	enum	Normal
PCRW_RP_RIGHT90	PrintRotation	enum	Right90
PCRW_RP_LEFT90	PrintRotation	enum	Left90
PCRW_RP_ROTATE180	PrintRotation	enum	Rotate180
PCRW_SUE_NOCARD	PointCardRW	Int32	StatusNoCard
PCRW_SUE_REMAINING	PointCardRW	Int32	StatusRemaining
PCRW_SUE_INRW	PointCardRW	Int32	StatusInserted
	PointCardKinds	enum	PrintingArea
	PointCardKinds	enum	MagneticTracks
	PointCardKinds	enum	PrintingAreaAndMagneticTracks
PPAD_DISP_UNRESTRICTED	PinPadDisplay	enum	Unrestricted
PPAD_DISP_PINRESTRICTED	PinPadDisplay	enum	PinRestricted
PPAD_DISP_RESTRICTED_LIST	PinPadDisplay	enum	RestrictedList
PPAD_DISP_RESTRICTED_ORDER	PinPadDisplay	enum	RestrictedOrder
PPAD_DISP_NONE	PinPadDisplay	enum	None
PPAD_MSG_ENTERPIN	PinPadMessage	enum	EnterPin
PPAD_MSG_PLEASEWAIT	PinPadMessage	enum	PleaseWait
PPAD_MSG_ENTERVALIDPIN	PinPadMessage	enum	EnterValidPin
PPAD_MSG_RETRIESEXCEEDED	PinPadMessage	enum	RetriesExceeded
PPAD_MSG_APPROVED	PinPadMessage	enum	Approved
PPAD_MSG_DECLINED	PinPadMessage	enum	Declined
PPAD_MSG_CANCELED	PinPadMessage	enum	Canceled
PPAD_MSG_AMOUNTOK	PinPadMessage	enum	AmountOK
PPAD_MSG_NOTREADY	PinPadMessage	enum	NotReady
PPAD_MSG_IDLE	PinPadMessage	enum	Idle
PPAD_MSG_SLIDE_CARD	PinPadMessage	enum	SlideCard
PPAD_MSG_INSERTCARD	PinPadMessage	enum	InsertCard
PPAD_MSG_SELECTCARDTYPE	PinPadMessage	enum	SelectCardType
PPAD_LANG_NONE	PinPadLanguage	enum	None
PPAD_LANG_ONE	PinPadLanguage	enum	One
PPAD_LANG_PINRESTRICTED	PinPadLanguage	enum	PinRestricted
PPAD_LANG_UNRESTRICTED	PinPadLanguage	enum	Unrestricted
PPAD_TRANS_DEBIT	EftTransactionType	enum	Debit
PPAD_TRANS_CREDIT	EftTransactionType	enum	Credit
PPAD_TRANS_INQ	EftTransactionType	enum	Inquiry
PPAD_TRANS_RECONCILE	EftTransactionType	enum	Reconcile
PPAD_TRANS_ADMIN	EftTransactionType	enum	Admin
PPAD_EFT_NORMAL	EftTransactionControl	enum	Normal
PPAD_EFT_ABNORMAL	EftTransactionControl	enum	Abnormal
PPAD_SUCCESS	PinEntryStatus	enum	Success
PPAD_CANCEL	PinEntryStatus	enum	Cancel
	PinEntryStatus	enum	Timeout
	PinEntryStatus	enum	BadKey
	PinPadSystem	enum	MasterSession
	PinPadSystem	enum	Dukpt
	PinPadSystem	enum	Apacs40
	PinPadSystem	enum	AS2805
	PinPadSystem	enum	Hgepos
	PinPadSystem	enum	Jdebit2

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
OPOS_EPPAD_BAD_KEY	PinPad	Int32	ExtendedErrorBadKey
	PrinterStation	enum	None
PTR_S_JOURNAL	PrinterStation	enum	Journal
PTR_S_RECEIPT	PrinterStation	enum	Receipt
PTR_S_SLIP	PrinterStation	enum	Slip
PTR_S_JOURNAL_RECEIPT	FiscalPrinterStations	enum	JournalReceipt
PTR_S_JOURNAL_SLIP	FiscalPrinterStations	enum	JournalSlip
PTR_S_RECEIPT_SLIP	FiscalPrinterStations	enum	ReceiptSlip
PTR_TWO_RECEIPT_JOURNAL	PrinterStation	enum	TwoReceiptJournal
PTR_TWO_SLIP_JOURNAL	PrinterStation	enum	TwoSlipJournal
PTR_TWO_SLIP_RECEIPT	PrinterStation	enum	TwoSlipReceipt
	CharacterSetCapability	enum	Numeric
PTR_CCS_ALPHA	CharacterSetCapability	enum	Alpha
PTR_CCS_ASCII	CharacterSetCapability	enum	Ascii
PTR_CCS_KANA	CharacterSetCapability	enum	Kana
PTR_CCS_KANJI	CharacterSetCapability	enum	Kanji
PTR_CCS_UNICODE	CharacterSetCapability	enum	Unicode
PTR_CS_UNICODE	PosCommon	Int32	CharacterSetUnicode
PTR_CS_ASCII	PosCommon	Int32	CharacterSetAscii
PTR_CS_WINDOWS	PosCommon	Int32	
PTR_CS_ANSI	PosCommon	Int32	CharacterSetAnsi
PTR_EL_NONE	PrinterErrorLevel	enum	None
PTR_EL_RECOVERABLE	PrinterErrorLevel	enum	Recoverable
PTR_EL_FATAL	PrinterErrorLevel	enum	Fatal
PTR_MM_DOTS	MapMode	enum	Dots
PTR_MM_TWIPS	MapMode	enum	Twips
PTR_MM_ENGLISH	MapMode	enum	English
PTR_MM_METRIC	MapMode	enum	Metric
	PrinterColors	enum	None
PTR_COLOR_PRIMARY	PrinterColors	enum	Primary
PTR_COLOR_CUSTOM1	PrinterColors	enum	Custom1
PTR_COLOR_CUSTOM2	PrinterColors	enum	Custom2
PTR_COLOR_CUSTOM3	PrinterColors	enum	Custom3
PTR_COLOR_CUSTOM4	PrinterColors	enum	Custom4
PTR_COLOR_CUSTOM5	PrinterColors	enum	Custom5
PTR_COLOR_CUSTOM6	PrinterColors	enum	Custom6
PTR_COLOR_CYAN	PrinterColors	enum	Cyan
PTR_COLOR_MAGENTA	PrinterColors	enum	Magenta
PTR_COLOR_YELLOW	PrinterColors	enum	Yellow
PTR_COLOR_FULL	PrinterColors	enum	Full
PTR_CART_UNKNOWN	PrinterCartridgeStates	enum	Unknown
PTR_CART_OK	PrinterCartridgeStates	enum	OK
PTR_CART_REMOVED	PrinterCartridgeStates	enum	Removed
PTR_CART_EMPTY	PrinterCartridgeStates	enum	Empty
PTR_CART_NEAREND	PrinterCartridgeStates	enum	NearEnd
PTR_CART_CLEANING	PrinterCartridgeStates	enum	Cleaning
PTR_CN_DISABLED	PrinterCartridgeNotify	enum	Disabled
PTR_CN_ENABLED	PrinterCartridgeNotify	enum	Enabled
PTR_CP_FULLCUT	PosPrinter	Int32	PrinterCutPaperFullCut
PTR_BC_LEFT	PosPrinter	Int32	PrinterBarCodeLeft
PTR_BC_CENTER	PosPrinter	Int32	PrinterBarCodeCenter
PTR_BC_RIGHT	PosPrinter	Int32	PrinterBarCodeRight
PTR_BC_TEXT_NONE	BarCodeTextPosition	enum	None
PTR_BC_TEXT_ABOVE	BarCodeTextPosition	enum	Above
PTR_BC_TEXT_BELOW	BarCodeTextPosition	enum	Below
	BarCodeSymbology	enum	Unknown

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
PTR_BCS_UPCA	BarCodeSymbology	enum	Upca
PTR_BCS_UPCE	BarCodeSymbology	enum	Upce
PTR_BCS_JAN8	BarCodeSymbology	enum	EanJan8
PTR_BCS_EAN8	BarCodeSymbology	enum	
PTR_BCS_JAN13	BarCodeSymbology	enum	EanJan13
PTR_BCS_EAN13	BarCodeSymbology	enum	
PTR_BCS_TF	BarCodeSymbology	enum	TF
PTR_BCS_ITF	BarCodeSymbology	enum	Itf
PTR_BCS_Codabar	BarCodeSymbology	enum	Codabar
PTR_BCS_Code39	BarCodeSymbology	enum	Code39
PTR_BCS_Code93	BarCodeSymbology	enum	Code93
PTR_BCS_Code128	BarCodeSymbology	enum	Code128
PTR_BCS_UPCA_S	BarCodeSymbology	enum	Upcas
PTR_BCS_UPCE_S	BarCodeSymbology	enum	Upces
PTR_BCS_UPCD1	BarCodeSymbology	enum	Upcd1
PTR_BCS_UPCD2	BarCodeSymbology	enum	Upcd2
PTR_BCS_UPCD3	BarCodeSymbology	enum	Upcd3
PTR_BCS_UPCD4	BarCodeSymbology	enum	Upcd4
PTR_BCS_UPCD5	BarCodeSymbology	enum	Upcd5
PTR_BCS_EAN8_S	BarCodeSymbology	enum	Ean8S
PTR_BCS_EAN13_S	BarCodeSymbology	enum	Ean13S
PTR_BCS_EAN128	BarCodeSymbology	enum	Ean128
PTR_BCS_OCRA	BarCodeSymbology	enum	Ocra
PTR_BCS_OCRB	BarCodeSymbology	enum	Ocrb
PTR_BCS_Code128_Parsed	BarCodeSymbology	enum	Code128 Parsed
PTR_BCS_RSS14	BarCodeSymbology	enum	Rss14 – Deprecated v1.12
PTR_BCS_RSS_EXPANDED	BarCodeSymbology	enum	RssExpanded – Deprecated v1.12
PTR_BCS_GS1DATABAR	BarCodeSymbology	enum	GS1 DataBar Omnidirectional
PTR_BCS_GS1DATABAR_S	BarCodeSymbology	enum	GS1 DataBar Stacked Omnidirectional
PTR_BCS_GS1DATABAR_E	BarCodeSymbology	enum	GS1 DataBar Expanded
PTR_BCS_GS1DATABAR_E_S	BarCodeSymbology	enum	GS1 DataBar Expanded Stacked
	BarCodeSymbology	enum	Cca
	BarCodeSymbology	enum	Ccb
	BarCodeSymbology	enum	Ccc
PTR_BCS_PDF417	BarCodeSymbology	enum	Pdf417
PTR_BCS_MAXICODE	BarCodeSymbology	enum	Maxicode
PTR_BCS_DATAMATRIX	BarCodeSymbology	enum	Data Matrix
PTR_BCS_QRCODE	BarCodeSymbology	enum	QR Code
PTR_BCS_UQRCODE	BarCodeSymbology	enum	Micro QR Code
PTR_BCS_AZTEC	BarCodeSymbology	enum	Aztec
PTR_BCS_UPDF417	BarCodeSymbology	enum	Micro Pdf417
PTR_BCS_OTHER	BarCodeSymbology	enum	Other
PTR_BM_ASIS	PosPrinter	Int32	PrinterBitmapAsIs
PTR_BM_LEFT	PosPrinter	Int32	PrinterBitmapLeft
PTR_BM_CENTER	PosPrinter	Int32	PrinterBitmapCenter
PTR_BM_RIGHT	PosPrinter	Int32	PrinterBitmapRight
PTR_RP_NORMAL	PrintRotation	enum	Normal
PTR_RP_RIGHT90	PrintRotation	enum	Right90
PTR_RP_LEFT90	PrintRotation	enum	Left90
PTR_RP_ROTATE180	PrintRotation	enum	Rotate180
PTR_RP_BARCODE	PrintRotation	enum	Barcode
PTR_RP_BITMAP	PrintRotation	enum	Bitmap
PTR_L_TOP	PrinterLogoLocation	enum	Top
PTR_L_BOTTOM	PrinterLogoLocation	enum	Bottom
PTR_TP_TRANSACTION	PrinterTransactionControl	enum	Transaction
PTR_TP_NORMAL	PrinterTransactionControl	enum	Normal
	PrinterMarkFeeds	enum	None
PTR_MF_TO_TAKEUP	PrinterMarkFeeds	enum	Takeup
PTR_MF_TO_CUTTER	PrinterMarkFeeds	enum	Cutter
PTR_MF_TO_CURRENT_TOF	PrinterMarkFeeds	enum	CurrentTopOfForm
PTR_MF_TO_NEXT_TOF	PrinterMarkFeeds	enum	NextTopOfForm
PTR_PS_UNKNOWN	PrinterSide	enum	Unknown
PTR_PS_SIDE1	PrinterSide	enum	Side1

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
PTR_PS_SIDE2	PrinterSide	enum	Side2
PTR_PS_OPPOSITE	PrinterSide	enum	Opposite
PTR_SUE_COVER_OPEN	PrinterStatus	enum	CoverOpen
PTR_SUE_COVER_OK	PrinterStatus	enum	CoverOK
PTR_SUE_JRN_EMPTY	PrinterStatus	enum	JournalEmpty
PTR_SUE_JRN_NEAREMPTY	PrinterStatus	enum	JournalNearEmpty
PTR_SUE_JRN_PAPEROK	PrinterStatus	enum	JournalPaperOK
PTR_SUE_REC_EMPTY	PrinterStatus	enum	ReceiptEmpty
PTR_SUE_REC_NEAREMPTY	PrinterStatus	enum	ReceiptNearEmpty
PTR_SUE_REC_PAPEROK	PrinterStatus	enum	ReceiptPaperOK
PTR_SUE_SLP_EMPTY	PrinterStatus	enum	SlipEmpty
PTR_SUE_SLP_NEAREMPTY	PrinterStatus	enum	SlipNearEmpty
PTR_SUE_SLP_PAPEROK	PrinterStatus	enum	SlipPaperOK
PTR_SUE_JRN_CARTRIDGE_EMPTY	PrinterStatus	enum	JournalCartridgeEmpty
PTR_SUE_JRN_CARTRIDGE_NEAREMPTY	PrinterStatus	enum	JournalCartridgeNearEmpty
PTR_SUE_JRN_HEAD_CLEANING	PrinterStatus	enum	JournalHeadCleaning
PTR_SUE_JRN_CARTRIDGE_OK	PrinterStatus	enum	JournalCartridgeOK
PTR_SUE_REC_CARTRIDGE_EMPTY	PrinterStatus	enum	ReceiptCartridgeEmpty
PTR_SUE_REC_CARTRIDGE_NEAREMPTY	PrinterStatus	enum	ReceiptCartridgeNearEmpty
PTR_SUE_REC_HEAD_CLEANING	PrinterStatus	enum	ReceiptHeadCleaning
PTR_SUE_REC_CARTRIDGE_OK	PrinterStatus	enum	ReceiptCartridgeOK
PTR_SUE_SLP_CARTRIDGE_EMPTY	PrinterStatus	enum	SlipCartridgeEmpty
PTR_SUE_SLP_CARTRIDGE_NEAREMPTY	PrinterStatus	enum	SlipCartridgeNearEmpty
PTR_SUE_SLP_HEAD_CLEANING	PrinterStatus	enum	SlipHeadCleaning
PTR_SUE_SLP_CARTRIDGE_OK	PrinterStatus	enum	SlipCartridgeOK
PTR_SUE_JRN_COVER_OPEN	PrinterStatus	enum	JournalCoverOpen
PTR_SUE_JRN_COVER_OK	PrinterStatus	enum	JournalCoverOK
PTR_SUE_REC_COVER_OPEN	PrinterStatus	enum	ReceiptCoverOpen
PTR_SUE_REC_COVER_OK	PrinterStatus	enum	ReceiptCoverOK
PTR_SUE_SLP_COVER_OPEN	PrinterStatus	enum	SlipCoverOpen
PTR_SUE_SLP_COVER_OK	PrinterStatus	enum	SlipCoverOK
PTR_SUE_IDLE	PrinterStatus	enum	Idle
OPOS_EPTR_COVER_OPEN	PosPrinter	Int32	ExtendedErrorCoverOpen
OPOS_EPTR_JRN_EMPTY	PosPrinter	Int32	ExtendedErrorJrnEmpty
OPOS_EPTR_REC_EMPTY	PosPrinter	Int32	ExtendedErrorRecEmpty
OPOS_EPTR_SLP_EMPTY	PosPrinter	Int32	ExtendedErrorSlpEmpty
OPOS_EPTR_SLP_FORM	PosPrinter	Int32	ExtendedErrorSlpForm
OPOS_EPTR_TOOBIG	PosPrinter	Int32	ExtendedErrorTooBig
OPOS_EPTR_BADFORMAT	PosPrinter	Int32	ExtendedErrorBadFormat
OPOS_EPTR_JRN_CARTRIDGE_REMOVED	PosPrinter	Int32	ExtendedErrorJrnCartridgeRemoved
OPOS_EPTR_JRN_CARTRIDGE_EMPTY	PosPrinter	Int32	ExtendedErrorJrnCartridgeEmpty
OPOS_EPTR_JRN_HEAD_CLEANING	PosPrinter	Int32	ExtendedErrorJrnHeadCleaning
OPOS_EPTR_REC_CARTRIDGE_REMOVED	PosPrinter	Int32	ExtendedErrorRecCartridgeRemoved
OPOS_EPTR_REC_CARTRIDGE_EMPTY	PosPrinter	Int32	ExtendedErrorRecCartridgeEmpty
OPOS_EPTR_REC_HEAD_CLEANING	PosPrinter	Int32	ExtendedErrorRecHeadCleaning
OPOS_EPTR_SLP_CARTRIDGE_REMOVED	PosPrinter	Int32	ExtendedErrorSlpCartridgeRemoved
OPOS_EPTR_SLP_CARTRIDGE_EMPTY	PosPrinter	Int32	ExtendedErrorSlpCartridgeEmpty
OPOS_EPTR_SLP_HEAD_CLEANING	PosPrinter	Int32	ExtendedErrorSlpHeadCleaning
PWR_UPS_FULL	UpsChargeStates	enum	Full
PWR_UPS_WARNING	UpsChargeStates	enum	Warning
PWR_UPS_LOW	UpsChargeStates	enum	Low
PWR_UPS_CRITICAL	UpsChargeStates	enum	Critical
PWR_SUE_UPS_FULL	PosPower	Int32	StatusUpsFull
PWR_SUE_UPS_WARNING	PosPower	Int32	StatusUpsWarning
PWR_SUE_UPS_LOW	PosPower	Int32	StatusUpsLow
PWR_SUE_UPS_CRITICAL	PosPower	Int32	StatusUpsCritical
PWR_SUE_FAN_STOPPED	PosPower	Int32	StatusFanStopped
PWR_SUE_FAN_RUNNING	PosPower	Int32	StatusFanRunning
PWR_SUE_TEMPERATURE_HIGH	PosPower	Int32	StatusTemperatureHigh
PWR_SUE_TEMPERATURE_OK	PosPower	Int32	StatusTemperatureOK
PWR_SUE_SHUTDOWN	PosPower	Int32	StatusShutDown
ROD_UID_1	DeviceUnits	enum	nit1

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
ROD_UID_2	DeviceUnits	enum	Unit2
ROD_UID_3	DeviceUnits	enum	Unit3
ROD_UID_4	DeviceUnits	enum	Unit4
ROD_UID_5	DeviceUnits	enum	Unit5
ROD_UID_6	DeviceUnits	enum	Unit6
ROD_UID_7	DeviceUnits	enum	Unit7
ROD_UID_8	DeviceUnits	enum	Unit8
ROD_UID_9	DeviceUnits	enum	Unit9
ROD_UID_10	DeviceUnits	enum	Unit10
ROD_UID_11	DeviceUnits	enum	Unit11
ROD_UID_12	DeviceUnits	enum	Unit12
ROD_UID_13	DeviceUnits	enum	Unit13
ROD_UID_14	DeviceUnits	enum	Unit14
ROD_UID_15	DeviceUnits	enum	Unit15
ROD_UID_16	DeviceUnits	enum	Unit16
ROD_UID_17	DeviceUnits	enum	Unit17
ROD_UID_18	DeviceUnits	enum	Unit18
ROD_UID_19	DeviceUnits	enum	Unit19
ROD_UID_20	DeviceUnits	enum	Unit20
ROD_UID_21	DeviceUnits	enum	Unit21
ROD_UID_22	DeviceUnits	enum	Unit22
ROD_UID_23	DeviceUnits	enum	Unit23
ROD_UID_24	DeviceUnits	enum	Unit24
ROD_UID_25	DeviceUnits	enum	Unit25
ROD_UID_26	DeviceUnits	enum	Unit26
ROD_UID_27	DeviceUnits	enum	Unit27
ROD_UID_28	DeviceUnits	enum	Unit28
ROD_UID_29	DeviceUnits	enum	Unit29
ROD_UID_30	DeviceUnits	enum	Unit30
ROD_UID_31	DeviceUnits	enum	Unit31
ROD_UID_32	DeviceUnits	enum	Unit32
ROD_ATTR_BLINK	VideoAttributes	enum	Blink
ROD_ATTR_BG_BLACK	VideoAttributes	enum	BackgroundBlack
ROD_ATTR_BG_BLUE	VideoAttributes	enum	BackgroundBlue
ROD_ATTR_BG_GREEN	VideoAttributes	enum	BackgroundGreen
ROD_ATTR_BG_CYAN	VideoAttributes	enum	BackgroundCyan
ROD_ATTR_BG_RED	VideoAttributes	enum	BackgroundRed
ROD_ATTR_BG_MAGENTA	VideoAttributes	enum	BackgroundMagenta
ROD_ATTR_BG_BROWN	VideoAttributes	enum	BackgroundBrown
ROD_ATTR_BG_GRAY	VideoAttributes	enum	BackgroundGray
ROD_ATTR_INTENSITY	VideoAttributes	enum	ntensity
ROD_ATTR_FG_BLACK	VideoAttributes	enum	ForegroundBlack
ROD_ATTR_FG_BLUE	VideoAttributes	enum	ForegroundBlue
ROD_ATTR_FG_GREEN	VideoAttributes	enum	ForegroundGreen
ROD_ATTR_FG_CYAN	VideoAttributes	enum	ForegroundCyan
ROD_ATTR_FG_RED	VideoAttributes	enum	ForegroundRed
ROD_ATTR_FG_MAGENTA	VideoAttributes	enum	ForegroundMagenta
ROD_ATTR_FG_BROWN	VideoAttributes	enum	ForegroundBrown
ROD_ATTR_FG_GRAY	VideoAttributes	enum	ForegroundGray
ROD_BDR_SINGLE	BorderType	enum	Single
ROD_BDR_DOUBLE	BorderType	enum	Double
ROD_BDR_SOLID	BorderType	enum	Solid
ROD_CLK_START	ClockFunction	enum	Start
ROD_CLK_PAUSE	ClockFunction	enum	Pause
ROD_CLK_RESUME	ClockFunction	enum	Resume
ROD_CLK_MOVE	ClockFunction	enum	Move
ROD_CLK_STOP	ClockFunction	enum	Stop
ROD_CRS_LINE	VideoCursorType	enum	Line
ROD_CRS_LINE_BLINK	VideoCursorType	enum	LineBlink
ROD_CRS_BLOCK	VideoCursorType	enum	Block
ROD_CRS_BLOCK_BLINK	VideoCursorType	enum	BlockBlink
ROD_CRS_OFF	VideoCursorType	enum	Off
ROD_CS_UNICODE	PosCommon	Int32	CharacterSetUnicode
ROD_CS_ASCII	PosCommon	Int32	CharacterSetAscii

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
ROD_CS_WINDOWS	PosCommon	Int32	
ROD_CS_ANSI	PosCommon	Int32	CharacterSetAnsi
ROD_TD_TRANSACTION	RemoteOderDisplayTransaction	enum	Transaction
ROD_TD_NORMAL	RemoteOderDisplayTransaction	enum	Normal
ROD_UA_SET	VideoAttributeCommand	enum	Set
ROD_UA_INTENSITY_ON	VideoAttributeCommand	enum	IntensityOn
ROD_UA_INTENSITY_OFF	VideoAttributeCommand	enum	IntensityOff
ROD_UA_REVERSE_ON	VideoAttributeCommand	enum	ReverseOn
ROD_UA_REVERSE_OFF	VideoAttributeCommand	enum	ReverseOff
ROD_UA_BLINK_ON	VideoAttributeCommand	enum	BlinkOn
ROD_UA_BLINK_OFF	VideoAttributeCommand	enum	BlinkOff
ROD_DE_TOUCH_DOWN	RemoteOrderDisplayEventTypes	enum	TouchDown
ROD_DE_TOUCH_MOVE	RemoteOrderDisplayEventTypes	enum	TouchMove
ROD_DE_TOUCH_UP	RemoteOrderDisplayEventTypes	enum	TouchUp
OPOS_EROD_BADCLK	RemoteOrderDisplay	Int32	ExtendedErrorBadClock
OPOS_EROD_NOCLOCKS	RemoteOrderDisplay	Int32	ExtendedErrorNoClocks
OPOS_EROD_NOREGION	RemoteOrderDisplay	Int32	ExtendedErrorNoRegion
OPOS_EROD_NOROOM	RemoteOrderDisplay	Int32	ExtendedErrorNoRoom
OPOS_EROD_NOBUFFERS	RemoteOrderDisplay	Int32	ExtendedErrorNoBuffers
SCAL_WU_GRAM	WaitUnit	enum	Gram
SCAL_WU_KILOGRAM	WaitUnit	enum	Kilogram
SCAL_WU_OUNCE	WaitUnit	enum	Ounce
SCAL_WU_POUND	WaitUnit	enum	Pound
OPOS_ESCAL_OVERWEIGHT	Scale	Int32	ExtendedErrorOverWeight
SCAN_SDT_UNKNOWN	BarCodeSymbology	enum	Unknown
SCAN_SDT_UPCA	BarCodeSymbology	enum	Upca
SCAN_SDT_UPCE	BarCodeSymbology	enum	Upce
SCAN_SDT_JAN8	BarCodeSymbology	enum	EanJan8
SCAN_SDT_EAN8	BarCodeSymbology	enum	
SCAN_SDT_JAN13	BarCodeSymbology	enum	EanJan13
SCAN_SDT_EAN13	BarCodeSymbology	enum	
SCAN_SDT_TF	BarCodeSymbology	enum	TF
SCAN_SDT_ITF	BarCodeSymbology	enum	Itf
SCAN_SDT_Codabar	BarCodeSymbology	enum	Codabar
SCAN_SDT_Code39	BarCodeSymbology	enum	Code39
SCAN_SDT_Code93	BarCodeSymbology	enum	Code93
SCAN_SDT_Code128	BarCodeSymbology	enum	Code128
SCAN_SDT_UPCA_S	BarCodeSymbology	enum	Upcas
SCAN_SDT_UPCE_S	BarCodeSymbology	enum	Upces
SCAN_SDT_UPCD1	BarCodeSymbology	enum	Upcd1
SCAN_SDT_UPCD2	BarCodeSymbology	enum	Upcd2
SCAN_SDT_UPCD3	BarCodeSymbology	enum	Upcd3
SCAN_SDT_UPCD4	BarCodeSymbology	enum	Upcd4
SCAN_SDT_UPCD5	BarCodeSymbology	enum	Upcd5
SCAN_SDT_EAN8_S	BarCodeSymbology	enum	Ean8S
SCAN_SDT_EAN13_S	BarCodeSymbology	enum	Ean13S
SCAN_SDT_EAN128	BarCodeSymbology	enum	Ean128
SCAN_SDT_OCRA	BarCodeSymbology	enum	Ocra
SCAN_SDT_OCRB	BarCodeSymbology	enum	Ocrb
SCAN_SDT_RSS14	BarCodeSymbology	enum	Rss14 – Deprecated v1.12
SCAN_SDT_RSS_EXPANDED	BarCodeSymbology	enum	RssExpanded – Deprecated b1.12
SCAN_SDT_GS1DATABAR	BarCodeSymbology	enum	GS1DataBar Omnidirectional
SCAN_SDT_GS1DATABAR_E	BarCodeSymbology	enum	GS1 DataBar Expanded
SCAN_SDT_CCA	BarCodeSymbology	enum	Cca
SCAN_SDT_CCB	BarCodeSymbology	enum	Ccb
SCAN_SDT_CCC	BarCodeSymbology	enum	Ccc
SCAN_SDT_PDF417	BarCodeSymbology	enum	Pdf417
SCAN_SDT_MAXICODE	BarCodeSymbology	enum	Maxicode
SCAN_SDT_OTHER	BarCodeSymbology	enum	Other
SC_CMODE_TRANS	SmartCardInterfaceModes	enum	Transaction

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
SC_CMODE_BLOCK	SmartCardInterfaceModes	enum	Block
SC_CMODE_APDU	SmartCardInterfaceModes	enum	Apdu
SC_CMODE_XML	SmartCardInterfaceModes	enum	Xml
SC_CMODE_ISO	SmartCardIsoEmvModes	enum	Iso
SC_CMODE_EMV	SmartCardIsoEmvModes	enum	Emv
SC_CTRANS_PROTOCOL_T0	SmartCardTransactionProtocols	enum	T0
SC_CTRANS_PROTOCOL_T1	SmartCardTransactionProtocols	enum	T1
SC_MODE_TRANS	SmartCardInterfaceModes	enum	Transaction
SC_MODE_BLOCK	SmartCardInterfaceModes	enum	Block
SC_MODE_APDU	SmartCardInterfaceModes	enum	Apdu
SC_MODE_XML	SmartCardInterfaceModes	enum	Xml
SC_MODE_ISO	SmartCardIsoEmvModes	enum	Iso
SC_MODE_EMV	SmartCardIsoEmvModes	enum	Emv
SC_TRANS_PROTOCOL_T0	SmartCardTransactionProtocols	enum	T0
SC_TRANS_PROTOCOL_T1	SmartCardTransactionProtocols	enum	T1
SC_READ_DATA	SmartCardReadAction	enum	ReadData
SC_READ_PROGRAM	SmartCardReadAction	enum	ReadProgram
SC_EXECUTE_AND_READ_DATA	SmartCardReadAction	enum	ExecuteAndReadData
SC_XML_READ_BLOCK_DATA	SmartCardReadAction	enum	XmlReadBlockData
SC_STORE_DATA	SmartCardWriteAction	enum	StoreData
SC_STORE_PROGRAM	SmartCardWriteAction	enum	StoreProgram
SC_EXECUTE_DATA	SmartCardWriteAction	enum	ExecuteData
SC_XML_BLOCK_DATA	SmartCardWriteAction	enum	XmlBlockData
SC_SECURITY_FUSE	SmartCardWriteAction	enum	SecurityFuse
SC_RESET	SmartCardWriteAction	enum	Reset
SC_SUE_NO_CARD			
SC_SUE_CARD_PRESENT			
OPOS_ESC_READ	SmartCardRW	Int32	ExtendedErrorRead
OPOS_ESC_WRITE	SmartCardRW	Int32	ExtendedErrorWrite
OPOS_ESC_TORN	SmartCardRW	Int32	ExtendedErrorTorn
OPOS_ESC_NO_CARD	SmartCardRW	Int32	ExtendedErrorNoCard
OPOS_ETOT_NOROOM	HardTotals	Int32	ExtendedErrorNoRoom
OPOS_ETOT_VALIDATION	HardTotals	Int32	ExtendedErrorValidation
OPOS_STAT_BarcodePrintedCount	PosPrinter	System.String	StatisticBarcodePrintedCount
OPOS_STAT_BumpCount	BumpBar	System.String	StatisticBumpCount
OPOS_STAT_CommunicationErrorCount	PosCommon	System.String	StatisticCommunicationErrorCount
	PosCommon	System.String	StatisticDeviceCategory
OPOS_STAT_DrawerFailedOpenCount	CashDrawer	System.String	StatisticDrawerFailedOpenCount
OPOS_STAT_DrawerGoodOpenCount	CashDrawer	System.String	StatisticDrawerGoodOpenCount
OPOS_STAT_FailedDataParseCount	Micr	System.String	StatisticFailedDataParseCount
OPOS_STAT_FailedPaperCutCount	PosPrinter	System.String	StatisticFailedPaperCutCount
OPOS_STAT_FailedPrintSideChangeCount	PosPrinter	System.String	StatisticFailedPrintSideChangeCount
OPOS_STAT_FailedReadCount	Micr	System.String	StatisticFailedReadCount
	Msr	System.String	StatisticFailedReadCount
OPOS_STAT_FailedSignatureReadCount	SignatureCapture	System.String	StatisticFailedSignatureReadCount
	PosCommon	System.String	StatisticFirmwareRevision
OPOS_STAT_FormInsertionCount	PosPrinter	System.String	StatisticFormInsertionCount
OPOS_STAT_GoodReadCount	Micr	System.String	StatisticGoodReadCount
	Msr	System.String	StatisticGoodReadCount
OPOS_STAT_GoodScanCount	Scanner	System.String	StatisticGoodScanCount
OPOS_STAT_GoodSignatureReadCount	SignatureCapture	System.String	StatisticGoodSignatureReadCount
OPOS_STAT_GoodWeightReadCount	Scale	System.String	StatisticGoodWeightReadCount
OPOS_STAT_HomeErrorCount	PosPrinter	System.String	StatisticHomeErrorCount
OPOS_STAT_HoursPoweredCount	PosCommon	System.String	StatisticHoursPoweredCount
	PosCommon	System.String	StatisticInstallationDate
	PosCommon	System.String	StatisticInterface
OPOS_STAT_InvalidPINEntryCount	PinPad	System.String	StatisticInvalidPINEntryCount
OPOS_STAT_JournalCharacterPrintedCount	PosPrinter	System.String	StatisticJournalCharacterPrintedCount
	PosPrinter	System.String	StatisticJournalCoverOpenCount
OPOS_STAT_JournalLinePrintedCount	PosPrinter	System.String	StatisticJournalLinePrintedCount

OPOSでの名称	POS for .NET		
	クラス	パラメータ	
		型	名前
OPOS_STAT_KeyPressedCount	PosKeyBoard	System.String	StatisticKeyPressedCount
OPOS_STAT_LockPositionChangeCount	Keylock	System.String	StatisticLockPositionChangeCount
	PosCommon	System.String	StatisticManufactureDate
	PosCommon	System.String	StatisticManufacturerName
OPOS_STAT_MaximumTempReachedCount	PosPrinter	System.String	StatisticMaximumTempReachedCount
	PosCommon	System.String	StatisticMechanicalRevision
	PosCommon	System.String	StatisticModelName
OPOS_STAT_MotionEventCount	MotionSensor	System.String	StatisticMotionEventCount
OPOS_STAT_NVRAMWriteCount	PosPrinter	System.String	StatisticNVRAMWriteCount
OPOS_STAT_OnlineTransitionCount	LineDisplay	System.String	StatisticOnlineTransitionCount
OPOS_STAT_PaperCutCount	PosPrinter	System.String	StatisticPaperCutCount
OPOS_STAT_PrinterFaultCount	PosPrinter	System.String	StatisticPrinterFaultCount
OPOS_STAT_PrintSideChangeCount	PosPrinter	System.String	StatisticPrintSideChangeCount
OPOS_STAT_ReceiptCharacterPrintedCount	PosPrinter	System.String	StatisticReceiptCharacterPrintedCount
OPOS_STAT_ReceiptCoverOpenCount	PosPrinter	System.String	StatisticReceiptCoverOpenCount
OPOS_STAT_ReceiptLineFeedCount	PosPrinter	System.String	StatisticReceiptLineFeedCount
OPOS_STAT_ReceiptLinePrintedCount	PosPrinter	System.String	StatisticReceiptLinePrintedCount
	PosCommon	System.String	StatisticSerialNumber
OPOS_STAT_SlipCharacterPrintedCount	PosPrinter	System.String	StatisticSlipCharacterPrintedCount
OPOS_STAT_SlipCoverOpenCount	PosPrinter	System.String	StatisticSlipCoverOpenCount
OPOS_STAT_SlipLineFeedCount	PosPrinter	System.String	StatisticSlipLineFeedCount
OPOS_STAT_SlipLinePrintedCount	PosPrinter	System.String	StatisticSlipLinePrintedCount
OPOS_STAT_StampFiredCount	PosPrinter	System.String	StatisticStampFiredCount
OPOS_STAT_ToneSoundedCount	ToneIndicator	System.String	StatisticToneSoundedCount
	PosCommon	System.String	StatisticUnifiedPOSVersion
OPOS_STAT_UnreadableCardCount	Msr	System.String	StatisticUnreadableCardCount
OPOS_STAT_ValidPINEntryCount	PinPad	System.String	StatisticValidPINEntryCount

構造体

POS for .NETでは、メソッドの呼び出しで返されるデータをまとめた構造体型を定義しています。この構造体型が必要なのは、POS for .NETにおけるパラメータが**入力**のみだからです。一方、POS for .NETでは、構造体型は、まとめたデータのタイプセーフな処理を実現するためにも利用されます。UnifiedPOSのプロパティやメソッドによって返される、いくつかのデータをつなげた形の文字列は、新しく定義された構造体型のメンバに分解されます。

構造体はクラスに似ています。しかし、構造体は値の意味を含んでおり、ヒープの割当ても必要ありません。構造体に対する言語の概念は、
http://msdn.microsoft.com/library/en-us/csspec/html/vclrfcsharpspec_11.asp
に記述されています。

POS for .NETでは、次の構造体が定義されています。

CashCount 構造体

構造体***CashCount***は、払い出す現金の金種と枚数を含みます。

構造体のプロパティ

名前	説明
Count	紙幣または貨幣の枚数を保持します。
NominalValue	額面を保持します。
Type	通貨が紙幣か貨幣かを示します。

参照元

CashChanger.DepositCountsプロパティとして返される配列の要素型に使用されます。POS for .NETでのシグネチャは以下になります。

```
public abstract CashCount[] DepositCounts
```

CashChanger.DispenseCashメソッドのパラメータCashCounts配列の要素型に使用されます。POS for .NETでのシグネチャは以下になります。

```
public abstract void DispenseCash( CashCount[] cashCounts )
```

CashCounts 構造体

構造体***CashCounts***は、**CashCount** 型要素の配列と、現金の不一致があるかどうかの値を集約したものです。

構造体のプロパティ

名前	説明
Counts	*CashCount* データを保持します。
Discrepancy	true：*CashCount* に含まれない現金があります。 false：一致します。

参照元

CashChanger.ReadCashCountsメソッドの戻り値の型として使用されます。POS for .NETでのシグネチャは以下になります。

public abstract CashCounts ReadCashCounts()

CashUnits 構造体

CashChangerでサポートされている現金の金種を保持します。現金の金種は、紙幣と貨幣とで別々の2つのString配列に格納されます。

構造体のプロパティ

名前	説明
Bills	紙幣の各金種の枚数を保持します。
Coins	貨幣の各金種の枚数を保持します。

参照元

CashChanger.DepositCashListプロパティの戻り値の型として使用されます。POS for .NETでのシグネチャは以下になります。

public abstract CashUnits DepositCashList

CashChanger.CurrencyCashListプロパティの戻り値の型として使用されます。POS for .NETでのシグネチャは以下になります。

public abstract CashUnits CurrencyCashList

CashChanger.ExitCashListプロパティの戻り値の型として使用されます。POS for .NETでのシグネチャは以下になります。

public abstract CashUnits ExitCashList

DirectIOData 構造体

構造体***DirectIOData***は、**DirectIO**メソッドが戻す値を集約したものです。

構造体のプロパティ

名前	説明
Data	コマンドとサービスオブジェクトにより異なる特定の値。
Object	コマンドとサービスオブジェクトにより異なる特定のオブジェクト。

参照元

PosCommon.DirectIOメソッドの戻り値の型として使用されます。POS for .NETでのシグネチャは以下になります。

public abstract DirectIOData DirectIO(int *command***, int** *data***, object** *obj* **)**

FiscalDataItem 構造体

構造体***FiscalDataItem***は、**FiscalPrinter**カテゴリの**GetData**メソッドが返す値を集約したものです。

構造体のプロパティ

名前	説明
Data	データを表す文字列。
ItemOption	オプションの追加パラメータ。この引数の使用方法に関する詳細は、サービスオブジェクトベンダのマニュアルを参照してください。

参照元

FiscalPrinter.GetDataメソッドの戻り値の型として使用されます。POS for .NETでのシグネチャは以下になります。

public abstract FiscalDataItem GetData(FiscalData *dataItem***, int** *itemOption***)**

TotalsFileInfo 構造体

構造体***TotalsFileInfo***は、**HardTotals**デバイスカテゴリの情報を集約したものです。

構造体のプロパティ

名前	説明
Handle	トータルファイルのハンドル。
Size	トータルファイルのサイズ。

参照元

Totals.Findメソッドの戻り値の型として使用されます。POS for .NETでのシグネチャは以下になります。

public abstract TotalsFileInfo Find(string *fileName* **)**

VatInfo 構造体

構造体***VatInfo***は、**FiscalPrinter**カテゴリで使用する付加価値税情報を集約したものです。

構造体のプロパティ

名前	説明
Amount	付加価値税額を示します。
Id	付加価値税 ID。

参照元

FiscalPrinter.PrintRecPackageAdjustVoid メソッドのパラメータ*vatAdjustments*配列の要素型に使用されます。POS for .NETでのシグネチャは以下になります。

public abstract void PrintRecPackageAdjustVoid(
FiscalAdjustmentType *adjustmentType***, VatInfo[]** *vatAdjustments* **)**

FiscalPrinter.PrintRecPackageAdjustment メソッドのパラメータ*vatAdjustments*配列の要素型に使用されます。POS for .NETでのシグネチャは以下になります。

public abstract void PrintRecPackageAdjustment(
FiscalAdjustmentType *adjustmentType***, string** *description***,**
VatInfo[] *vatAdjustments* **)**

VideoMode 構造体

構造体***VideoMode***は、**RemoteOrderDisplay**デバイスカテゴリが使用するビデオユニットのためにサポートされるビデオモードを保持します。

構造体のプロパティ

名前	説明
Colors	色数。
Columns	列数。
IsColor	true：ビデオはカラー。false：カラーでない。
Rows	行数。

参照元

RemoteOrderDisplay.VideoModesListプロパティとして返される配列の要素型に使用されます。POS for .NETでのシグネチャは以下になります。

public abstract VideoMode[] VideoModesList

提供されるクラスライブラリの一覧

インターフェイスクラス

- インターフェイスライブラリは、コード上の機能は提供しません。デバイスクラスへのインターフェイスとしてのみ意味を持ちます。インターフェイスクラスは、UnifiedPOSで定義されたデバイスクラスのそれぞれに対して存在します。
- インターフェイスは、UnifiedPOS仕様書の標準に準拠してます。場合によっては、拡張機能も定義しています。
- インターフェイスクラスは、デバイススタティスティクスの管理、イベントを介したステータスレポート、および一般的なエラー状態のレポートに必要なすべての定数を定義します。
- インターフェイスクラスは、すべてのデバイスクラスに必要な列挙体を定義します。

Basic class

- Basic classは、インターフェイスクラスを継承し、複数のデバイスクラスに共通の機能を実装します。例えば、Basic classは、Open()、Claim()、Release()メソッドを実装します。Basic classは、UnifiedPOSで定義されたデバイスクラスのそれぞれに対して存在します。
- Basic classは、すべての共通のプロパティやメソッドを管理するだけでなく、アプリケーションへのイベント通知、デバイススタティスティクスの取得や保管を管理し、あらゆる種類のエラー処理に対応し、さらにサービスオブジェクトにハードウェア状態の変化状況を通知する機能を提供します。

Base class

- Base classは、Basic classを継承し、デバイスクラス固有の機能を実装しています。POS for .NET V 1.0には、8つのBase classがあります。デバイスサービスオブジェクトの供給者は、ハードウェア固有の機能を実装するだけで済みます。
- Base classは、Basic classの機能の上に、すべての種類のイベントを実装する（同時にイベント通知を管理する）ように作られています。また、すべてのデバイススタティスティクスを更新、管理します。プロパティとパラメータ値の有効性を管理（必要ならばアプリケーションにエラーを通知）します。さらに、アプリケーションへのデータイベント通知の一環として仕様ガイドラインに従ってすべてのデバイス固有のプロパティを更新します。加えて、ハードウェア固有の機能を提供する場合には、プロテクトされたメソッドとヘルパークラスをアプリケーションが使用できるような柔軟な構造で提供します。

戻り値

多くのPOS for .NET APIの呼び出しは、値を返します。例えば、共通メソッド

string CheckHealth (HealthCheckLevel *level*);

は、ヘルスチェックの結果を示す文字列を返します。POS for .NETにおけるパラメータは、**入力**のみです。

返されるプロパティ

アプリケーションによるメソッドの呼び出しは、プロパティの値を変更することがあります。または、メソッドは、UnifiedPOS仕様書の定義どおり、特定のステータス値を返します。

例えば、下記の場合を想定してください。

ソフトウェアベンダ（ISV）は、特定のプロパティの値を変更する可能性があるメソッドを呼び出します。その後の処理は、プロパティの新しい値に依存します。UnifiedPOSのOPOS実装では、ソフトウェアベンダは、まず1つのメソッドを呼び出し、その後、プロパティの値を返すもう1つのメソッドを呼び出します。

```
MethodThatChangesAProperty()
Dim MyProperty as Property
GetPropertyValue(MyProperty)
// GetPropertyValueのパラメータは
//  参照渡しです。
Select MyProperty
case ….
```

POS for .NETでは、ソフトウェアベンダは、下記のように、メソッドを呼び出し、その戻り値をテストします。（Visual Basic .NETの例）

```
Select MethodThatChangesAProperty()
Case ….
```

返されるリスト

メソッドは、値のリストを返すことがあります。OPOSでは、リストを返すメソッドは、リスト項目のデータの型にかかわらず、カンマで区切られた形の文字列を返します。アプリケーションは、必要に応じてデータ項目を文字列に変換し、カンマで区切ることで、文字列を組み立てなければなりません。

アプリケーションは、文字列を解析し、データ項目をリストに関連付けられた型にキャストする必要があります。例えば、

```
CHAR nChar = “,”;
int x;
int y = 0;
CHAR* pMyElements[];
CHAR* psCurrent;
for(x=0;x<len(sReturn);x++)
{
        if(sReturn[x] == nChar)
        {
                pMyElements[y] = psCurrent;
                y++;
        }
        psCurrent += sReturn[x];
}
// すべての戻り値の型が文字列であると想定しています
// そうでない場合、適切なデータ型へのキャストが必要です。
```

POS for .NETでは、配列はnative data typesです。データ要素を無理に型キャストする必要はありません。さらに、配列は、リストのいずれの項目にも容易にアクセスできる反復関数を提供します。

```
// 各項目を必要なときに使用する
SomeMethod(ReturnedArray[0]);
SomeOtherMethod(ReturnedArray[1]);
```

注:

SOからは、下記のコードが示すようにクローンを返させます。これは、データを保護するために必要です。

return SomeArrary.Clone();

結合された文字列でなく配列を返す理由は、以下のとおりです。

配列は、.NETのnative data typesであり、for each文で列挙することができます。

区切られた形の文字列を組み立てて解析することは、メンテナンスが必要なより多くのコードが要求され、保守を要し、バグを生み出す可能性を増加させます。

配列を使用すると、コードの意図がより明瞭になります。

例:

UnifiedPOS、OPOS、POS for .NET間の違いをさらに示すために、プロパティ**PosPrinter.CharacterSetList**を見てみましょう。このプロパティは、UnifiedPOSでは以下のようなシグネチャを持ちます。

CharacterSetList: string { read-only, access after open }

UnifiedPOSでは、このプロパティは、コードページ番号のリストをカンマで区切られた文字列で返します。アプリケーションプログラムは、文字列を解析して、コードページ番号を抽出し、必要に応じて整数値に変換する必要があります。POS for .NETでは、プロパティ**PosPrinter**.**CharacterSetList**は、以下のメソッドシグネチャを持ちます。

public abstract int[] CharacterSetList

これは、コードページ番号のリストを整数型の配列として返します。文字列を解析してコードページ番号を整数値に変換する必要がありません。この手法は、アプリケーションプログラマにとって、より型安全（タイプセーフ）であり、処理がより容易です。

パラメータの主な違い

OPOSが定数を利用するのに対して、POS for .NETでは、列挙体を利用します。OPOSが結合された文字列を利用するのに対して、POS for .NETでは、配列を利用します。POS for .NETは、native integer typesを使用します。OPOSが、妥当かつ想定される範囲よりも多くのバイト数を要するデータ型の値を利用するのに対して、POS for .NETは、表現されるデータ型と一致する「ぴったりの大きさ」の変数を利用します。

POS for .NETでは、UnifiedPOSのメソッドが2～3の値しか取りえないパラメータを持つ場合、これを複数のPOS for .NETメソッドに分割しています。例えば、**FiscalPrinter**のメソッド**printReport**は、UnifiedPOSでは以下のようなシグネチャを持ちます。

printReport(reportType: *int32*, startNum: *string*, endNum: *string*): void

パラメータ**reportType**は、FPTR_RT_ORDINAL、またはFPTR_RT_DATEのいずれかの値しか取りません。後続する2つのパラメータは、FPTR_RT_DATEの場合は、日付を表す文字列として解釈し、そうでない場合は、整数値として使用する必要があります。

POS for .NETでは、**reportType**パラメータは省略されます。その代わりに、異なるシグネチャの**printReport**()として定義された2つの新しいメ

ソッドが導入されています。これらは、より型安全（タイプセーフ）です。

void PrintReport(DateTime *startDate*, DateTime *endDate*)

void PrintReport(int *startNumber*, int *endNumber*)

次の表は、UnifiedPOSのメソッド、パラメータに対するPOS for .NETのメソッド、パラメータの違いの一覧です。違いが列挙体を使用するかどうかだけのメソッドは、一覧から省略しています。

UnifiedPOSのメソッド	POS for .NETのシグネチャ
CashChanger	
dispenseCash(**cashCounts**: *string*): **void**	**void DispenseCash**(**CashCount[]** *cashCounts*)
FiscalPrinter	
getData(**dataItem**: *int32*, **inout optArgs**: *int32*, **inout data**: *string*): **void**	**FiscalDataItem GetData**(**FiscalData** *dataItem*, **int** *itemOption*)
printPeriodicTotalsReport(**date1**: *string*, **date2**: *string*): **void**	**void PrintPeriodicTotalsReport**(**DateTime** *startingDate*, **DateTime** *endingDate*)
printRecItem(**description**: *string*, **price**: *currency*, **quantity**: *int32*, **vatInfo**: *int32*, **unitPrice**: *currency*, **unitName**: *string*): **void**	**void PrintRecItem**(**string** *description*, **decimal** *price*, **decimal** *quantity*, **int** *vatId*, **decimal** *unitPrice*, **string** *unitName*)
printRecPackageAdjustment(**adjustmentType**: *int32*, **description**: *string*, **vatAdjustment**: *string*): **void**	**void PrintRecPackageAdjustment**(**FiscalAdjustmentType** *adjustmentType*, **string** *description*, **VatInfo[]** *vatAdjustments*)
printRecPackageAdjustVoid(**adjustmentType:** *int32*, **vatAdjustment**: *string*): **void**	**void PrintRecPackageAdjustVoid**(**FiscalAdjustmentType** *adjustmentType*, **VatInfo[]** *vatAdjustments*)
printReport(**reportType**: *int32*, **startNum**: *string*, **endNum**: *string*): **void**	**void PrintReport**(**DateTime** *startDate*, **DateTime** *endDate*)
printReport(**reportType**: *int32*, **startNum**: *string*, **endNum**: *string*): **void**	**void PrintReport**(**int** *startNumber*, **int** *endNumber*)
printReport(**reportType**: *int32*, **startNum**: *string*, **endNum**: *string*): **void**	**void PrintReport**(**int** *startNumber*)
setDate(**date**: *string*): **void**	**void SetDate**(**DateTime** *newDate*)
setVatValue(**vatID**: *int32*, **vatValue**: *string*): **void**	**void SetVatValue**(**int** *vatId*, **decimal** *vatRate*)

プロパティシグネチャの主な違い

いくつかのプロパティは、UnifiedPOSとは異なるPOS for .NETシグネチャを持ちます。これらは、カンマで区切られたリスト文字列の代わり

に、配列、または構造体を使用します。次の表で、そのプロパティを示します。

UnifiedPOSのプロパティ	POS for .NETのシグネチャ
CashChanger	
CurrencyCodeList	public abstract string[] CurrencyCodeList
CurrencyCashList	public abstract CashUnits CurrencyCashList
DepositCodeList	public abstract string[] DepositCodeList
DepositCounts	public abstract CashCount[] DepositCounts
ExitCashList	public abstract CashUnits ExitCashList
CheckScanner	
QualityList	public abstract int[] QualityList
FiscalPrinter	
PredefinedPaymentLines	public abstract string[] PredefinedPaymentLines
POSPrinter	
CharacterSetList	public abstract int[] CharacterSetList
FontTypefaceList	public abstract string[] FontTypefaceList
RecBarCodeRotationList	public abstract Rotation[] RecBarCodeRotationList
RecBitmapRotationList	public abstract Rotation[] RecBitmapRotationList
SlpBarCodeRotationList	public abstract Rotation[] SlpBarCodeRotationList
SlpBitmapRotationList	public abstract Rotation[] SlpBitmapRotationList
RemoteOrderDisplay	
VideoModesList	public abstract VideoMode[] VideoModesList

関連情報

サンプルは、POS for .NET SDKに含まれます。
（http://www.microsoft.com/downloads/details.aspx?FamilyID=ADAA1129-5CB1-415E-B339-E508FCA55CA0&displaylang=en でダウンロード可能）

PosExplorer API

アプリケーションは、**PosExplorer**を利用することで、インストールされたPOSデバイスの一覧を取得し、それらデバイスをオープン、またはそのインスタンスを作成し、デバイスがシステムから着脱されたときにプラグアンドプレイイベントを受け取ることができます。

PosExplorerプロパティ

PosRegistryKey プロパティ

形式　public static string PosRegistryKey

説明　HKEY_LOCAL_MACHINEからの相対位置で示されたPOS for .NETコンフィギュレーションルートレジストリキー。

StatisticsFile プロパティ

形式　public static string StatisticsFile

説明　デバイススタティスティクスが置かれたファイルへのパス。

SynchronizingObject プロパティ　R/W

形式　public ISynchronizeInvoke SynchronizingObject

説明　IsynchronizeInvokeオブジェクトへの設定かその値。

PosExplorerメソッド

CreateInstance メソッド

形式　public PosDevice CreateInstance(DeviceInfo *device*)

説明　DeviceInfoオブジェクトのプロパティ値で与えられる情報に基づいて、デバイスをインスタンス化します。

パラメータ	説明
Device	インスタンスを作成しようとするデバイスを記述するオブジェクト。DeviceInfoクラスのインスタンスです。DeviceInfoは、デバイスに対するCompatibility、Description、HardwareIDなどのプロパティを含みます。

GetDevice メソッド (string)

形式　public DeviceInfo GetDevice(string *type*)

説明　指定された種類のデバイスを取得します。

パラメータ	説明

Type	**DeviceType**ヘルパークラスによって定義されたUnifiedPOSのデバイスの種類中の1つを格納する文字列。

システムに存在するその種類のデバイスは、1つだけでなければなりません。あるいは、複数存在する場合は、1つが既定のデバイスとして設定されている必要があります。指定された種類のデバイスが複数存在し、既定のデバイスが設定されていない場合は、**PosLibraryException**がスローされます。

GetDeviceのこのシグネチャは、POS for .NETシステムでデバイスを取得して、インスタンス化する最も単純な方法を示します。デバイスを1つ取得し、そのサービスオブジェクトをインスタンス化するには、アプリケーションは以下のことをするだけです。

PosExplorerのインスタンスを作成します。

上記メソッドシグネチャを用いて、**GetDevice**を呼び出します。

CreateInstanceを呼び出します。

POS for .NETは、指定された種類のデバイス、または、その種類のデバイスが複数存在する場合には、その種類の中で既定として設定されたデバイスを初期化します。

GetDevice メソッド (string, string)

形式 public DeviceInfo GetDevice(string *type*, string *logicalName*)

説明 指定された種類および名前（またはエイリアス）のデバイスを取得します。

パラメータ	説明
Type	**DeviceType**ヘルパークラスによって定義されたUnifiedPOSデバイスタイプの1つを格納する文字列。
LogicalName	デバイスの論理名、またはエイリアス。

GetDevices メソッド

形式 public DeviceCollection GetDevices()

説明 システムにインストールされている、すべてのPOSデバイスを取得します。

GetDevices メソッド (DeviceCompatibilities)

形式　public DeviceCollection GetDevices(DeviceCompatibilities *compatibility*)

説明　システムにインストールされている、すべてのPOSデバイスを、互換性レベルに基づいて取得します。

パラメータ	説明
Compatibility	互換性レベルを示すDeviceCompatibilities列挙体。

GetDevices メソッド (string)

形式　public DeviceCollection GetDevices(string *type*)

説明　指定された種類のすべてのPOSデバイスを取得します。

パラメータ	説明
Type	DeviceTypeヘルパークラスによって定義されたUnifiedPOSデバイスタイプの1つを格納する文字列。

GetDevices メソッド (string, DeviceCompatibilities)

形式 public DeviceCollection GetDevices(string *type*, DeviceCompatibilities *compatibility*)

説明 指定された種類のすべてのPOSデバイスを、互換性レベルに基づいて取得します。

パラメータ	説明
Type	DeviceTypeヘルパークラスによって定義されたUnifiedPOSデバイスタイプの1つを格納する文字列。
Compatibility	互換性レベルを示すDeviceCompatibilities列挙体。

Refresh メソッド

形式 public void Refresh()

説明 接続されたPOSデバイスの一覧を再列挙し、内部データの構造体を再構築します。

PosExplorerイベント

DeviceAddedEvent イベント

形式 public event DeviceChangedEventHandler DeviceAddedEvent;

説明 POSデバイスがシステムに追加された時にアプリケーションに通知されます。

DeviceAddedEventは、サービスオブジェクトがインストールされているPOSデバイスに対してのみ通知されます。

イベントハンドラは、追加デバイスに対するDeviceInfoオブジェクトを含むDeviceChangedEventArgs型の引数を受け取ります。

DeviceRemovedEvent イベント

形式 public event DeviceChangedEventHandler DeviceRemovedEvent;

説明 POSデバイスがシステムから削除された時にアプリケーションに通知されます。

DeviceRemovedEventは、サービスオブジェクトがインストールされているPOSデバイスに対してのみ通知されます。

イベントハンドラは、削除デバイスに対する**DeviceInfo**オブジェクトを含む**DeviceChangedEventArgs**型の引数を受け取ります。

グローバル設定

PosExplorerは、グローバルコンフィグレーションファイル（config.xml）を読み込みます。このファイルでは、アプリケーション開発者は、プラグアンドプレイ、および非プラグアンドプレイデバイスへのエイリアスを指定でき、非プラグアンドプレイサービスオブジェクトに対して物理デバイスを定義できます。

また、グローバルコンフィグレーションファイルでは、アプリケーション開発者は、非プラグアンドプレイサービスオブジェクトに対して複数の物理デバイスを定義し、それらのエイリアスや、COMポートなどのデバイスパスを指定することもできます。これにより、アプリケーション開発者は、特定の物理デバイスに、非プラグアンドプレイサービスオブジェクトのターゲットを絞ることができます。

サービスオブジェクトレジストリ

OPOSでは、サービスオブジェクトの設定情報は、レジストリに保存されます。POS for .NETでは、設定情報はConfig.xmlに保存されます。POS for .NETは、PosExplorerを介して、COMサービスオブジェクトのレジストリ情報へのシームレスなアクセスを可能にし、共通コントロールライブラリは、レジストリ設定情報を収集する役割を果たします。

サービスオブジェクトの利用

OPOS

コントロールオブジェクトは、対応するサービスオブジェクトへのアプリケーションインターフェイスに相当します。UnifiedPOS標準は、コントロールオブジェクトに対していかなる規定もしていません。ですが、
http://www.monroecs.com/oposccos.htmにあるOPOSコントロールオブジェクトを使用すること、または最低でもテスト対象とすることが推奨されています。さらに、このサイトには、すべてのデバイスクラスを集約したActiveX®コントロールがあります。これは、一般にコモンコントロールオブジェクトと呼ばれています。

OPOSの下では、ハードウェアベンダ、ソフトウェアベンダ、およびOEM業者が、コントロールオブジェクトの独自のバージョンを作成し、基準となるコモンコントロールオブジェクトの使用やテストをしないことが慣例になっています。これは、ハードウェア、サービス、アプリケーション間の互換性の問題につながります。

OPOSの実装は、以下の手順からなります。

コントロールオブジェクトのインスタンスをインスタンス化します。

コントロールオブジェクトを呼び出します。

- サービスオブジェクトをロードするために、名前でオープン
- クレイム
- イネーブル

デバイスによっては、デバイス固有の機能についてデバイスとやり取りする前に、プロパティを読み込む、または設定する必要があることに注意してください。

POS for .NET

POS for .NETでサービスオブジェクトをインスタンス化するには、次のようにします。
PosExplorerオブジェクトをインスタンス化します。

PosExplorer.GetDevice、またはGetDevicesメソッドを使用して、システムに接続されたデバイスを表す1つ以上のDeviceInfoオブジェクトからなるリストを取得します。

ロードしたいデバイスのDeviceInfoを渡して、PosExplorer.CreateInstanceを呼び出します。

返されたサービスオブジェクトのメソッド、プロパティを呼び出します。

供給される**PosExplorer**ツールは、サービスオブジェクトファクトリの役を果たすヘルパークラスです。開発者は、次のようにインスタンス化します。POSExplorer.GetDevice(…);

この手法は、以下のような利点をもたらします。

一連の機能（「POS for .NETの機能」を参照）のサポートに必要な基盤を実現します。

アプリケーションを単純化します。すなわち、数行のコードで、サービスオブジェクトを動的にインスタンス化することができます。

ほとんどの場合、特定のメーカーの周辺機器のハードウェアに関する詳細な知識が必要ではなくなります。

アプリケーションは、デバイスに実際に接続されていて、利用可能なPOS周辺機器のリストを容易に取得できます。（プラグアンドプレイ対応の場合）

アプリケーション側には、.NET SOとOPOS SOの違いがありません。

サービスオブジェクトの制作

POS for .NET

次の3つの方法があります。

インターフェイスクラスからサービスオブジェクトを派生する

Basic classからサービスオブジェクトを派生する

Base classからサービスオブジェクトを派生する

それぞれの方法について、サービスオブジェクト製作者に要求される作業のレベルは異なります。例えば、インターフェイスクラスから派生する場合、サービスアプリケーションが実装すべきコードの量は最も多くなりますが、サービスオブジェクトの動作に対して最も細かい制御を可能にします。Basic classからの派生では、サービスアプリケーションは、デバイスの中心的な機能だけしか実装する必要がありません。Basic classには、共通の機能が既に用意されています。Base class

からの派生では、サービスアプリケーションがやるべきことは、ハードウェア固有の機能を実装するだけです。デバイスクラスの基本的な機能は既に用意されています。

ステータス、状態モデル、および例外

次に示すように、共通で使う列挙体、イベント、プロパティの中には、ステータス、エラーコード、状態モデルが設定されているものがあります。

StatusUpdateEvent

特定のクラス固有の状態やステータス変数が変更されたときに発生するイベントです。

ControlState

現在の状態を格納する列挙体です。可能な値は以下のものです。

Closed

Idle

Busy

Error

例外

すべてのPOS for .NETメソッドの呼び出しやプロパティへのアクセスは、失敗時に**PosControlException**をスローする可能性があります。ただし、プロパティ**DeviceControlVersion**、**DeviceControlDescription**、および**State**へのアクセスを除きます。それ以外の種類の例外は、スローされません。

PosControlExceptionは、名前空間**Microsoft.PointOfService**で定義されており、**System.Exception**を拡張します。

Publicプロパティ

名前	説明
ErrorCode	エラー例外の原因を表すエラーコード。エラーコード一覧を参照。
ErrorCodeExtended	エラー例外の原因を表す拡張エラーコード。これには、サービス固有の値が格納されることがあります。

コンストラクタには、以下のように定義されるいくつかの種類があります。

PosControlException (string *message*, ErrorCode *errorCode*)

PosControlException (string *message*, ErrorCode *errorCode*, Exception *innerException*)

PosControlException (string *message*, ErrorCode *errorCode*, int *errorCodeExtended*)

PosControlException (string *message*, ErrorCode *errorCode*, int *errorCodeExtended*, Exception *innerException*)

パラメータは以下のように定義されます。

パラメータ	説明
errorCode	POS for .NETのエラーコード。**ErrorCode**取得メソッドを介してアクセスします。
errorCodeExtended	拡張エラーコードが格納されることがあります。選択されたコンストラクタで与えられていない場合は、0がセットされます。**ErrorCodeExtended**取得メソッドを介してアクセスします。
message	エラーの説明文です。選択されたコンストラクタで与えられていない場合は、*errorCode*と*errorCodeExtended*パラメータから形成されます。スーパークラスの取得メソッド**Message**、またはメソッド**ToString**を介してアクセスします。
innerException	元の例外です。POS for .NETサービスがPOS for .NET以外の例外をキャッチした場合、適当な*errorCode*が選択され、元の例外にはこのパラメータで参照します。それ以外の場合、nullがセットされます。継承された取得メソッド**InnerException**を介してアクセスします。

デバイス共有モデルについて

POS for .NETのデバイス共有モデルは、複数のアプリケーションが部分的または全面的に共有できるデバイスのほか、同時には1つのアプリケーションのみが排他的に使用するデバイスをサポートします。すべてのPOS for .NETサービスオブジェクトは、1つ以上のアプリケーションにオープンされます。しかし、サービスオブジェクトで実行できるアプリケーションの動作の中には、デバイスへのアクセス権を獲得するアプリケーションを1つに制限するものがあります。

排他使用デバイス

もっとも一般的なデバイスタイプは「排他使用デバイス」で、一例がPOSプリンタです。物理的特性、動作上の特性により、このデバイスは同時には1つのアプリケーションでしか利用できません。アプリケーションは、大半のメソッド、プロパティ、イベントが有効になる前に、**Claim**メソッドを呼び出して排他アクセスできるようにしなければなりません。排他アクセス権を獲得するまえに、メソッドを呼び出したり、プロパティを設定したりすると、*Illegal*エラーが発生し、イベントはアプリケーションに通知されません。

密接に関連する2つのアプリケーションが共有方式で排他使用デバイスを使用したい場合、一方のアプリケーションは一部の時間だけデバイスを排他アクセスして、その後、他方のアプリケーションも使用できるように、排他アクセス権を解放するという方法もあります。

Claimメソッドが再び呼び出されると、設定可能なデバイスの特性は、**Release**メソッド時の状態に復元されます。復元される例として、ラインディスプレイの輝度、磁気ストライプリーダの読み取りトラック、POSプリンタの行あたりの文字数などがあります。POSPrinterのセンサプロパティのような**State**特性は復元されません。その代わり、現在の状態に設定されます。

共有可能デバイス

いくつかのデバイスは共有可能デバイスです。一例がキーロックです。共有可能デバイスの場合、複数のアプリケーションでそのメソッドを呼び出して、プロパティにアクセスすることが可能となります。また、デバイスをオープンしたすべてのアプリケーションにイベントが通知されます。ただし、共有可能デバイスを排他アクセスしたアプリケーションにより、一部のメソッドやプロパティのアクセスを制限することや、そのアプリケーションだけにイベントを通知させることもできます。

イベント

POS for .NETは、UnifiedPOSのイベントを、マルチキャストデリゲートを持つ標準の.NETイベントとして実装します。

イベントは、デバイスの様々な動作や変化、あるいはデバイスの着脱をアプリケーションに通知します。イベントには以下5つがあります。

イベント	説明
DataEvent	入力データがデバイスクラス固有のプロパティに格納されています。
ErrorEvent	イベント駆動入力中または非同期出力中にエラーが発生しました。
StatusUpdateEvent	デバイスの状態変化を報告します。
OutputCompleteEvent	非同期出力が正常に完了しました。
DirectIOEvent	このイベントは、サービスオブジェクトプロバイダによって定義され、本仕様でカバーできないものを提供します。

サービスオブジェクトはイベントが発生するとイベントをキューイング（Queue）します。キューイングされたイベントは通知可能な状態になるとアプリケーションに通知（Deliver）されます。イベントの引き渡しを遅らせる要因には以下のものがあります。

アプリケーションがFreezeEventsプロパティをtrueに設定している。

イベントタイプがDataEventまたは入力ErrorEventで、DataEventEnabledプロパティがfalseである。

“POS for .NET Implementation Reference”（英語版）では、イベントの通知に関する用語の解説が下記のとおり記載され、仕様書内で一貫して使い分けられています。日本版仕様書においては、本項「イベント」以外ではこれらの使い分けは行っていませんが、参考のため掲載しています。

備考: このドキュメントでは、下記のイベントに関する用語が使用されています。

Queue SOがアプリケーションに対しイベントの通知（Fire）の必要性を決定したとき、SOが内部のイベントキューにイベントを詰め込みます。

Deliver イベントキューが空でなく、キューの先頭のイベントについてのすべての条件が揃ったとき、このイベントはキューから外され、アプリケーションに対するイベント通知要求が実行されます。

Fire キューイング（Queue）と通知（Deliver）の組み合わせと言えます。時にこの用語は大まかに使用され、これらのステップの片方だけしか意味しない場合もあります。読者は前後関係からこれらを識別しなければなりません。

イベントのキューの管理に関する規定は以下のとおりです。

サービスオブジェクトは、デバイスがイネーブルである間は新しいイベントをキューイング（Queue）するのみです。

サービスオブジェクトは、アプリケーションがReleaseメソッド（排他使用デバイス）またはCloseメソッド（すべてのデバイス）を呼び出すまで、キューイング（Queue）されたイベントの通知（Deliver）を行う可能性があります。また、前記メソッド実行時には残存しているイベントはすべて削除されます。

入力デバイスでは、ClearInputメソッドはデータと入力エラーイベントをクリアします。イベントハンドラ内で、アプリケーションは、プロパティにアクセスしたりメソッドの呼び出しを行えます。しかし、アプリケーションはReleaseメソッドやCloseメソッドをイベントハンドラ内から呼び出してはなりません。Releaseメソッドはイベントハンドリング（イベントを通知（Deliver）させるスレッドを含んでいる）を閉鎖してしまい、Closeは戻る前にイベントハンドリングを閉鎖しなければならないためです。

入力モデルについて

POS for .NETの入力モデルは、イベント駆動入力をサポートします。イベント駆動入力では、入力データは**DeviceEnabled**がtrueに設定された後に受信できるようになります。受信されたデータは、条件が満たされている場合に、アプリケーションに通知される**DataEvent**としてキューイングされます。データ受信時に**AutoDisable**プロパティがtrueの場合、コントロールは**DeviceEnabled**をfalseに設定して自動的にデバイスをディセーブルにします。これはサービスオブジェクトが更に入力をキューイングするのを妨げ、可能な場合は物理的にデバイスをディセーブルにします。

アプリケーションは、デバイスからの入力を受信する準備ができたら**DataEventEnabled**プロパティをtrueに設定します。入力を受信すると（通常、ハードウェア割込みの結果として）、コントロールは、**DataEvent**をキューイングし、通知します。（入力がすでにキューイングされている場合は、**DataEvent**が通知されます。）このイベントは、数値パラメータを介して入力ステータス情報を含むことがあります。コントロールは、イベントを通知する直前に、入力データと、必要に応じてその他の情報をデバイス固有のプロパティに格納します。

イベントを通知する直前に、コントロールは**DataEventEnabled**プロパティをfalseに設定することによって、それ以上のデータイベントを不可能にします。これによりアプリケーションが現在の入力と関連のプロパティを処理している間、それ以降の入力データはコントロールによってキューイングされます。アプリケーションは、現在の入力の処理を終了し、次のデータを受け付ける準備ができたとき、

DataEventEnabledをtrueに設定してイベントの通知を再開するようにします。

入力デバイスが排他使用デバイスの場合、アプリケーションは、デバイスが入力の読み取りを開始する前に、そのデバイスの排他アクセス権の獲得とイネーブルの両方を行わなければなりません。

共有可能な入力デバイスの場合、アプリケーションは、デバイスが入力の読み取りを開始する前に、そのデバイスをオープンして、イネーブルにしなければなりません。アプリケーションは、コントロールが**DataEvent**を使用してデータを送る前に、**Claim**メソッドを呼び出してデバイスの排他アクセスを要求しなければなりません。イベント駆動入力を受信したとき、アプリケーションがデバイスを排他アクセスしていない場合、アプリケーションがデバイスの排他アクセス権を獲得する（かつ**DataEventEnabled**プロパティがtrueになる）まで、入力はキューイングされます。これによって、複数のアプリケーションがデバイスを順序よく共有でき、アプリケーション間で入力フォーカスを効果的に受け渡すことができます。

イベント駆動入力によりデータの入力処理をしているときに、コントロールがエラーを見つけると、コントロールはその状態をErrorに変更し、アプリケーションにエラー状態を警告するために、1～2個の**ErrorEvent**をキューイングします。**DataEventEnabled**プロパティがtrueになるまで、このイベントは通知されないので、アプリケーションは順序よく処理することができます。エラーイベントは、次のエラー位置を示すデータを付帯して通知されます。

InputData：1つ以上の**DataEvent**イベントがキューイングされている間にエラーが発生した場合に、キューイングされます。このイベントは、すべての**DataEvent**より先にキューイングされます。このイベントにより、アプリケーションは入力を直ちにクリアするか、オプションでユーザにエラーの警報を出して、バッファリングされた入力済のデータを処理することができます。

後者のケースはスキャナコントロールで有効です。ユーザに直ちにエラー警報を出すことができるので、エラーが解除されるまで後続のアイテムはスキャンされません。エラー復旧を実行する前に、以前スキャンしたアイテムを正常に処理できます。

Input：エラーが発生し、利用できるデータがないときに通知されます。（一般的な実装では、このイベントをイベントキューの末尾にセットします。）エラーが発生したときに入力データの一部をすでにキューイングしている場合、エラー位置が**InputData**の**ErrorEvent**がまずキューイング、通知され、このエラーイベントはすべての**DataEvent**が通知された後に通知されます。（“**InputData**”イベントが通知され、アプリケーションのイベントハンドラが”Clear”応答した場合、本“**Input**”イベントは通知されません。）

コントロールは、以下のいずれかが発生した場合に、Error状態を終了します。

アプリケーションがInputのErrorEventを終了する場合。

アプリケーションがInputDataのErrorEventをClearのErrorResponseで終了する場合。

アプリケーションがClearInputメソッドを呼び出した場合。

コントロールによっては、イベント駆動入力を開始するためにメソッドを呼び出す必要があります。コントロールが入力データを受け取った後、一般的には、入力を再初期化するためにメソッドを再度呼び出すまでは、以降の入力データを受け付けません。例として、MICRとシグニチャキャプチャデバイスがあります。このタイプのイベント駆動入力は「非同期入力」と呼ばれることもあります。

DataCountプロパティはコントロールでキューイングされた**DataEvent**の数を得るために参照されます。

コントロールがキューイングしたすべての入力データは、**ClearInput**メソッドを呼び出せば削除できます。**ClearInput**は、共有可能デバイスでは**Open**後に、排他使用デバイスでは**Claim**後に呼び出せます。

一般的なイベント駆動入力モデルは、入力データを直接返すメソッドやプロパティを含むデバイスクラスの定義を特に定めていません。デバイスクラスの中には、より直感的または柔軟に操作できるようなメソッドやプロパティを定義しているものがあります。例として、キーロックデバイスがあります。このタイプの入力は「同期入力」と呼ばれることもあります。

出力モデルについて

POS for .NETの出力は、同期と非同期の2つの出力タイプからなります。各デバイスクラスは、一方または両方のタイプをサポートすることも、両方ともサポートしないこともあります。

同期出力

デバイス出力を迅速に実行したい場合にこの出力タイプが有効です。そのメリットは単純さにあります。

アプリケーションはクラス固有のメソッドを呼び出して出力を実行します。サービスオブジェクトは出力が完了するまで制御を返しません。

非同期出力

デバイス出力が低速のハードウェアで、対話的処理を必要とする場合に、この出力タイプが有効です。デバイスが出力を実行しているとき

に、アプリケーションが別の処理を実行できることから、確実な応答性があることがメリットです。

アプリケーションはクラス固有のメソッドを呼び出して出力を開始します。サービスオブジェクトは、プログラムメモリ内のリクエストをバッファリングして、物理デバイスがそれを受け取り、処理できるようになると、すぐに物理デバイスに送付し、このリクエストの識別子を**OutputId**プロパティに設定して、できる限り迅速に戻ります。デバイスがリクエストを正常に終了すると、POS for .NETは、**OutputCompleteEvent**を通知します。このイベントのパラメータは完了したリクエストの**OutputId**です。

非同期リクエストを実行しているときにエラーが発生すると、**ErrorEvent**が通知されます。アプリケーションのイベントハンドラは、未処理の出力を再試行するか、またはそれをクリアできます。**ErrorEvent**の処理中は、サービスオブジェクトはError状態です。（備考：エラーを引き起こす原因が除去されていない場合は、サービスオブジェクトは直ちにError状態に戻り、別の**ErrorEvent**を通知します。）非同期出力はFIFOベースで実行されます。バッファリングされた出力はすべて（非同期出力をすべて含む）、**ClearOutput**メソッドを呼び出すことによって削除できます。**OutputCompleteEvent**は、クリアされた出力に対しては通知されません。このメソッドは（可能ならば）処理中の出力も停止します。

デバイス電源通知モデルについて

アプリケーションでは、使用しているデバイスの電源状態を知る必要がある場合があります。この状態は、**PowerState**列挙体によって管理されます。

注: このモデルは、PCやPOS端末本体の電源状況（「バッテリ・オン」や「バッテリ・ロー」など）を通知することを目的にしてい

ません。これらの情報の通知は、現在PosPower列挙体によって管理されています。

モデル

POS for .NETは、デバイスの電源を以下の4つの状態に分類します。

Online　デバイスは電源オン状態でかつレディ状態です。これは「使用可能」状態を示します。

Off　デバイスは電源オフ状態または本体に未接続状態です。これは「使用不可能」状態を示します。

Offline　デバイスは電源オン状態ですが、ノットレディ状態か応答不可状態です。ボタンを押してオンライン状態にする必要があるかもしれません。または、本体からの要求に応答できない状態かもしれません。これは「使用不可能」状態を示します。

上記の状態の組み合わせとして、下記の状態が定義されています。

OffOffline　デバイスは電源オフ状態、またはオフライン状態です。SOはこれらの2つの状態を識別することができません。

電源通知は、デバイスがオープン、排他アクセス権の獲得（排他デバイスの場合）を経てイネーブル状態となっている場合にのみ機能します。

備考：イネーブル／ディセーブル vs. 電源状態

これら2つの状態は、全く別のもので通常は関連がありません。POS for .NETでは、「イネーブル」／「ディセーブル」は論理的な状態であるのに対し、電源状態は物理的な状態であると定義しています。デバイスによっては、論理的には「イネーブル」であっても、物理的には「オフライン」かもしれません。または、論理的には「ディセーブル」であっても、物理的には「オンライン」かもしれません。物理的な電源状態にかかわらず、POS for .NETは、デバイスがイネーブル状態の場合のみ状態を通知します。（SOは、一般的にイネーブル状態の間しかデバイスと通信できないので、この制限が必要です。）デバイスが「オフライン」の場合でも、SOは、デバイスを「イネーブル」にしようとして失敗することがあります。しかし、一度「イネーブル」になると、SOは電源状態が変わってもデバイスを「ディセーブル」にはしません。

電源通知プロパティ

POS for .NETのデバイス電源通知モデルは、すべてのデバイスクラスに対して下記の共通APIを追加しています。

CapPowerReportingプロパティ：デバイスの通知可能レベルを識別します。このプロパティの値は、以下の**PowerReporting**列挙体の値のいずれかです。

None	SOはデバイスの電源状態の判断ができません。よって、電源通知機能は機能しません。
Standard	SOは、OffOffline（電源オフまたはオフライン）とOnlineの2種類の電源状態を判断し通知が可能です。
Advanced	SOは、OnLine、OffLine、Offの3種類すべての電源状態を判断し通知が可能です。

PowerState列挙体：判断が可能ならば、SOによって現在の電源状態にセットされます。値は下記のうちのいずれかです。

Unknown

Online

Off

Offline

OffOffline

PowerNotifyプロパティ：**StatusUpdateEvent**や**PowerState**列挙体を介しての電源通知機能を有効にするために、アプリケーションがこのプロパティをセットします。このプロパティはデバイスがイネーブルになる前（すなわち**DeviceEnabled**がtrueにセットされる前）にのみセットが可能です。この制限により、アプリケーションに対する悪影響を発生させることなく、電源通知機能の単純な実装が可能となっています。アプリケーションは、電源通知を受けるか否かのいずれかですが、それを切り換える仕組みは必要ありません。このプロパティは、**PowerNotification**列挙体を返し、その値は、Disabled、またはEnabledのいずれかです。

DeviceEnabledプロパティと電源通知機能

CapPowerReportingがNone以外の値で、かつ**PowerNotify**がEnabledの場合、下記の記述内容が、**DeviceEnabled**に追加適用されます。

コントロールが**DeviceEnabled**をfalseからtrueに変更すると、電源状態の監視が開始されます。

デバイスがOnlineの場合:

- PowerStateは、Onlineに設定されます。
- StatusUpdateEventが、StatusUpdateEventArgs.Statusプロパティに Onlineが設定されて、通知されます。

デバイスの電源状態がOff、Offline、OffOfflineの場合、コントロールはイネーブルが失敗したと判断して、**PosControlException**をスローし、**ErrorCode**にNoHardware、またはOfflineを設定します。

しかし、デバイスをイネーブルにできない他の条件がなく、またコントロールがイネーブル要求に対し正常終了を返す場合:

- PowerStateは、Off、Offline、またはOffOfflineに設定されます。
- StatusUpdateEventが、StatusUpdateEventArgs.Statusプロパティに Off、Offline、またはOffOfflineが設定されて、通知されます。

デバイスインフォメーションレポーティングモデルについて

システム管理エージェントのようなPOSアプリケーションでは、POS端末に取り付けられた様々なPOSデバイスの現在の構成や使用状態を監視する必要がある場合があります。

構成情報は、デバイスのシリアル番号、ファームウェア版数や接続方法などです。POSプリンタデバイスの使用状態の例は、印刷行数、稼働時間、用紙カット回数などです。スキャナデバイスの使用状態の例は、スキャン回数、稼動時間などです。磁気ストライプリーダデバイスの使用状態の例は、読み取り成功回数、読み取り失敗回数、稼働時間などです。

情報はデバイス自身に収集格納される場合や、サービスオブジェクトによって収集され、POS端末やストアコントローラに格納される場合があります。

複数のアプリケーション（例えば、POSアプリケーションとシステム管理アプリケーション）が同一のデバイスからスタティスティクスを獲得したい場合、それぞれが必要な時に整然とデバイスにアクセスできるように、両方のアプリケーションが適切な注意を払わなければなりません。これは、デバイスへのアクセスが必要になった時に、**Claim**メソッドを利用して、**DeviceEnabled**プロパティにtrueをセットすることで、その後デバイスへのアクセスが不要となった時に

DeviceEnabledプロパティにfalseをセットして、**Release**メソッドを利用することによって行います。この方式によるデバイスアクセスの調整については、各アプリケーション自身の責任で行います。

スタティスティクスレポーティングプロパティとメソッド

UnifiedPOSのデバイスインフォメーションレポーティングモデルは、すべてのデバイスクラスに下記の共通プロパティとメソッドを追加しています。

- CapStatisticsReportingプロパティ：デバイスのスタティスティクスレポーティング能力を示します。falseの場合、そのデバイスに関する有効なスタティスティクスはありません。これは以前の版数の仕様と互換性のあるサービスと同じです。trueの場合、そのデバイスに関するいくつかのスタティスティクスが有効です。
- CapUpdateStatisticsプロパティ：収集されたスタティスティクス（またはその一部）をアプリケーションがリセット/変更できるかどうかを示します。このプロパティは、CapStatisticsReportingがtrueのときのみ有効です。CapUpdateStatisticsがfalseの場合、アプリケーションがリセット/変更できるスタティスティクスはありません。trueの場合、スタティスティクス（の一部）をアプリケーションがリセット/変更できます。
- ResetStatisticsメソッド：CapStatisticsReportingとCapUpdateStatisticsの両方がtrueのときのみ呼び出し可能です。このメソッドはリセット可能なデバイススタティスティクスの1つ、複数、あるいはすべてを0にリセットします。
- RetrieveStatisticsメソッド：CapStatisticsReportingがtrueのときのみ呼び出し可能です。このメソッドは収集されているデバイススタティスティクスの1つ、複数、あるいはすべてを取得します。
- UpdateStatisticsメソッド：CapStatisticsReportingとCapUpdateStatisticsの両方がtrueのときのみ呼び出し可能です。このメソッドはリセット可能なデバイススタティスティクスの1つ、複数、あるいはすべてを指定の値に変更します。

POS for .NETコンポーネントの解説

POS for .NETのデータの型

以降の説明で用いるパラメータと戻り値の型は次のとおりです。

型	.NET Framework の型	説明	UnifiedPOSの型
bool	System.Boolean	ブーリアン値（true、または false）	boolean
decimal	System.Decimal	金額	currency
int	System.Int32	32 ビット符号付き整数	int32
Object	System.Object	オブジェクトの参照。これは、通常クラス階層の最上位の 1 つ下のサブクラスで、**directIO** や **DirectIOEvent** のデバイスサービスに固有のパラメータに使用します。	object
string	System.String	Unicode 文字からなる変更できない固定長文字列	string
byte[]	要素型が System.Byte の System.Array	任意のバイナリデータ	binary

POS for .NETの共通プロパティ、メソッド、イベント、スタティスティクス、定数

共通プロパティ

名前	型
AutoDisable	bool
CapPowerReporting	PowerReporting
CapStatisticsReporting	bool
CapUpdateStatistics	bool
CheckHealthText	string
Claimed	bool
Compatibility	DeviceCompatibilities
DataCount	int
DataEventEnabled	bool
DeviceDescription	string
DeviceEnabled	bool
DeviceName	string
DevicePath	string
FreezeEvents	bool
OutputId	int
PowerNotify	PowerNotification
PowerState	PowerState
ServiceObjectDescription	string
ServiceObjectVersion	System.version
State	ControlState
SynchronizingObject	System.ComponentModel.ISynchronizeInvoke

共通プロパティの詳細は、後述します。

共通メソッド

以下は、共通メソッドのPOS for .NET実装の独自の定義です。

CheckHealth (**HealthCheckLevel** *level*);

Claim (**int** *timeout*);
ClearInput ();
ClearOutput ();
Close ();
DeleteConfigurationProperty (**string** *propertyName*);
DirectIO (**int** *command*, **int** *data*, **object** *obj*);
GetConfigurationProperty (**string** *propertyName*);
Invoke (**Delegate** *method*, **object**[] *args*);
Open ();
Release ();
ResetStatistics ();
ResetStatistics (**StatisticCategories** *statistics*);
ResetStatistics (**string**[] *statistics*);
RetrieveStatistics (**StatisticCategories** *statistics*);
RetrieveStatistics (**string**[] *statistics*);
RetrieveStatistic (**string** *statistic*);
UpdateStatistic (**string** *name*, **object** *value*);
UpdateStatistics (**Statistic**[] *statistics*);
UpdateStatistics (**StatisticCategories** *statistics*, **object** *value*);

共通メソッドの詳細は、後述します。

共通イベント

.NET Frameworkのイベントは、デリゲートモデルに基づいています。デリゲートモデル、アプリケーションにおけるイベントの使用方法、およびクラスからのイベントの発生方法についての詳細は、
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconevents.asp
をご覧ください。

以下は、共通イベントのPOS for .NET実装の独自の定義です。

DirectIOEventHandler **DirectIOEvent**;
StatusUpdateEventHandler **StatusUpdateEvent**;
DataEventHandler **DataEvent**;
OutputCompleteEventHandler **OutputCompleteEvent**;
DeviceErrorEventHandler **DeviceErrorEvent**;

共通イベントの詳細は、後述します。

共通スタティスティクス

StatisticUnifiedPOSVersion= “UnifiedPOSVersion”;
StatisticDeviceCategory= “DeviceCategory”;
StatisticManufacturerName= “ManufacturerName”;
StatisticModelName = “ModelName”;
StatisticSerialNumber= “SerialNumber”;
StatisticManufactureDate= “ManufactureDate”;
StatisticMechanicalRevision= “MechanicalRevision”;
StatisticFirmwareRevision= “FirmwareRevision”;
StatisticInterface = “Interface”;
StatisticInstallationDate= “InstallationDate”;
StatisticHoursPoweredCount= “HoursPoweredCount”;
StatisticCommunicationErrorCount = “CommunicationErrorCount”;

共通定数

int **WaitForever**= -1;
int **StatusPowerOnline**= 2001;
int **StatusPowerOff**= 2002;
int **StatusPowerOffline**= 2003;
int **StatusPowerOffOffline**= 2004;
int **ExtendedErrorStatistics**= 280;

第 2 章 （ OLE 版 ）

共通プロパティ、メソッド、イベント

一覧

プロパティ

名称	*版数*	*型*	*アクセス*	*注*
AutoDisable	1.2	Boolean	R/W	1
BinaryConversion	1.2	Long	R/W	
CapCompareFirmwareVersion	1.9	Boolean	R	
CapPowerReporting	1.3	Long	R	
CapStatisticsReporting	1.8	Boolean	R	
CapUpdateFirmware	1.9	Boolean	R	
CapUpdateStatistics	1.8	Boolean	R	
CheckHealthText	1.0	String	R	
Claimed	1.0	Boolean	R	
DataCount	1.2	Long	R	1
DataEventEnabled	1.0	Boolean	R/W	1
DeviceEnabled	1.0	Boolean	R/W	
FreezeEvents	1.0	Boolean	R/W	
OpenResult	1.5	Long	R	
OutputID	1.0	Long	R	2
PowerNotify	1.3	Long	R/W	
PowerState	1.3	Long	R	
ResultCode	1.0	Long	R	
ResultCodeExtended	1.0	Long	R	
State	1.0	Long	R	
ControlObjectDescription	1.0	String	R	
ControlObjectVersion	1.0	Long	R	
ServiceObjectDescription	1.0	String	R	
ServiceObjectVersion	1.0	Long	R	
DeviceDescription	1.0	String	R	
DeviceName	1.0	String	R	

注：**1.** イベント駆動入力のあるデバイスのみに使用。
　　2. 非同期出力のデバイスのみに使用。

メソッド

名称	*版数*
Open	1.0
Close	1.0
ClaimDevice	1.0
ReleaseDevice	1.0
CheckHealth	1.0
ClearInput	1.0
ClearOutput	1.0
CompareFirmwareVersion	1.9
DirectIO	1.0
ResetStatistics	1.8
RetrieveStatistics	1.8
UpdateFirmware	1.9
UpdateStatistics	1.8

イベント

名称	*版数*
DataEvent	1.0
DirectIOEvent	1.0
ErrorEvent	1.0
OutputCompleteEvent	1.0
StatusUpdateEvent	1.0

概説

この節では、後続のデバイスカテゴリの多くに共通のプロパティ、イベント、メソッドを示します。

各デバイスクラスの一覧の節では、そのクラスに適用しない共通プロパティ、イベント、メソッドに「*適用外*」の記載をしています。これらはそのクラスのコントロールには存在しません。

プロパティ

AutoDisableプロパティ R/W *第1.2版にて追加*

形式 BOOL AutoDisable;

説明 このプロパティはイベント駆動入力デバイスに適用されます。これは、入力データの受信を制御するための追加オプションをアプリケーションに提供します。アプリケーションがひとつの入力のみを受信し処理したい場合や、一度に一つの入力のみを受信したい場合、このプロパティをTRUEに設定します。

TRUEの場合、**DataEvent**として通知されるデータをサービスオブジェクトが受信するとすぐに、**DeviceEnabled**＝FALSEに設定されます。次の入力が受信できるようにするためには、アプリケーションは**DeviceEnabled**＝TRUEに設定しなければなりません。

FALSEの場合、サービスオブジェクトはデータ受信時にデバイスを自動的にディセーブルにしません。これは第1.2版より前のOPOSコントロールの動作です。

このプロパティは**Open**メソッドによりFALSEに初期化されます。

戻り値 本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティの設定に成功しました。

参照 「**OpenPOSコントロール概要**」

BinaryConversion プロパティ　R/W　*第1.2版にて追加*

形式　　LONG BinaryConversion;

説明　　OPOSはBStringを使用してマルチキャラクタ入力／出力を受け渡します。BStringは安全にテキストデータに使用できます。Bstringはアプリケーションと OPOSコントロールの間の受け渡し時に、OLEはUnicodeコードから、またはUnicodeコードへと、言語固有の変換を行います。BStringがバイナリデータの受け渡しに使用される場合、アプリケーションでのBString文字内のデータバイトがコントロールでの対応データと適合しない時は、これらの変換はデータを変更する場合があります。 Unicodeコードキャラクタがアプリケーションやコントロールに渡される時、各コンポーネントの言語固有の違いが誤解釈を引き起こすため、この不適合はBStringポインタが使用されている場合にも起こり得ます。（これは、MBCSコードページ932を使用してる日本語で報告されましたが、他の言語でも起こり得ます。）

0x00から0x7Fまでの文字は言語固有の変換という心配もなく引き渡されます。0x80から0xFFまでの文字のみが時々不正な変換を引き起こします。

本仕様書では、**BinaryConversion**に影響されるプロパティとメソッドのパラメータについては、各々のプロパティとメソッドの説明箇所に、以下の行が記述されています。

> このデータのフォーマットは**BinaryConversion**プロパティの値に依存します。詳細は**BinaryConversion**プロパティを参照してください。

BinaryConversionの値は以下の通りです。

値	意味
OPOS_BC_NONE	データは変換されずにBstring一文字は1バイトで格納されます。（これはデフォルトで、第1.2版より前のOPOSサービスオブジェクトの動作です。）
OPOS_BC_NIBBLE	各バイトは二つの文字に変換されます。（このオプションでバイナリとASCII文字間の最速変換ができます。）

各データバイトは以下のように変換されます。　第一文字＝0x30＋データバイトの7-4ビット　第二文字＝0x30＋データバイトの3-0ビット

例：バイト値154=0x9Aは文字0x39 0x3A（文字列”9:”）の文字に変換されます。この変換方法は、バイト値154=0x9Aをコード0x39 0x41（文字列の”9A”）の文字に変換する、一般的な16進値のASCII文字変換とは異なります。

OPOS_BC_DECIMAL各バイトは三つの文字に変換されます。（このオプションでVisual Basicやそれに似た言語用にバイナリとASCII文字間の最も簡単な変換ができます。）
VAL（*string*）はASCIIからバイナリへ変換するために各３文字で使用されます。各バイトから3つのASCII文字を作成するために RIGHT(“^^”+STR(*byte*),3)が使用され、’^’はスペース文字を表します。

例１：バイト値154=0x9Aは、文字0x31 0x35 0x34になります（＝文字列”154”）。

例２：バイト値8は、文字0x30 0x30 0x38になります（＝文字列”008”）。

サービスオブジェクトに必要な事項は以下の通りです。

（１）サービスオブジェクトがASCIIからバイナリへ変換する時は、STR（*byte*）は先行スペースを作成するので、先行スペースまたはASCIIゼロを用意しておかなければなりません。(例えば、アプリケーションは”^^8^27”を受け渡すとき、’^’はスペース文字を表し、8（0x08）と27（0x1B）の２つのバイトと解釈されます．）

（2）サービスオブジェクトがバイナリからASCIIに変換する時は、常に各バイトを正確に三つのASCII10進文字（0x30から0x39の範囲で）に変換しなければなりません。

BinaryConversion設定時（OPOS_BC_NONEでない場合）で、プロパティやメソッドパラメータが**BinaryConversion**の指定が適用される場合は、アプリケーションに以下の義務があります。

プロパティの設定とメソッドパラメータの受け渡し前に、文字列データを**BinaryConversion**値に指定されたフォーマットに変換する。

プロパティの設定とメソッドパラメータの受け取り後に、文字列データを**BinaryConversion**値に指定されたフォーマットから変換する。

よりよく理解するための変換の指針として、データの流れが入力モデルか出力モデルかに従うこと。出力モデルならば、アプリケーションは、上記の前者の義務を守ること。入力モデルならば、上記の後者の義務を守ること。

このプロパティは**Open**メソッドによりOPOS_BC_NONEに初期化されます。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティの設定に成功しました。
OPOS_E_ILLEGAL	不正な値が指定されました。

CapCompareFirmwareVersion プロパティ　*第1.9版にて追加*

形式　BOOL CapCompareFirmwareVersion;

説明　TRUEならばサービス/デバイスは、ファームウェアファイルのバージョンと物理デバイス内のファームウェアバージョンを比較する機能を有します。

このプロパティはOpenメソッドにより初期化されます。

参照　CompareFirmwareVersionメソッド

CapPowerReporting プロパティ　*第1.3版にて追加*

形式　LONG CapPowerReporting;

説明　デバイスの通知能力を識別します。

電源通知能力を示す値は下記のとおりです。

値	意味
OPOS_PR_NONE	SOはデバイスの電源状態の判断ができません。よって、電源通知機能は機能しません。
OPOS_PR_STANDARD	SOは、２種類の電源状態を判断し通知が可能です。―― OFF_OFFLINE（電源オフまたはオフライン）とONLINE。
OPOS_PR_ADVANCED	SOは、３種類のすべての電源状態判断し通知が可能です。―― ONLINE、OFFLINE とOFF。

このプロパティはOpenメソッドにより初期化されます。

参照　「デバイス電源通知モデルについて」、PowerStateプロパティ、PowerNotifyプロパティ

CapStatisticsReporting プロパティ *第1.8版にて追加*

形式　BOOL CapStatisticsReporting;

説明　TRUEならばデバイスは使用状態に関する様々なスタティスティクスを収集提供可能です。収集され通知される情報はデバイスに固有であり、RetrieveStatisticsメソッドを使用して取得されます。

このプロパティはOpenメソッドにより初期化されます。

参照　「デバイスインフォメーションレポーティングモデルについて」、RetrieveStatisticsメソッド

CapUpdateFirmware プロパティ　*第1.9版にて追加*

形式　BOOL CapUpdateFirmware;

説明　TRUEならばデバイスのファームウェアUpdateFirmwareメソッドにより更新されることを可能とします。

このプロパティはOpenメソッドにより初期化されます。

参照　UpdateFirmwareメソッド

CapUpdateStatistics プロパティ　*第1.8版にて追加*

形式　BOOL CapUpdateStatistics;

説明　TRUEならばデバイスのスタティスティクスのいくつかは、ResetStatisticsメソッドを使用して 0 にリセットできるか、UpdateStatisticsメソッドを使用して変更できます。CapStatisticsRepotingプロパティがFALSEならばCapUpdateStatisticsプロパティもFALSEです。

このプロパティはOpenメソッドにより初期化されます。

参照　「デバイスインフォメーションレポーティングモデルについて」、CapStatisticsRepotingプロパティ、ResetStatisticsメソッド、UpdateStatisticsメソッド

CheckHealthText プロパティ

形式　BSTR CheckHealthText;

説明　直前に呼び出したCheckHealthメソッドの結果を保持します。以下に診断結果の例を示します。

“Internal HCheck: Successful”

“External HCheck: Not Responding”

“Interactive HCheck: Complete”

最初のCheckHealthメソッド呼び出し以前にはこの値は初期化されません。

参照　CheckHealthメソッド

Claimed プロパティ

形式　BOOL Claimed;

説明　TRUE：デバイスの排他アクセス権が獲得されています。
FALSE：デバイスはほかのアプリケーションと共有できるよう解放されています。

多くの場合、デバイスの排他アクセス権を獲得した後で、メソッドやプロパティへのアクセス、イベントの発生が可能となります。

Claimedプロパティの値はOpenメソッドによってFALSEに初期化されます。

参照　「OpenPOSコントロール概要」、「デバイス共有モデルについて」、ClaimDeviceメソッド、ReleaseDeviceメソッド

ControlObjectDescription プロパティ

形式　BSTR ControlObjectDescription;

説明　コントロールオブジェクトとそれを製造している会社を示す文字列です。

このプロパティでコントロールオブジェクトを識別します。

このプロパティはいつでも読取り可能です。

参照　ControlObjectVersionプロパティ

ControlObjectVersion プロパティ

形式　LONG ControlObjectVersion;

説明　コントロールオブジェクトバージョン番号

このプロパティはコントロールオブジェクトバージョン番号を保持します。次の3つのバージョンレベルがあります。

バージョンレベル	説明
メジャー	百万の位 デバイスクラスに対してOPOSメジャーバージョンレベルを変更すると、かなりのインタフェース拡張が行われ、古くなったインタフェースのサポートを前メジャーバージョンレベルから削除します。
マイナ	千から十万までの位 デバイスクラスに対してOPOSマイナバージョンレベルを変更すると、多少のインタフェース拡張が行われ、このメジャーバージョンレベルでの前インタフェースのスーパーセットを提供しなければなりません。
ビルド	一から百までの位 コントロールオブジェクトディベロッパが設定する内部レベルです。COインプリメントに修正を行うと更新されます。

バージョン番号の例を次に示します。

```
1002038
```

この値はバージョン"1.2.38"として表示され、コントロールオブジェクトのメジャーバージョンが1、マイナバージョンが2、ビルドが38を意味します。

このプロパティはいつでも読取り可能です。

参照　ControlObjectDescriptionプロパティ

備考

デバイスクラスの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サービスオブジェクトのメジャーバージョン番号と一致する限り、そのクラスのサービスオブジェクトとともに動作[1]します。メジャーバージョン番号が一致するけれども、コントロールオブジェクトのマイナーバージョン番号がサービスオブジェクトのマイナーバージョン番号より大きい場合は、コントロールオブジェクトはサービスオブジェクトのリリース版では、サポートされていないいくつかの新しいメソッドやプロパティをサポートしている場合があります。

以下の規定は、コントロールオブジェクトのリリースでサポートされていて、サービスオブジェクトの旧リリースではサポートされていないAPIに適応します。

サポートされていないプロパティの読み取り：コントロールオブジェクトはプロパティの初期化されていない値を返します。（初期化されていないプロパティのデフォルト値は「メソッドとプロパティの使い方」の項を参照願います。）

サポートされていないプロパティの書き込み：コントロールオブジェクトは次のように返しますが、サポートされていないプロパティの書き込みやメソッド呼び出しが行われたことを認識しなければなりません。

アプリケーションが**ResultCode**プロパティを読み取る場合、コントロールオブジェクトはOPOS_E_NOSERVICEの値を返さなければなりません（サービスオブジェクトから現在の**ResultCode**を読み込むのではなく）。**ResultCode**が、次のプロパティ書き込みやメソッド呼び出しで設定されるまで、本処理を行わなければなりません。

サポートされていないメソッドの呼び出し：コントロールオブジェクトはOPOS_E_NOSERVICEの値を返します。しかし、サポートされていないプロパティの書き込みやメソッド呼び出しが行われたことを認識しなければなりません。

アプリケーションが**ResultCode**プロパティを読み取る場合、コントロールオブジェクトは、OPOS_E_NOSERVICEの値を返さなければなりません（サービスオブジェクトから現在の**ResultCode**を読み込むのではなく）。**ResultCode**が、次のプロパティ書き込みやメソッド呼び出しで設定されるまで、本処理を行わなければなりません。

---

[1] **注意**：日本版仕様の第1.0版とその他の版では、インクルードファイルのシンボル定義値の変更等があり、同一メジャー番号の場合でも動作しないデバイスクラスが存在します。

DataCountプロパティ　　*第1.2版にて追加*

形式　　LONG DataCount；

説明　　コントロールでキューイングされている**DataEvent**の数が入っています。

アプリケーションは、 **DataCount**を参照することにより保留されているデバイスからの入力の有無を確認できます。保留されている入力は、アプリケーションが他の処理中であったり、イベントが凍結していたりなどの原因でまだイベント通知が発行されていません。

このプロパティは**Open**メソッドによりゼロに初期化されます。

参照　　「**入力モデルについて**」、**DataEvent**イベント

DataEventEnabled プロパティ　R/W

形式　　BOOL DataEventEnabled;

説明　　TRUE：入力データを受信するとすぐに**DataEvent**が通知されます。TRUEに変更すると、入力データがすでにバッファリングされていれば、**DataEvent**が即座に通知されます。（ただし、「即座の」通知を遅らせる他の条件もあることに注意が必要です。もし、**FreezeEvents**がTRUEか、またはアプリケーションで他のイベントを既に処理中の場合、**DataEvent**は状況が変化するまでSO内にキューイングされたままの状態となります。）

FALSE：後でアプリケーションに送れるよう、入力データはバッファリングされます。また、入力エラーが発生しても、**DataEventEnabled**がFALSEの間、**ErrorEvent**は通知されません。

このプロパティは**Open**メソッドによってFALSEに初期化されます。

戻り値　　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティの設定に成功しました。

参照　　「**入力モデルについて**」、**DataEvent**イベント

DeviceDescription プロパティ

形式　　BSTR DeviceDescription;

説明　　デバイスを識別する文字列です。

このプロパティは、デバイスとそれに関連する情報を示します。

このプロパティはOpenメソッドによって初期化されます。

参照　　DeviceNameプロパティ

DeviceEnabled プロパティ　R/W

形式　BOOL DeviceEnabled;

説明　TRUE：デバイスはイネーブルにされています（動作状態です）。TRUEに変更するとイネーブルにされます。

FALSE：デバイスはディセーブルにされています。FALSEに変更すると、可能ならば物理的にディセーブルにされます。後続の入力は廃棄され、出力動作を行うことはできません。

このプロパティを変更しても、普通、物理的には出力デバイスに影響しません。しかし、整合性を保つため、出力デバイスを使用する前に、アプリケーションがこのプロパティをTRUEに設定しなければなりません。

第1.3版以降：**DeviceEnabled**がTRUEの間、デバイスの電源状態が通知されます。
詳細は**「デバイス電源通知モデルについて」**を参照して下さい。

このプロパティは**Open**メソッドによりFALSEに初期化されます。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティの設定に成功しました。
OPOS_E_NOTCLAIMED	排他使用デバイスをイネーブルにするには、その前に排他アクセス権を獲得しなければなりません。
その他の値	**ResultCode**の項目を参照してください。

参照　**「OpenPOSコントロール概要」**

DeviceName プロパティ

形式　　BSTR DeviceName;

説明　　デバイスを識別する文字列です。

このプロパティは、デバイスとそれに関連する情報を示します。**DeviceDescription**の短縮バージョンで、30文字までに制限されています。

DeviceNameは、通常、アプリケーションのメッセージボックスにデバイスを識別する時に、**DeviceDescription**の文字列では長すぎる場合に使用します。

このプロパティは**Open**メソッドにより初期化されます。

参照　　DeviceDescriptionプロパティ

FreezeEvents プロパティ　R/W

形式　BOOL FreezeEvents;

説明　TRUEの場合、コントロールがイベントを通知しないよう、アプリケーションがリクエストしています。凍結が解除されるまで、イベントはコントロールが保持します。

FALSEの場合、アプリケーションにイベントを通知できます。凍結されていた間に保持されたイベントは、**FreezeEvents**をFALSEに変更すると、そのイベントが通知されます。[2]

イベントによる割り込みが望ましくないコードシーケンスに対して、アプリケーションはイベントの凍結を選択できます。

このプロパティは**Open**メソッドによりFALSEに初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	このプロパティの設定に成功しました。

[2] 実際には、イベントの通知をアプリケーションに延期することもできます。コントロールコンテナがイベント通知の凍結をコントロールにリクエストできます。たとえば、モーダルダイアログボックスがアクティブのとき、VisualBasicがこの機能を利用できます。したがって、FreezeEventがFALSEで、コンテナがイベント凍結をリクエストしていなかった場合に、イベントが発生します。コンテナが開始したイベント凍結については、本マニュアルのどこでも言及していません。アプリケーションがそれを認識することはほとんどなく、直接制御することはできないからです。その他ｺﾝﾃﾞｼｮﾝについては、“イベント”の節に説明されています。

OpenResult プロパティ　　*第1.5版にて追加*

形式　　LONG OpenResult;

説明　　最新のOpenメソッドでの結果を保持しています。

OpenResultの値は下記。

値	意味
OPOS_SUCCESS	オープンが成功しました。
OPOS_OR_ALREADYOPEN	コントロールはすでにオープンされています。
OPOS_OR _REGBADNAME	レジストリに指定したデバイス名称キーが存在しません。
OPOS_OR _REGPROGID	デバイス名キーのデフォルト値が読めないか、そこに保持されたプログラマティクIDを有効なクラスIDに変換できませんでした。
OPOS_OR _CREATE	サービスオブジェクトインスタンスを生成できなかったか、IDispatchインターフェースを取得できませんでした。
OPOS_OR _BADIF	サービスオブジェクトは指定版数で要求されるひとつ、もしくはひとつ以上のメソッドをサポートしていません。
OPOS_OR _FAILEDOPEN	サービスオブジェクトはオープン呼び出しでエラーのステータスを返しました、しかしながらそれ以上の定義されたエラーコードを返しませんでした。
OPOS_OR _BADVERSION	サービスオブジェクトのメジャーバージョンはコントロールオブジェクトのメジャーバージョンに適合していません。

以下の値は、サービスオブジェクトがオープン呼び出しでエラーコードを返した時に返されます。サービスオブジェクトは、利用可能ならこれらの内の一つを返すか、その他の値を定義するかを選択します。（サービスオブジェクトがこれらの値をどのように返すかの詳細は、Control Programmer's Guide のGetOpenResultの記述を参照。サービスオブジェクトがGetOpenResultを実装していなければ、OpenResultはOPOS_OR_FAILEDOPENを返します。）

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、オープンでなんらかの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でしょう。

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。

このプロパティは**Open**メソッドで初期化されます。

OutputID プロパティ

形式　LONG OutputID;

説明　直前に開始した非同期出力の識別子を保持します。

メソッドが非同期出力の開始に成功すると、コントロールはリクエストに識別子を割り当てます。出力が完了すると、**OutputCompleteEvent**を通知して、その出力IDをパラメータとして渡します。

出力ID番号は、コントロールによって割り当てられ、未処理非同期出力の出力の中でユニークであることが保証されます。IDについてその他のことは規定されません。

参照　「**出力モデルについて**」、OutputCompleteEventイベント

PowerNotify プロパティ　R/W　*第1.3版にて追加*

形式　LONG PowerNotify;

説明　アプリケーションから設定された電源通知機能のタイプです。

電源通知機能を示す値は下記のとおりです。

値	意味
OPOS_PN_DISABLED	コントロールはアプリケーションに対していかなる電源通知をも提供しません。電源通知に関するStatusUpdateEventsは通知されず、またPowerStateプロパティにも何も設定されません。
OPOS_PN_ENABLED	DeviceEnabledがTRUEに設定されると、コントロールは電源通知に関するStatusUpdateEventsの通知、PowerStateプロパティの更新が行われます。機能レベルは、CapPowerReportingの値に依存します。

PowerNotifyプロパティは、デバイスがディセーブルの間、すなわちなDeviceEnabledプロパティがFALSEの間のみ設定が可能です。

このプロパティはOpenメソッドによりOPOS_PN_DISABLEDに初期化されます。この設定により、OPOSの従来バージョンとの互換性を保ちます。

戻り値　本プロパティ設定時、次の値のいすれかがResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	このプロパティの設定に成功しました。
OPOS_E_ILLEGAL	次のいずれかが発生しました： • デバイスは既にイネーブル。 • **PowerNotify** = OPOS_PN_ENABLED だが **CapPowerReporting** = OPOS_PR_NONE
その他	ResultCodeの項目を参照してください。

参照　「デバイス電源通知モデルについて」、CapPowerReportingプロパティ、PowerStateプロパティ

PowerState プロパティ　　*第1.3版にて追加*

形式　　LONG PowerState;

説明　　判断が可能であれば、現在のデバイスの電源状態が設定されます。

電源状態を示す値は下記のとおりです。

値	意味
OPOS_PS_UNKNOWN	下記の理由のいずれかにより、デバイスの電源状態の判断ができません。 • **CapPowerReporting** = OPOS_PR_NONE、デバイスは電源通知機能をノンサポート。 • **PowerNotify** = OPOS_PN_DISABLED、電源通知機能がディセーブル。 • **DeviceEnabled** = FALSE、電源状態監視はデバイスがイネーブルになるまで動作せず。
OPOS_PS_ONLINE	デバイスは電源オンでかつレディ状態です。 **CapPowerReporting** = OPOS_PR_STANDARD、または、OPOS_PR_ADVANCED の場合、この値が返されます。
OPOS_PS_OFF	デバイスは電源オフ、または本体に接続されていません。 **CapPowerReporting** = OPOS_PR_ADVANCEDの場合のみ、この値が返されます。
OPOS_PS_OFFINE	デバイスは電源オンですが、ノットレディ状態か要求に対する応答が不可能状態です。 **CapPowerReporting** = OPOS_PR_ADVANCEDの場合のみ、この値が返されます。
OPOS_PS_OFF_OFFLINE	デバイスは電源オフ、またはオフライン状態です。 **CapPowerReporting** = OPOS_PR_STANDARDの場合のみ、この値が返されます。

このプロパティは**Open**メソッドによりOPOS_PS_UNKNOWNに初期化されます。**PowerNotify**がイネーブルにセットされ、**DeviceEnabled**がTRUEの場合、このプロパティはSOが検出した電源状態の検出に従い更新されます。

参照　「デバイス電源通知モデルについて」、CapPowerReportingプロパティ、PowerNotifyプロパティ

ResultCode プロパティ

形式　LONG ResultCode;

説明　このプロパティは各メソッドが設定します。書込み可能プロパティを設定したときも設定されます。

このプロパティはいつでも読取り可能です。Openメソッドを呼び出すまでは、値OPOS_E_CLOSEDを戻します。

特定のエラーに対し、下記の結果コードの内、ひとつ以上が有効であると考えられます。そのような展開でのエラー報告の優先順位は下記の通りです。

OPOS_E_CLAIMED

OPOS_E_NOTCLAIMED

OPOS_E_DISABLED

結果コードの値は次のとおりです。

値	意味
OPOS_SUCCESS	正常動作。
OPOS_E_CLOSED	クローズされているデバイスにアクセスしようとしました。
OPOS_E_CLAIMED	別のプロセスが排他アクセスしているデバイスにアクセスしようとしました。このアクセスを行えるには、その前に別のプロセスが排他アクセス権を解放しなければなりません。排他使用デバイスの場合、アプリケーションがデバイスの排他アクセス権を獲得していないと、アクセスができません。
OPOS_E_NOTCLAIMED	メソッドまたはプロパティ設定処理を使用する前に、排他アクセス権の獲得をしなければならない排他使用デバイスにアクセスしようとしました。デバイスを別のプロセスがすでに排他アクセスしている場合には、ステータスOPOS_E_CLAIMEDが戻されます。
OPOS_E_NOSERVICE	コントロールがサービスオブジェクトと通信できません。おそらく、セットアップエラーかコンフィギュレーションエラーを修正しなければなりません。
OPOS_E_DISABLED	デバイスをディセーブルしているときには動作を実行できません。

OPOS_E_ILLEGAL　デバイスに無効な動作か、サポートされていない動作を実行しようとしたか、無効なパラメータ値を使用しました。

OPOS_E_NOHARDWARE　デバイスをシステムに接続していないか、電源が入っていません。

OPOS_E_OFFLINE　デバイスがオフラインです。

OPOS_E_NOEXIST　ファイル名（またはほかの指定値）が存在していません。

OPOS_E_EXISTS　ファイル名（またはほかの指定値）がすでに存在しています。

OPOS_E_FAILURE　デバイスがシステムに接続され、電源が入っていて、オンラインですが、リクエストされた処理をデバイスが実行できません。

OPOS_E_TIMEOUT　デバイスからの応答を待ち合わせていたサービスオブジェクトがタイムアウトしたか、サービスオブジェクトからの応答を待ち合わせていたコントロールがタイムアウトしました。

OPOS_E_BUSY　現在のSOの状態は、この要求を受け付けられません。例えば、非同期出力が実行中の場合、いくつかのメソッドは受け付けられません。

OPOS_E_EXTENDED　クラス固有エラー状態が発生しました。**ResultCodeExtended**プロパティでエラー状態コードを確認できます。

参照　「**ステータス、結果コード、状態モデル**」

ResultCodeExtended プロパティ

形式　　LONG ResultCodeExtended;

説明　　ResultCodeがOPOS_E_EXTENDEDの場合、このプロパティにデバイスクラスの説明で記述されているクラス固有のエラー情報値が設定されます。

ResultCodeが別の値の場合、サービスオブジェクトがSO固有の値をこのプロパティに設定できます。これらの値に意味があるのは、アプリケーションがSO固有の値を追加して処理する場合だけです。

参照　　ResultCodeプロパティ

ServiceObjectDescription プロパティ

形式　　BSTR ServiceObjectDescription;

説明　　デバイスをサポートしているサービスオブジェクトと、そのデバイスを製造している会社を示す文字列です。

このプロパティはOpenメソッドによって初期化されます。

ServiceObjectVersion プロパティ

形式　LONG ServiceObjectVersion;

説明　サービスオブジェクトバージョン番号

このプロパティはサービスオブジェクトバージョン番号を保持します。次の3つのバージョンレベルがあります。

バージョンレベル	説明
メジャー	百万の位です。 デバイスクラスに対してOPOSメジャーバージョンレベルを変更すると、かなりのインタフェース拡張が行われ、古くなったインタフェースのサポートを前メジャーバージョンレベルから削除します。
マイナ	千から十万までの位です。 デバイスクラスに対してOPOSマイナバージョンレベルを変更すると、多少のインタフェース拡張が行われ、このメジャーバージョンレベルでの前インタフェースのスーパーセットを提供しなければなりません。
ビルド	一から百までの位.です。 サービスオブジェクトディベロッパが設定する内部レベル。SOインプリメントに修正を行うと更新されます。

バージョン番号の例を次に示します。

1002038

この値はバージョン"1.2.38"として表示され、サービスオブジェクトのメジャーバージョンが1、マイナバージョンが2、ビルドが38であることを意味します。

このプロパティは**Open**メソッドによって初期化されます。

備考
デバイスクラスのサービスオブジェクトは、メジャーバージョン番号がコントロールオブジェクトのメジャーバージョン番号と一致する限り、そのクラスのコントロールオブジェクトとともに動作します[3]。メジャーバージョン番号が一致するけれども、サービスオブジェクトのマイナーバージョン番号がコントロールオブジェクトのマイナーバージョン番号より大きい場合は、サービスオブジェクトはコントロールオブジェクトの該リリース版では、アクセスできない、いくつかの新しいメソッドやプロパティをサポートしている場合があります。

アプリケーションがそれらの機能を使用するためには、サービスオブジェクトに対応するコントロールオブジェクトに更新する必要があります。

State プロパティ

形式　LONG State;

説明　コントロールの現在の状態を示します。

値	意味
OPOS_S_CLOSED	コントロールはクローズしています。
OPOS_S_IDLE	コントロールは正常な状態にあり、ビジーではありません。
OPOS_S_BUSY	コントロールは正常な状態にあり、出力を実行しているためビジーです。
OPOS_S_ERROR	エラーが報告され、通常のI/Oが再開できるには、その前にアプリケーションがコントロールを正常な状態に戻さなければなりません。

このプロパティはいつでも読取り可能です。

参照　**「ステータス、結果コード、状態モデル」**

[3] **注意**：日本版仕様の第1.0版とその他の版では、インクルードファイルのシンボル定義値の変更等があり、同一メジャー番号の場合でも動作しないデバイスクラスが存在します。

メソッド

CheckHealth メソッド

形式 LONG CheckHealth (LONG *Level*);

*Level*パラメータは、デバイスで実行するヘルスチェックのタイプを示します。以下の値を指定できます。

値	意味
OPOS_CH_INTERNAL	デバイスを物理的に使用しないヘルスチェックを実行します。デバイスを内部テストで可能なだけテストします。
OPOS_CH_EXTERNAL	デバイスを使用して完全なテストを実行します。たとえば、パターンをプリンタに印刷できます。
OPOS_CH_INTERACTIVE	デバイスとの対話テストを実行します。サポートするサービスオブジェクトは、通常、モーダルダイアログボックスを表示して、テストオプションと結果を示します。

説明 デバイスの状態をテストするときに呼び出します。

このメソッドの結果は**CheckHealthText**プロパティに格納されます。

CheckHealthメソッドは常に同期です。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	ヘルスチェックプロシージャが適切に開始されたことを示し、確認できた場合、デバイスが正常であることを示します。しかし、多くのデバイスでは、正常かどうかはテスト結果を見ないと決定できません。
OPOS_E_ILLEGAL	指定されたヘルスチェックレベルをサービスオブジェクトがサポートしていません。
OPOS_E_BUSY	出力が進行中の間、実行できません。
その他	**ResultCode**の項目を参照してください。

参照 「**OpenPOSコントロール概要**」、**CheckHealthTextプロパティ**

ClaimDevice メソッド　　*第1.5版にて名称変更*

形式　　LONG ClaimDevice (LONG *Timeout*);

*Timeout*パラメータは、排他アクセス権を獲得するまでの最大待ち時間（ミリ秒単位）を示します。ゼロの場合、メソッドはデバイスの排他アクセスが獲得できなかった場合でも、その結果を直ちに戻します。
OPOS_FOREVER(-1)が設定されている場合は、メソッドは排他アクセス権が獲得できるまで必要なだけ待ちます。

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。多くのデバイスは、まずアプリケーションが排他アクセス権を獲得して、それからデバイスを使用しなければなりません。

成功すると、**Claimed**プロパティはTRUEに設定されます。

第 1.0 – 1.4 版

第1.5版より前では、このメソッド名称は**Claim**です。

第 1.5 版以降

アーリーバウンドアプリケーションは**ClaimDevice**を使用しなくてはなりません。レイトバウンドアプリケーションとの互換性のため、コントロールオブジェクトのIDispatchインタフェースは**ClaimDevice**と**Claim**の両者をサポートします。1.5版対応のアプリケーションは**Claim**ではなく**ClaimDevice**を使用することを推奨します。

アーリーバウンドアプリケーションは、アプリケーション開発時に、クラスID、インタフェースID、メソッド、プロパティ、イベントをコールするための詳細を読み取ることを要求します。これにより、メソッドやプロパティの呼び出し、およびイベント受理をスタティックな手順で構築できます。Microsoft Visual C++、Visual Basic、さらにほとんどのコンパイル言語は、アーリーバインディングをサポートします。

レイトバウンドアプリケーションは実行時に詳細を読み込むことを要求します。メソッドやプロパティの呼び出しに、イベント受理を加えた実行手順をダイナミックに構築します。レイトバインディングは多くのコンパイル言語で実装されていますが、プログラマーに付加的な作業、特にイベント受理処理を要求することがあります。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味

OPOS_SUCCESS	排他アクセスが認められました。**Claimed**プロパティはTRUEになっています。このアプリケーションがすでにデバイスを排他アクセスしていた場合も戻されます。
OPOS_E_ILLEGAL	このデバイスを排他アクセス用に占有することはできません。または、無効な*Timeout*パラメータが指定されています。
OPOS_E_TIMEOUT	別のアプリケーションがデバイスに排他アクセスしていて、解放されるのを待っていたが*Timeout*時間（ミリ秒単位）が過ぎました。

参照　「**デバイス共有モデルについて**」、**ReleaseDevice**メソッド

ClearInput メソッド

形式　LONG ClearInput ();

説明　バッファリ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。

保留になっていたデータイベントや入力エラーイベント（ **DataEventEnabled**がTRUEに、**FreezeEvents**をFALSEに設定されるのを待ち合わせている。）はクリアされます。

戻り値　次の値が戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	入力がクリアされました。
OPOS_E_CLAIMED	デバイスは他のプロセスにより排他アクセスされています。
OPOS_E_NOTCLAIMED	本メソッドの実行前にデバイスの排他アクセス権を獲得する必要があります。

参照　「**入力モデルについて**」

ClearInputProperties メソッド　　*第1.10版にて追加*

形式　　LONG ClearInputProperties();

説明　　DataEventやErrorEventの結果通知された全てのデータプロパティは、ClearInputPropertiesメソッドを呼び出せば初期値に戻ります。

この呼び出しでは、DataCountとStateプロパティはリセットしません。

戻り値　　次の値が戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	入力がクリアされました。
OPOS_E_CLAIMED	デバイスは他のプロセスにより排他アクセスされています。
OPOS_E_NOTCLAIMED	本メソッドの実行前にデバイスの排他アクセス権を獲得する必要があります。

参照　　「**入力モデルについて**」

ClearOutput メソッド　　第1.7版にて更新

形式　LONG ClearOutput ();

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。

保留になっていた出力エラーイベント（**FreezeEvents**がFALSEに設定されるのを待ち合わせている。）もクリアされます。

戻り値　次の値が戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	出力がクリアされました。
OPOS_E_CLAIMED	デバイスは他のプロセスにより排他アクセスされています。
OPOS_E_NOTCLAIMED	本メソッドの実行前にデバイスの排他アクセス権を獲得する必要があります。

参照　「**出力モデルについて**」

Close メソッド

形式　LONG Close ();

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。

DeviceEnabledプロパティがTRUEの場合、まずデバイスがディセーブルにされます。

ClaimedプロパティがTRUEの場合、まずデバイスの排他アクセスが解除されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	デバイスがディセーブルにされ、クローズしました。
その他	**ResultCode**の項目を参照してください。

参照　「**OpenPOSコントロール概要**」、**Open**メソッド

CompareFirmWareVersion メソッド　　*第1.9版にて追加*

形式　LONG CompareFirmWareVersion (String *FirmWareFileName*, Long *Result*);

パラメータ	説明
FirmWareFileName	デバイスのファームウェアとバージョンを比較されるために、ファームウェアを含むファイル名か、ファームウェアフィルのセットを服務ファイル名を示します。
Result	比較結果がこのここで指定した領域の戻されます。

説明　このメソッドは、引数で示されるファイルに含まれるファームウェアのバージョンと物理デバイスに含まれるファームウェアのバージョンがより新しい、より古い、同じであるということを決定します。

サービスオブジェクトは、指定されたファームウェアのファイルが存在する事と、その内容が比較処理を実行することを試みる前に、デバイスにとって正しいことを確認すべきです。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
CFV_FIRMWARE_OLDER	ファームウェアファイルのひとつ以上のバージョンが、デバイスのファームウェアより古いことを示します。そして、ファームウェアファイルにはデバイスのファームウェアより新しい物がないことをあらわします。
CFV_FIRMWARE_SAME	ファームウェアフィレのすべてのファイルが、デバイスのファームウェアと同じであることを示します。
CFV_FIRMWARE_NEWER	ファームウェアファイルのひとつ以上のバージョンが、デバイスのファームウェアより新しいことを示します。そして、ファームウェアファイルにはデバイスのファームウェアより古い物がないことをあらわします。
CFV_FIRMWARE_DIFFERENT	

ファームウェアファイルのひとつ以上のバージョンが、デバイスのファームウェアより異なっていることを示します。しかし、年代の連携が確定できないか、連携が矛盾しているかのいずれかですーひとつ以上が古いか、新しいかです。

CFV_FIRMWARE_UNKNOWN

二つのファームウェアのバージョン間の繋がりが確定できなません。この戻り値のために可能な結果は、日本語と米国のファームウェアのバージョンを比較を試みることでできます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_E_ILLEGAL	**CapComperFirmwareVersion**がfalse
OPOS_E_NOEXIST	*FirmwareFileName*によって示されるファイルが存在しないか、*FirmwareFileName*がファイルリストを示す場合、構成するファームウェアファイルのひとつ以上が誤っています。
OPOS_E_EXTENDED	*ErrorCodeExtended*=EFIRMWARE_BAD_FILE：指定されたファームウェアファイルがありません。しかし、ひとつ以上が正しくないフォーマットか壊れています。

参照　**CapCompareFirmwareVersionプロパティ**

DirectIO メソッド

形式 LONG DirectIO (LONG *Command*, LONG* *pData*, BSTR* *pString*);

パラメータ	説明
Command	コマンド番号。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。値はコマンド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。値はコマンド番号とサービスオブジェクトによって変化します。このデータのフォーマットは**BinaryConversion**プロパティの値に依存します。詳細は**BinaryConversion**プロパティを参照してください。

説明 サービスオブジェクトと直接通信するときに呼び出します。

このメソッドにより、標準コントロールオブジェクトではサポートしていないデバイスクラスの機能を、サービスオブジェクトがアプリケーションに提供できるようになります。サービスオブジェクトによるコマンドの定義に応じて、このメソッドは同期にも非同期にもなります。

DirectIOを使用するとアプリケーションの移植性がなくなります。しかし、アプリケーションは、条件コード内で**DirectIO**を実行することによって、移植性を維持できます。条件コードは、**ServiceObjectDescription**、**DeviceDesciription**、または**DeviceName**プロパティの値に基づきます。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	**DirectIO**が成功しました。
その他	**ResultCode**の項目を参照してください。

参照 **DirectIOEvent**イベント

Open メソッド

形式　　LONG Open (BSTR *DeviceName*);

*DeviceName*パラメータは、オープンするデバイス名を指定します。

説明　　デバイスをオープンするときに呼び出します。

デバイス名は、このコントロールオブジェクトがサポートするデバイスのどれを使用するかを指定します。*DeviceName*はこのデバイスクラスのシステムレジストリに存在しなければなりません。デバイス名と物理デバイスの関係は、オペレーティングシステムレジストリ内のエントリで決まります。これらのエントリはベンダが提供するセットアップユーティリティかコンフィギュレーションユーティリティが更新します。（付録A「**OPOSレジストリの利用法**」を参照してください。）

Openメソッドが成功すると、**Claimed**、**DeviceEnabled**、**DataEventEnabled**、**FreezeEvents**のプロパティと、OPOSコピーライト記述およびバージョン番号が設定されます。その他のクラス固有プロパティも初期化されます。

第 1.5 版以降

OpenResultプロパティの値は**Open**メソッドで設定されます。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	オープンに成功しました。
OPOS_E_ILLEGAL	該当のコントロールはすでにオープンしています。
OPOS_E_NOEXIST	指定された*DeviceName*が見つかりません。
OPOS_E_NOSERVICE	対応するサービスオブジェクトに接続できません。
その他	**ResultCode**の項目を参照してください。

備考
Openメソッド実行後の**ResultCode**プロパティの値は、次の２つのケースでは、**Open**メソッドの戻り値と異なります。

コントロールがクローズしており**Open**メソッドが失敗した時：
ResultCodeプロパティは、OPOS_E_CLOSEDのままとなる。

コントロールが既にオープンされている時： **Open**メソッドの戻り値は、OPOS_E_ILLEGALとなりますが、 **ResultCode**プロパティ値は、 **Open**メソッド実行前の値を保持しています。

参照　「**OpenPOSコントロール概要**」、Closeメソッド

ReleaseDevice メソッド　　*第1.5版にて名称変更*

形式　　LONG ReleaseDevice ();

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。

DeviceEnabledプロパティがTRUEで排他デバイスならば、デバイスはディセーブルにされます。（共有可能デバイスの場合、ReleaseDeviceはデバイスのイネーブル状態に影響を与えません。）

第 1.0 – 1.4 版

第1.5版より前では、このメソッド名称はReleaseです。

第 1.5 版以降

アーリーバウンドアプリケーションはReleaseDeviceを使用しなくてはなりません。レイトバウンドアプリケーションとの互換性のため、コントロールオブジェクトのIDispatchインタフェースはReleaseDeviceとReleaseの両者をサポートします。1.5版対応のアプリケーションはReleaseではなくReleaseDeviceを使用することを推奨します。

アーリーバウンドアプリケーションは、アプリケーション開発時に、クラスID、インタフェースID、メソッド、プロパティ、イベントをコールするための詳細を読み取ることを要求します。これにより、メソッドやプロパティの呼び出し、およびイベント受理をスタティックな手順で構築できます。Microsoft Visual C++、Visual Basic、さらにほとんどのコンパイル言語は、アーリーバインディングをサポートします。

レイトバウンドアプリケーションは実行時に詳細を読み込むことを要求します。メソッドやプロパティの呼び出しに、イベント受理を加えた実行手順をダイナミックに構築します。レイトバインディングは多くのコンパイル言語で実装されていますが、プログラマーに付加的な作業、特にイベント受理処理を要求することがあります。

戻り値　　次の値のいずれかが戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	排他アクセスが解除されました。Claimedプロパティは FALSEになります。
OPOS_E_ILLEGAL	アプリケーションは該当のデバイスへの排他アクセス権を持っていません。

参照　　「デバイス共有モデルについて」、ClaimDeviceメソッド

ResetStatistics メソッド　　*第1.8版にて追加*

形式　LONG ResetStatistics (BSTR *StatisticsBuffer*);

パラメータ	意味
StatisticsBuffer	リセットするスタティスティクスを定義したデータバッファです。

これは名前のカンマ区切りリストです。空文字列の時はすべてのリセット可能なスタティスティクスをリセットします。”U_” はUnifiedPOSで定義されたすべてのリセット可能なスタティスティクスをリセットします。”M_” はベンダ固有のすべてのリセット可能なスタティスティクスをリセットします。そして ”名前1,名前2”(XMLファイルに定義されているもの)の時は、指定されたリセット可能なスタティスティクスをリセットします。

説明　デバイスの中で定義されたリセット可能なスタティスティクスをリセットします。

指定されたすべてのスタティスティクスのリセットに成功した場合のみこのメソッドが成功します。それ以外ではOPOS_E_EXTENDEDが戻ります。

このメソッドを使うには**CapStatisticsReporting**プロパティと**CapUpdateStatistics**プロパティが両方ともTRUEでなければなりません。

このメソッドは常に同期で実行されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	すべての指定されたスタティスティクスのリセットに成功しました。
OPOS_E_ILLEGAL	**CapStatisticsReporting**プロパティと**CapUpdateStatistics**プロパティのいずれかがFALSEです。または指定された名前のスタティスティクスのすべてが未定義であるかリセット可能ではありません。
OPOS_E_EXTENDED	*ResultCodeExtended*=OPOS_ESTATS_ERROR 指定されたスタティスティクスの１つ以上がリセットできませんでした。

その他	**ResultCode**の項目を参照してください。

参照 「デバイスインフォメーションレポーティングモデルについて」、**CapStatisticsRepoting**プロパティ、**CapUpdateStatistics**プロパティ

RetrieveStatistics メソッド *第1.10版にて更新*

形式 LONG RetrieveStatistics (BSTR **pStatisticsBuffer*);

パラメータ	意味
pStatisticsBuffer	取得するスタティスティクスを定義したデータバッファであり、かつ通知されるスタティスティクスを格納する場所でもあります。

これは名前のカンマ区切りリストです。空文字列の時はすべてのスタティスティクスを取得します。”U_” はUnifiedPOSで定義されたすべてのスタティスティクスを取得します。”M_” はベンダ固有のすべてのスタティスティクスを取得します。そして ”名前1,名前2”(XMLファイルに定義されているもの)の時は、指定されたスタティスティクスを取得します。

説明 デバイスから指定されたスタティスティクスを取得します。

このメソッドを使うには**CapStatisticsReporting**プロパティがTRUEでなければなりません。

このメソッドは常に同期で実行されます。

すべての **RetrieveStatistics** 呼び出しは少なくとも以下のXMLを戻します

```
<?xml version='1.0' ?>
<UPOSStat version="1.8.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nrf-arts.org/IXRetail/namespace/"
xsi:schemaLocation="http://www.nrf-
arts.org/IXRetail/namespace/UPOSStat.xsd">
  <Event>
    <Parameter>
      <Name>RequestedStatistic</Name>
      <Value>StatisticValue</Value>
    </Parameter>
  </Event>
  <Equipment>
    <UnifiedPOSVersion>Version</UnifiedPOSVersion>
    <DeviceCategory UPOS="DeviceCategoryName"/>
    <ManufacturerName>Device
Manufacturer</ManufacturerName>
    <ModelName>Device Model Name</ModelName>
```

```
  </Equipment>
</UPOSStat>
```

デバイスがサポートしていないスタティスティクス名をアプリケーションが指定したら、<Parameter>エントリは<Value>を空で戻します。例えば、

```
<Parameter>
  <Name>RequestedStatistic</Name>
  <Value></Value>
</Parameter>
```

デバイスが収集しているベンダ固有の（スキーマに定義されていない）すべてのスタティスティクスは、<Parameter>タグではなく<ManufacturerSpecific>タグの中に戻されます。

```
<ManufacturerSpecific>
  <Name>TheAnswer</Name>
  <Value>42</Value>
</ManufacturerSpecific>
```

アプリケーションがデバイスのすべてのスタティスティクスを指定した場合、<UPOSStat>タグのバージョン属性で指定された版数のXMLスキーマに定義されたデバイスカテゴリに定義されたすべてのスタティスティクスを<Parameter>エントリに戻します。デバイスがスタティスティクスのどれかを記録していない場合、対応する<Value>タグは空になります。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	指定されたスタティスティクスの取得に成功しました。
OPOS_E_ILLEGAL	**CapStatisticsReporting**プロパティがFALSEです。または指定された名前のスタティスティクスのすべてが未定義です。
OPOS_E_EXTENDED	*ResultCodeExtended*=OPOS_ESTATS_ERROR 指定されたスタティスティクスを変更するにはさらに一つ以上のスタティスティクスを追加して変更する必要があります。
その他	**ResultCode**の項目を参照してください。

参照　**「デバイスインフォメーションレポーティングモデルについて」、CapStatisticsRepotingプロパティ**

UpdateFirmware メソッド　　*第1.9版にて追加*

形式　LONG UpdateFirmware (String *FirmWareFileName*);

パラメータ	説明
FirmWareFileName	デバイスにダウンロードされるファームウェアを含むファイルか、ファームウェアファイルのセットを含むファイルの名称を指定します。

説明

このメソッドは、ファームウェアのバージョンがすでにデバイスにあるファームウェアのバージョンと同じか古いか新しいかに関わらず、FirmWareFileNameパラメータで示されるファイルに含まれる、または定義されるファームウェアのバージョンに、デバイスのファームウェアを更新します。FirmWareFileNameがファイルリストを示しているならば、ファームウェアに関する全てのファイルは、ファームウェアのリストファイルと同じディレクトリに置かれるべきです。これにより、ファームウェアのリストファイルを変更せずに、更新されたファームウェアの配布が可能となります。

このメソッドが起動されると、サービスオブジェクトは、指定されたファームウェアが存在し、内容がデバイスに適していることを確認します。正常であればこのメソッドは直ちに終了し、ファームウェア更新処理の残りは非同期に継続されます。サービスオブジェクトは、SUE_UF_PROGESSに1から100を加えたファームウェア更新経過のパーセント値で**StatusUpdateEvent**を通知し、ファームウェア更新処理の状態をアプリケーションに報告します。アプリケーションが使いやすいように、**StatusUpdateEvent**のSUE_UF_COMPLETEの値はSUE_UF_PEOGRESS+100と同じ値に定義されています。

一貫性のために、ファームウェア更新処理の終了は、新しいファームウェアが物理デバイスにダウンロードされ、必要な物理デバイスの初期化が完了し、サービスオブジェクトと物理デバイスがファームウェア更新の開始以前の状態に戻ったときとします。

一貫性のために、サービスオブジェクトは、デバイスがファームウェア更新の経過を物理的に通知できないとしても、途中の終了パーセント（つまり1から99パーセント）で最低一回は**StatusUpdateEvent**を通知しなければなりません。ファームウェア更新処理が正常に終了したら、サービスオブジェクトは、100パーセントかそれと同等の値のSUE_UF_COMPLETEで**StatusUpdateEvent**を通知しなければなりません。これらのサービスオブジェクトへの要求事項により、アプリケーションは進捗状況を常に通知できる設計手段を利用できるようになります。

ファームウェア更新を非同期に実行している途中でエラーが検出されれば、以下の**StatusUpdateEvent**が通知されます。

値	意味
SUE_UF_FAILD_DEV_OK	ファームウェア更新処理が失敗しましたが、デバイスはまだ操作可能です。
SUE_UF_FAILD_DEV_UNRECOVERABLE	ファームウェア更新処理が失敗し、デバイスは使用することもソフト的に修復することもできません。デバイスは操作できる状態に戻すための保守が必要です。
SUE_UF_FAILD_DEV_NEEDS_FIRMWARE	ファームウェア更新処理が失敗し、デバイスは他のファームウェア更新が成功するまで操作できません。
SUE_UF_FAILD_DEV_UNKNOWN	ファームウェア更新処理が失敗し、デバイスは不安定な状態です。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_E_ILLEGAL	**CapComperFirmwareVersion**はfalse。
OPOS_E_NOEXIST	*FirmwareFileName*により示されるファイルが存在しないか、*FirmwareFileName*がファイルリストを示しているが一つ以上のファームウェアファイルが欠落しています。
OPOS_E_EXTENDED	*ErrorCodeExtended*=EFIRMWARE_BAD_FILE：ファームウェアファイルを指定していますが、一つ以上のファイルが不正フォーマットであるか壊れています。

参照　**CapUpdateFirmware**プロパティ

UpdateStatistics メソッド　　*第1.10版にて更新*

形式　LONG UpdateStatistics (BSTR *StatisticsBuffer*);

パラメータ	意味
StatisticsBuffer	変更するスタティスティクスを定義したデータバッファです。

これは名前のカンマ区切りリストです。空文字列の名前(“”=”値1”)はすべてのリセット可能なスタティスティクスを”値1”に変更します。”U_=値2” はUnifiedPOSで定義されたすべてのリセット可能なスタティスティクスを”値2”に変更します。”M_=値3” はベンダ固有のすべてのリセット可能なスタティスティクスを”値3”に変更します。そして ”名前1=値4,名前2=値5”(XMLファイルに定義されているもの)の時は、指定されたリセット可能なスタティスティクスを指定された値に変更します。

説明　デバイスの中で定義されたリセット可能なスタティスティクスを変更します。

指定されたすべてのスタティスティクスの変更に成功した場合のみこのメソッドが成功します。それ以外ではOPOS_E_EXTENDEDが戻ります。

このメソッドを使うには**CapStatisticsReporting**プロパティと**CapUpdateStatistics**プロパティが両方ともTRUEでなければなりません。

このメソッドは常に同期で実行されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	すべての指定されたスタティスティクスの変更に成功しました。
OPOS_E_ILLEGAL	**CapStatisticsReporting**プロパティと**CapUpdateStatistics**プロパティのいずれかがFALSEです。または指定された名前のスタティスティクスのすべてが未定義であるかリセット可能ではありません。

OPOS_E_EXTENDED	*ResultCodeExtended*=OPOS_ESTATS_ERROR 指定されたスタティスティクスを変更するにはさらに一つ以上のスタティスティクスを追加して変更する必要があります。
その他	**ResultCode**の項目を参照してください。

参照　「デバイスインフォメーションレポーティングモデルについて」、**CapStatisticsRepoting**プロパティ、**CapUpdateStatistics**プロパティ

イベント

DataEvent イベント

形式　　void DataEvent (LONG *Status*);

*Status*パラメータには入力ステータスが入ります。そ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　　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示すために通知されます。**DataEventEnabled**プロパティはFALSEとなり、アプリケーションがこのプロパティをTRUEに戻すまで、データイベントはそれ以上発生しません。実際の入力データはデバイス固有プロパティに格納されます。

データ受信時に**DataEventEnabled**がFALSEであった場合、データは内部OPOSバッファで待ち合わされ、デバイス固有入力データプロパティは更新されず、イベントも通知されません。（このプロパティがTRUEに戻されたときに、待ち合わされている入力データがあり、**FreezeEvents**プロパティがFALSEであれば、直ちにイベントの通知が行われます。）

参照　　「**入力モデルについて**」、**DataEventEnabled**プロパティ、**FreezeEvents**プロパティ

DirectIOEvent イベント

形式　void DirectIOEvent (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。イベント番号とサービスオブジェクトで変化する特定値です。
pString	文字列のポインタ。イベント番号とサービスオブジェクトで変化する特定値です。このデータのフォーマットはBinaryConversionプロパティの値に依存します。詳細はBinaryConversionプロパティを参照してください。

説明　アプリケーションと直接通信するために、サービスオブジェクトが通知します。

このイベントにより、コントロールオブジェクトではサポートされないイベントを、サービスオブジェクトがアプリケーションに提供することが可能となります。

参照　DirectIOメソッド

ErrorEvent イベント　　第1.13版にて更新

形式　void ErrorEvent (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**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**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　データ送信を再試行します。エラー状態は解除されます。いくつかの入力デバイスで*ErrorLocus*がOPOS_EL_INPUTの場合、エラー状態を解除し、再読み込みを行うときに使用します。非同期出力デバイスで*ErrorLocus*がOPOS_EL_OUTPUTの場合、エラー状態が解除され、再送信を行うときに使用します。*ErrorLocus*がOPOS_EL_OUTPUTの場合のデフォルトです。

OPOS_ER_CLEAR　*ErrorLocus*がOPOS_EL_OUTPUT、OPOS_EL_INPUT、OPOS_EL_INPUT_DATAの場合に使用します。全てのバッファリングされている入力または、出力データ（非同期出力を含む。）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応答です。

OPOS_ER_CONTINUEINPUT　*ErrorLocus*がOPOS_EL_INPUT_DATAの場合だけ使用します。データエラーを容認し、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応答です。

説明　エラーが検出され、コントロールの**State**がエラー状態に遷移したときに通知します。

正しいアプリケーションシーケンシングが行われるよう、**DataEventEnabled**プロパティがTRUEになるまで、入力エラーイベントは通知されません。

参照　**「ステータス、結果コード、状態モデル」**

OutputCompleteEvent イベント　　第1.13版にて更新

形式　　void OutputCompleteEvent (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　　前に開始された非同期出力リクエストが正常に終了したときに通知します。

参照　　「出力モデルについて、OutputIDプロパティ」

StatusUpdateEvent イベント　　第1.9版にて更新

形式　　void StatusUpdateEvent (LONG *Status*);

*Status*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　　コントロールがデバイスステータス変化の警報をアプリケーションに出す必要がある場合に通知します。

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。

デバイスがイネーブルとなったとき、アプリケーションにデバイスの状態を知らせるために、コントロールは最初のStatusUpdateEventを発行する場合もあります。しかし、この動作（イネーブル直後のイベント通知）は必須ではありません。

第1.3版以降　――　電源状態通知機能

PowerNotify = OPOS_PN_ENABLEDの場合、すべてのデバイスクラスは下記のいずれかの *Status*パラメータ値とともにStatusUpdateEventを発行します。

値	意味
OPOS_SUE_POWER_ONLINE	デバイスは電源オンでかつレディ状態です。CapPowerReporting = OPOS_PR_STANDARD、または、OPOS_PR_ADVANCED の場合、この値が返されます。
OPOS_SUE_POWER_OFF	デバイスは電源オフ、または本体に接続されていません。CapPowerReporting = OPOS_PR_ADVANCEDの場合のみ、この値が返されます。
OPOS_SUE_POWER_OFFLINE	デバイスは電源オンですが、ノットレディ状態か要求に対する応答が不可能な状態です。CapPowerReporting = OPOS_PR_ADVANCEDの場合のみ、この値が返されます。

OPOS_SUE_POWER_OFF_OFFLINE
デバイスは電源オフ、またはオフライン状態です。
CapPowerReporting = OPOS_PR_STANDARDの場合のみ、この値が返されます。

共通プロパティ**PowerState**にも、デバイスの現在の電源状態を保持されます。

第1.9版以降 —— ファームウェア更新通知

ファームウェア更新の機能は、第1.9版で追加され、非同期ファームウェア更新処理の状態と進行を通知する以下のStatusc値が追加されました。

値	意味

SUE_UF_PROGRESS + 1 to 100
ファームウェア更新処理は、全ての処理の1から100%を正常に完了しました。

SUE_UF_COMPLETE
ファームウェア更新処理が正常に完了しました。この定数の値はSUE_UF_PROGRESS+100と同じです。

SUE_UF_COMPLETE_DEV_NOT_RESTORED
ファームウェア更新処理は成功しましたが、サービスオブジェクトか物理デバイスがファームウェア更新前の状態に戻れません。
サービスオブジェクトと物理デバイスの状態が正常であるかを確認するため、アプリケーションはサービスオブジェクトを**Close**、**Open**、**Claim**して再度イネーブルし、さらにアプリケーションの固有設定を全て保存する必要があります。

SUE_UF_FAILD_DEV_OK
ファームウェア更新処理が失敗しましたが、デバイスはまだ操作可能です。

SUE_UF_FAILD_DEV_UNRECOVERABLE
ファームウェア更新処理が失敗し、デバイスは使用することもソフト的に修復することもできません。デバイスは操作できる状態に戻すための保守が必要です。

SUE_UF_FAILD_DEV_NEEDS_FIRMWARE
ファームウェア更新処理が失敗し、デバイスは他のファームウェア更新が成功するまで操作できません。

SUE_UF_FAILD_DEV_UNKNOWN
ファームウェア更新処理が失敗し、デバイスは不安定な状態です。

説明　このイベントは、デバイスがデバイス状態の変化をアプリケーションに警告する必要があるときに通知されます。例として、キャッシュドロワ位置（開と閉）の変化や、POSプリンタセンサ（伝票の有り無し）の変化があります。

デバイスがイネーブルされると、コントロールはデバイス状態をアプリケーションに報告するためにこのイベントを通知します。しかしながら、この動作は必須ではありません。

参照　**「ステータス、結果コード、状態モデル」；「デバイス電源通知モデルについて」；CapPowerReportingプロパティ、PowerNotifyプロパティ**

第 2 章 （ .NET 版 ）
共通プロパティ、メソッド、イベント

プロパティ

AutoDisable プロパティ

型 **bool**

説明 trueの場合、サービスは、データを受信し、**DataEvent**としてキューイングした後で、**DeviceEnabled**をfalseに設定します。次の入力を受信できるようにするためには、アプリケーションは**DeviceEnabled**をtrueに設定しなければなりません。

falseの場合、サービスは、データ受信時にデバイスを自動的にはディセーブルにしません。

このプロパティは、入力データの受信を制御するための追加オプションをアプリケーションに提供します。アプリケーションが1つの入力のみを受信し処理したい場合や、一度に1つの入力のみを受信したい場合、このプロパティをtrueに設定します。このプロパティは、イベント駆動入力デバイスに適用されます。

このプロパティは**Open**メソッドによりfalseに初期化されます。

エラー このプロパティにアクセスしたときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

CapPowerReporting プロパティ

型　PowerReporting

説明　デバイスの通知能力を識別します。可能な値は、PowerReporting列挙体によって定義されています。

サービスオブジェクトは、デバイスの通知能力に基づいてPowerReportingを設定します。

電源通知能力を示す値は、下記のとおりです。

値	意味
None	SOはデバイスの電源状態の判断ができません。よって、電源通知機能は機能しません。
Standard	SOは、OffOffLine（電源オフまたはオフライン）とOnlineの2種類の電源状態を判断し通知が可能です。
Advanced	SOは、Off、OffLine、OnLineの3種類すべての電源状態を判断し通知が可能です。

エラー　なし

CapStatisticsReporting プロパティ

型　bool

説明　trueの場合、デバイスは使用状態に関する様々なスタティスティクスを収集提供可能です。収集される情報はデバイスに固有であり、RetrieveStatistic(s)メソッドを使用して取得できます。

エラー　このプロパティにアクセスしたときに、PosControlExceptionがスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

CapUpdateStatistics プロパティ

型 **bool**

説明 trueの場合、デバイスのスタティスティクスの一部またはすべてが、ResetStatistic(s)メソッドを使用して0にリセットしたり、UpdateStatistic(s)メソッドを使用して変更したりできます。

CapStatisticsRepotingプロパティがfalseの場合、CapUpdateStatisticsプロパティは常にfalseです。

エラー このプロパティにアクセスしたときに、PosControlExceptionがスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

CheckHealthText プロパティ

型 **string**

説明 デバイスのヘルスを表す文字列を格納します。アプリケーションがCheckHealthメソッドを呼び出すと、サービスオブジェクトによって更新されます。

エラー このプロパティにアクセスしたときに、PosControlExceptionがスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

Claimed プロパティ

型 **bool**

説明 trueの場合、デバイスの排他アクセス権が獲得されています。falseの場合、デバイスは、ほかのアプリケーションと共有できるよう解放されています。

排他使用デバイスは、サービスオブジェクトがそのメソッドやプロパティへのアクセスを有効にする前に、そしてサービスオブジェクトがイベントをアプリケーションに通知する前に、Claimメソッドを呼び出して排他アクセス権を取得する必要があります。

エラー このプロパティにアクセスしたときに、PosControlExceptionがスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

DataCount プロパティ

型　　**int**

説明　　キューイングされている**DataEvent**の数を保持します。

アプリケーションは、このプロパティを参照することで、デバイスからキューイングされているにもかかわらず、アプリケーションが他の処理中であったり、イベントが凍結しているなどの原因でまだ通知されていない入力があるかどうかを確認できます。

このプロパティは**Open**メソッドにより0に初期化されます。

エラー　　このプロパティにアクセスしたときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

DataEventEnabled プロパティ

型　　**bool**

説明　　trueの場合、入力データがキューイングされるとすぐに**DataEvent**が通知されます。trueに変更され、入力データがすでにキューイングされていれば、**DataEvent**が即座に通知されます。（ただし、「即座」の通知を遅らせる他の条件もあることに注意が必要です。もし、**FreezeEvents**がtrueか、またはアプリケーションで既に他のイベントを処理中の場合、**DataEvent**は状況が改善するまでサービス内にキューイングされたままとなります。）

falseの場合、後でアプリケーションに通知できるよう、入力データはキューイングされます。また、入力エラーが発生しても、このプロパティがfalseの間、**ErrorEvent**は通知されません。

このプロパティは**Open**メソッドによりfalseに初期化されます。

エラー　　このプロパティにアクセスしたときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

DeviceDescription プロパティ

型 **string**

説明 デバイスを識別する文字列、およびデバイスに関連する情報を格納します。例えば、以下の文字列です。

“NCR 7192-0184 Printer, Japanese Version”

このプロパティは、アプリケーションがOpenメソッドを呼び出すと初期化されます。

エラー なし

DeviceEnabled プロパティ

型 **bool**

説明 trueの場合、デバイスはイネーブルにされています（動作状態です）。trueに変更すると、デバイスはイネーブルにされます。

falseの場合、デバイスはディセーブルにされています。falseに変更すると、可能ならば物理的にディセーブルにされます。後続の入力は廃棄され、出力動作を行うことはできません。

DeviceEnabledを変更しても、通常、物理的には出力デバイスに影響しません。しかし、整合性を保つため、アプリケーションは、出力デバイスを使用する前に、**DeviceEnabled**をtrueに設定しなければなりません。

エラー このプロパティにアクセスしたときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

DeviceName プロパティ

型　**string**

説明　デバイスを識別する短い文字列、およびデバイスに関連する情報を格納します。

これは、**DeviceDescription**の短縮バージョンで、30文字までに制限されています。

DeviceNameの典型的な使用例は、アプリケーションのメッセージボックスでデバイスを識別する際に、完全な記述では長すぎるときです。**DeviceName**文字列の例は、以下のものです。

“NCR 7192 Printer, Japanese”

エラー　なし

FreezeEvents プロパティ

型　**bool**

説明　trueの場合、アプリケーションは、サービスオブジェクトがイベントを通知しないようにリクエストしています。アプリケーションが**FreezeEvents**をfalseに変更するまで、サービスオブジェクトは、イベントをキューイングしますが通知しません。

falseの場合、アプリケーションは、イベントの通知を許可しています。凍結中に保持されていたイベントがあり、イベントを通知するすべての条件が調った場合、**FreezeEvents**をfalseに変更すると、イベントが通知されます。

アプリケーションは、イベントによる割り込みが起きて欲しくない一連のコードを実行する間、イベントを凍結することができます。

エラー　このプロパティにアクセスしたときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

OutputId プロパティ

型 **int**

説明 直前に開始した非同期出力の識別子を保持します。

サービスは、メソッドが非同期出力の開始に成功すると、リクエストに識別子を割り当てます。出力が完了すると、その出力IDをパラメータとして、**OutputCompleteEvent**をキューイングします。

出力ID番号は、サービスによって割り当てられ、未処理の非同期出力の中でユニークであることが保証されます。IDについてその他のことは規定されていません。

エラー このプロパティにアクセスしたときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

PowerNotify プロパティ

型 **PowerNotification**

説明 アプリケーションによって選択された電源通知機能のタイプを格納します。可能な値は、**PowerNotification**列挙体によって定義されています。

PowerNotifyは、デバイスがディセーブルの間、すなわち**DeviceEnabled**がfalseの間のみ設定が可能です。

エラー このプロパティにアクセスしたときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

PowerState プロパティ

型　PowerState

説明　現在の電源状態を格納します。可能な値は、PowerState列挙体によって定義されています。

PowerNotifyが有効で、DeviceEnabledがtrueの場合、PowerStateは、サービスオブジェクトが電源状態の変化を検知したときに更新されます。電源状態が変化すると、サービスオブジェクトは、PowerStateを更新し、StatusUpdateEventイベントをキューイングして、アプリケーションに通知します。

エラー　なし

ServiceObjectDescription プロパティ

型　string

説明　デバイスをサポートしているサービスオブジェクトと、そのデバイスを製造した会社を示す文字列を格納します。

ServiceObjectDescription文字列の例は、以下のものです。

“TM-T88IV Printer POS for .Net Service Driver, (C) 2005 Epson”

エラー　このプロパティにアクセスしたときに、PosControlExceptionがスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

ServiceObjectVersion プロパティ

型 System.version

説明 ServiceObjectVersionは、サービスオブジェクトのバージョン番号を保持します。バージョン番号は、メジャー、マイナ、ビルド、およびレビジョンの2～4つの整数値からなります。ビルドとレビジョンは省略できますが、レビジョンは、ビルドが指定されていないときのみ省略可能です。

メジャーおよびマイナバージョン番号は、サービスオブジェクトが実装するUnifiedPOSのバージョンに相当します。UnifiedPOS 1.8の仕様を実装するサービスオブジェクトは、メジャーに1を、マイナに8をセットします。ビルドおよびレビジョンバージョン番号は省略可能で、サービスオブジェクトが内部バージョンを確認するのに使用します。

エラー このプロパティにアクセスしたときに、PosControlExceptionがスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

State プロパティ

型 ControlState

説明 デバイスの現在の状態を格納します。可能な値は、ControlState列挙体によって定義されています。

Stateは、Openメソッドによって**ControlState.Idle**にセットされ、デバイスの状態にかかわらず、いつでも読み取り可能です。

エラー なし

SynchronizingObject プロパティ

型　System.ComponentModel.ISynchronizeInvoke

説明　**ISynchronizeInvoke**クラスのインスタンスを格納します。アプリケーションは、このプロパティを使用して、イベントが通知されるスレッドを指定することができます。**SynchronizingObject**がnullにセットされている場合、イベントは、サービスオブジェクトが所有する内部スレッドに通知されます。Windowsのフォームを使用するアプリケーションは、**SynchronizationObject**にメインフォームのFormクラスの**this**ポインタをセットして、Formクラスの要件どおり、イベントがメインのアプリケーションスレッドに通知されるようにします。

エラー　このプロパティにアクセスしたときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

メソッド

CheckHealth メソッド

形式 string CheckHealth (HealthCheckLevel *level*);

説明 アプリケーションは、デバイスの状態をテストするときに、CheckHealthを呼び出します。CheckHealthは常に同期的に実行されます。サービスオブジェクトは、ヘルスレベルを表す文字列を返し、CheckHealthTextプロパティを更新します。

*level*パラメータは、デバイスで実行するヘルスチェックのタイプを示します。可能な値は、HealthCheckLevel列挙体によって定義されています。

値	意味
Internal	デバイスを物理的に使用しないヘルスチェックを実行します。デバイスを、内部テストで可能な範囲で検査します。
External	デバイスを使用して完全なテストを実行します。例えば、パターンをプリンタに印刷できます。
Interactive	デバイスとの対話テストを実行します。サポートするサービスオブジェクトは、通常、モーダルダイアログボックスを表示して、テストのオプションと結果を示します。

エラー このメソッドを呼び出したときに、PosControlExceptionがスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

CheckHealthは、次のPosControlExceptionをスローすることがあります。

ErrorCodeの値	説明
Illegal	指定されたヘルスチェックレベルをサービスオブジェクトがサポートしていません。

Claim メソッド

形式　　void Claim (int *timeout*);

説明　　アプリケーションは、デバイスに対して排他アクセスを要求するときに、**Claim**を呼び出します。多くのデバイスは、まずアプリケーションが排他アクセス権を獲得して、それからデバイスを使用しなければなりません。。

*timeout*パラメータが0の場合、メソッドはデバイスの排他アクセスが獲得できなかった場合でも、その結果を直ちに戻します。*timeout*が**WaitForever** (-1) に設定されている場合は、**Claim**は排他アクセス権が獲得できるまで待ち続けます。

アプリケーションがデバイスの排他アクセス権を複数回要求しても、エラーは発生しません。**Claim**が成功すると、**Claimed**プロパティはtrueに設定されます。

*timeout*パラメータは、排他アクセス権を獲得するまでの最大待ち時間（ミリ秒単位）を示します。

エラー　　このメソッドを呼び出したときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

Claimは、次の**PosControlException**をスローすることがあります。

ErrorCodeの値	説明
Illegal	次のいずれかの状況が発生しています。 デバイスの排他アクセス権を獲得することができないか、 または、*timeout* パラメータに-1 未満の値が指定されています。
Timeout	別のアプリケーションがデバイスに排他アクセスしていて、解放されるのを待っていたが *timeout* 時間（ミリ秒単位）が過ぎました。

ClearInput メソッド

形式 void ClearInput ();

説明 バッファリングされているデバイス入力をすべてクリアします。

キューイングされている（通常**DataEventEnabled**がtrueに、**FreezeEvents**がfalseに設定されるのを待っている）データイベントや入力エラーイベントもクリアされます。

エラー このメソッドを呼び出したときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

ClearOutput メソッド

形式 void ClearOutput ();

説明 非同期出力をすべて含む、バッファリングされているすべての出力データを、すべてクリアします。また、可能ならば、処理中の出力も停止します。

キューイングされている（通常**FreezeEvents**がfalseに設定されるのを待っている）出力エラーイベントもクリアされます。

エラー このメソッドを呼び出したときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

Close メソッド

形式　　void Close ();

説明　　アプリケーションは、デバイスとそのリソースを解放するときに、**Close**を呼び出します。

DeviceEnabledプロパティがtrueの場合、デバイスはディセーブルにされます。**Claimed**プロパティがtrueの場合、デバイスの排他アクセスが解放されます。

エラー　　このメソッドを呼び出したときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

Closeは、次の**PosControlException**をスローすることがあります。

ErrorCodeの値	説明
Busy	**State** プロパティが **ControlState.Busy** に設定されています。これは、デバイスが使用中で、停止できないことを示します。
Closed	デバイスは既にクローズされています。

DirectIO メソッド

形式 DirectIOData DirectIO (int *command*, int *data*, object *obj*);

説明 アプリケーションは、サービスオブジェクトと直接通信するときに、DirectIOを呼び出します。

DirectIOを使用すると、標準のサービスインターフェイスではサポートしていないデバイスクラスの機能を、サービスオブジェクトがアプリケーションに提供できるようになります。サービスオブジェクトによるコマンドの定義に応じて、DirectIOは同期にも非同期にもなります。

エラー このメソッドを呼び出したときに、PosControlExceptionがスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

DirectIOは、DirectIODataの構造体のインスタンスを返します。

パラメータ	説明
command	コマンド番号。サービスオブジェクトが割り当てた特定の値です。
data	付加的な数値データ。値はコマンド番号とサービスオブジェクトによって変化します。
obj	サービスオブジェクトが提供する付加的なデータ。値はコマンド番号とサービスオブジェクトが何を送信するかによって変化します。

Open メソッド

形式 void Open ( );

説明 アプリケーションは、入出力処理のためにデバイスをオープンするときにOpenを呼び出します。Openは、DataEventEnabled、FreezeEvents、AutoDisable、Claimedなど、数々のプロパティの値を初期化します。

エラー このメソッドを呼び出したときに、PosControlExceptionがスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

Openは、次のPosControlExceptionをスローすることがあります。

ErrorCodeの値	説明
Illegal	デバイスは既にオープンされています。

Release メソッド

形式　　void Release ( );

説明　　アプリケーションは、デバイスの排他アクセスを解放するときに、**Release**を呼び出します。

DeviceEnabledプロパティがtrueで排他使用デバイスならば、デバイスは、まずディセーブルにされます。（共有可能デバイスの場合、**Release**はデバイスのイネーブル状態に影響を与えません。）**Release**が成功すると、**Claimed**プロパティはfalseに設定されます。

エラー　　このメソッドを呼び出したときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

Releaseは、次の**PosControlException**をスローすることがあります。

ErrorCodeの値	説明
Busy	デバイスは使用中です。
Illegal	次のいずれかの状況が発生しています。 アプリケーションが該当デバイスへの排他アクセス権を持っていないか、 または、デバイスが排他アクセス権を獲得されていません。

ResetStatistic メソッド (string)

形式 void ResetStatistic (string *statistic*);

説明 ***statistic***には、リセットするスタティスティクスを指定します。

アプリケーションは、指定したスタティスティクスを0にリセットするときに、**ResetStatistic**を呼び出します。**ResetStatistic**が成功するには、**CapStatisticsReporting**プロパティと**CapUpdateStatistics**プロパティが両方ともtrueでなければなりません。

ResetStatisticは常に同期で実行されます。

エラー このメソッドを呼び出したときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

ResetStatisticは、次の**PosControlException**をスローすることがあります。

ErrorCodeの値	説明
Illegal	次のいずれかの状況が発生しています。 **CapStatisticsRepoting** か **CapUpdateStatistics** のいずれかのプロパティが false に設定されている、 ***statistic*** パラメータが null、 または、指定されたスタティスティクスが存在しません。
Extended	**ExtendedErrorStatistics** です。指定されたスタティスティクスがリセットできません。

ResetStatistics メソッド ()

形式　　void ResetStatistics ( );

説明　　デバイスに関連付けられたすべてのスタティスティクスを0にリセットします。

ResetStatisticsが成功するには、**CapStatisticsReporting**プロパティと**CapUpdateStatistics**プロパティが両方ともtrueに設定していなければなりません。

ResetStatisticsは常に同期で実行されます。

エラー　　このメソッドを呼び出したときに、PosControlExceptionがスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

ResetStatisticsは、次のPosControlExceptionをスローすることがあります。

ErrorCodeの値	説明
Illegal	**CapStatisticsRepoting** か **CapUpdateStatistics** のいずれかが false に設定されています。
Extended	**ExtendedErrorStatistics** です。指定されたスタティスティクスの少なくとも 1 つがリセットできません。

ResetStatistics メソッド (StatisticsCategories)

形式 void ResetStatistics (StatisticCategories ***statistics***);

説明 指定されたカテゴリのすべてのスタティスティクスを0にリセットします。

ResetStatisticsが成功するには、**CapStatisticsReporting**プロパティと**CapUpdateStatistics**プロパティが両方ともtrueでなければなりません。

ResetStatisticsは常に同期で実行されます。

statisticsパラメータには、アプリケーションがデバイスのスタティスティクスをリセットしたいカテゴリを格納します。可能なカテゴリは、**StatisticsCategories**列挙体によって定義されています。

エラー このメソッドを呼び出したときに、PosControlExceptionがスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

ResetStatisticsは、次のPosControlExceptionをスローすることがあります。

ErrorCodeの値	説明
Illegal	次のいずれかの状況が発生しています。 **CapStatisticsRepoting** か **CapUpdateStatistics** のいずれかが false に設定されているか、 または、指定されたカテゴリが正しくありません。
Extended	**ExtendedErrorStatistics** です。指定されたスタティスティクスの少なくとも 1 つがリセットできません。

ResetStatistics メソッド (String[])

形式　　void ResetStatistics (string [] *statistics*);

説明　　指定されたスタティスティクスを0にリセットします。

ResetStatisticsが成功するには、**CapStatisticsReporting**プロパティと**CapUpdateStatistics**プロパティが両方ともtrueでなければなりません。

ResetStatisticsは常に同期で実行されます。

statisticsパラメータには、スタティスティクスをカンマで区切った文字列を格納します。

エラー　　このメソッドを呼び出したときに、PosControlExceptionがスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

ResetStatisticsは、次のPosControlExceptionをスローすることがあります。

ErrorCodeの値	説明
Illegal	次のいずれかの状況が発生しています。 **CapStatisticsRepoting** か **CapUpdateStatistics** のいずれかが false に設定されているか、 または、指定されたスタティスティクスの少なくとも 1 つが定義されていません。
Extended	**ExtendedErrorStatistics** です。指定されたスタティスティクスの少なくとも 1 つがリセットできません。

RetrieveStatistic メソッド (string)

形式 string RetrieveStatistic (string *statistic*);

説明 アプリケーションは、デバイススタティスティクスを指定して取得するときに、**RetrieveStatistic**を呼び出します。

RetrieveStatisticは常に同期で実行されます。

statisticパラメータには、取得するスタティスティクスを指定します。

RetrieveStatisticは、成功すると、スタティスティクスのXML文字列を返します。

エラー このメソッドを呼び出したときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

RetrieveStatisticは、次の**PosControlException**をスローすることがあります。

ErrorCodeの値	説明
Illegal	次のいずれかの状況が発生しています。 **CapStatisticsRepoting** が false に設定されているか（これは、デバイスがスタティスティクスの通知をサポートしていないことを示します）、 ***statistic*** パラメータが null か長さが 0 であるか、 または、指定されたスタティスティクスが存在しません。

RetrieveStatistics メソッド ()

形式　　string RetrieveStatistics ( );

説明　　アプリケーションは、すべてのデバイススタティスティクスを取得するときに、**RetrieveStatistics**を呼び出します。

RetrieveStatisticsは常に同期で実行されます。

RetrieveStatisticsは、成功すると、スタティスティクスのXML文字列を返します。

エラー　　このメソッドを呼び出したときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

RetrieveStatisticsは、次の**PosControlException**をスローすることがあります。

ErrorCodeの値	説明
Illegal	**CapStatisticsRepoting** が false に設定されています。これは、デバイスがスタティスティクスの通知をサポートしていないことを示します。

RetrieveStatistics メソッド (StatisticCategories)

形式 **string RetrieveStatistics (StatisticCategories *statistics*);**

説明 指定されたカテゴリのスタティスティクスを取得します。

RetrieveStatisticsは常に同期で実行されます。

statisticsパラメータには、アプリケーションが取得したいスタティスティクスのカテゴリを格納します。可能な値は、**StatisticCategories**列挙体によって定義されています。

RetrieveStatisticsは、成功すると、スタティスティクスのXML文字列を返します。

エラー このメソッドを呼び出したときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

RetrieveStatisticsは、次の**PosControlException**をスローすることがあります。

ErrorCodeの値	説明
Illegal	次のいずれかの状況が発生しています。 **CapStatisticsRepoting** が false に設定されているか（これは、デバイスがスタティスティクスの通知をサポートしていないことを示します）、 ***statistics*** パラメータが null か長さが 0 であるか、 または、指定されたスタティスティクスのカテゴリが正しくありません。

RetrieveStatistics メソッド (String[])

形式　　**string RetrieveStatistics (string [] *statistics*);**

説明　　指定されたカテゴリのスタティスティクスを取得します。

RetrieveStatisticsは常に同期で実行されます。

statisticsパラメータには、スタティスティクスをカンマで区切った文字列を格納します。指定された文字列のスタティスティクスを取得します。

RetrieveStatisticsは、成功すると、スタティスティクスのXML文字列を返します。

エラー　　このメソッドを呼び出したときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

RetrieveStatisticsは、次の**PosControlException**をスローすることがあります。

ErrorCodeの値	説明
Illegal	次のいずれかの状況が発生しています。 **CapStatisticsRepoting** が false に設定されているか（これは、デバイスがスタティスティクスの通知をサポートしていないことを示します)、 ***statistics*** パラメータが null か長さが 0 であるか、 または、指定されたスタティスティクスが 1 つ以上存在しません。

UpdateStatistic メソッド

形式 void UpdateStatistic (string *name*, object *value*);

説明 アプリケーションは、デバイススタティスティクスを指定してその値を変更するときに、**UpdateStatistic**を呼び出します。

UpdateStatisticが成功するには、**CapStatisticsReporting**プロパティと**CapUpdateStatistics**プロパティが両方ともtrueでなければなりません。

UpdateStatisticは常に同期で実行されます。

パラメータ	説明
Name	変更するスタティスティクスの名前
Value	スタティスティクスに設定する値

エラー このメソッドを呼び出したときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

UpdateStatisticは、次の**PosControlException**をスローすることがあります。

ErrorCodeの値	説明
Illegal	次のいずれかの状況が発生しています。 **CapStatisticsRepoting** か **CapUpdateStatistics** のいずれかが false に設定されているか、 または、指定されたスタティスティクスが存在しません。
Extended	**ExtendedErrorStatistics** です。指定されたスタティスティクスが変更できません。

UpdateStatistics メソッド (Statistic[])

形式　　void UpdateStatistics (Statistic [] ***statistics***);

説明　　一連のスタティスティクスを指定した値で変更します。

UpdateStatisticsが成功するには、**CapStatisticsReporting**プロパティと**CapUpdateStatistics**プロパティが両方ともtrueでなければなりません。

UpdateStatisticsは常に同期で実行されます。

statisticsパラメータには、**Statistic**クラスのインスタンス（名前と値のペア）の配列を格納します。

エラー　　このメソッドを呼び出したときに、**PosControlException**がスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

UpdateStatisticsは、次の**PosControlException**をスローすることがあります。

ErrorCodeの値	説明
Illegal	次のいずれかの状況が発生しています。 **CapStatisticsRepoting** か **CapUpdateStatistics** のいずれかが false に設定されているか、 ***statistics*** パラメータが null か、 または、指定されたスタティスティクスが 1 つ以上存在しません。
Extended	**ExtendedErrorStatistics** です。指定されたスタティスティクスの少なくとも 1 つが変更できません。

UpdateStatistics メソッド (StatisticCategories, Object)

形式 void UpdateStatistics (StatisticCategories *statistics*, object *value*);

説明 指定されたスタティスティクスのカテゴリを指定された値で変更します。

UpdateStatisticsが成功するには、**CapStatisticsReporting**プロパティと**CapUpdateStatistics**プロパティが両方ともtrueでなければなりません。

UpdateStatisticsは常に同期で実行されます。

パラメータ	説明
statistics	アプリケーションが変更したいスタティスティクスのカテゴリを格納します。可能なカテゴリは、**StatisticsCategories** 列挙体によって定義されています。
value	指定されたカテゴリのスタティスティクスを変更するのに使用する値を格納します。

エラー このメソッドを呼び出したときに、PosControlExceptionがスローされることがあります。詳しくは、「ステータス、状態モデル、および例外」をご覧ください。

UpdateStatisticsは、次のPosControlExceptionをスローすることがあります。

ErrorCodeの値	説明
Illegal	次のいずれかの状況が発生しています。 **CapStatisticsRepoting** か **CapUpdateStatistics** のいずれかが false に設定されているか、 または、指定されたスタティスティクスのカテゴリが正しくありません。
Extended	**ExtendedErrorStatistics** です。指定されたスタティスティクスの少なくとも 1 つが変更できません。

イベント

DataEvent イベント

説明　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示すために通知されます。**DataEventEnabled**プロパティはfalseとなり、アプリケーションがこのプロパティをtrueに戻すまで、データイベントはそれ以上発生しません。実際の入力データはデバイス固有のプロパティに格納されます。

データ受信時に**DataEventEnabled**がfalseの場合、データは内部バッファで待ち合わされ、デバイス固有の入力データのプロパティは更新されず、イベントも通知されません。（このプロパティがtrueに戻されたときに、待ち合わされている入力データがあり、**FreezeEvents**プロパティがfalseであれば、直ちにイベントの通知が行われます。）

DirectIOEvent イベント

説明　アプリケーションと直接通信するために、サービスオブジェクトが通知します。**DirectIOEvent**によって、デバイスに対して定義された他のイベントやプロパティではサポートされない情報を、イベントという形式で、サービスオブジェクトがアプリケーションに提供することが可能となります。このイベントを使用する場合は、アプリケーションは、サービスオブジェクトのこのイベントに対する必要性を認識していない他のベンダのデバイスを使用できないという制限を受ける可能性があります。

ErrorEvent イベント

説明　エラーが検出され、サービスオブジェクトの**State**がエラー状態に遷移したときに通知されます。

アプリケーションのコードの実行順序の制御（シーケンシング）が正しく行われるよう、**DataEventEnabled**プロパティがtrueになるまで、入力エラーイベントは通知されません。

OutputCompleteEvent イベント

説明　前に開始された非同期出力のリクエストが正常に終了したときに通知します。**OutputID**プロパティは、完了した非同期出力のリクエストのID番号を示します。

StatusUpdateEvent イベント

説明

サービスオブジェクトがデバイスステータスの変化の警報をアプリケーションに出す必要がある場合に通知します。

例えば、キャッシュドロワーの位置の変化（開閉）や、POSプリンタのセンサの変化（用紙の有無）、デバイスの電源状態の変化などです。

デバイスがイネーブルになったとき、アプリケーションにデバイスの状態を知らせるために、サービスオブジェクトは最初の**StatusUpdateEvent**を通知する場合もあります。しかし、この動作は必須ではありません。

POS for .NETのメンバとUnifiedPOSのメンバ

POS for .NETのクラスのメンバ名は、UnifiedPOS仕様書の名前と異なることがあります。ほとんどの場合、違いは大文字小文字の差だけです。（.NETでは、メソッド、プロパティ、およびイベントにPascal式命名規則を使用します。）例えば、UnifiedPOS仕様書の共通プロパティ**OutputID**は、POS for .NETでは**OutputId**です。

いくつかのデバイスでは、POS for .NETは、UnifiedPOS仕様書にはないプロパティやメソッドを導入しています。

次の表は、UnifiedPOS仕様書とは異なるプロパティ名の一例です。

UnifiedPOSのプロパティ	対応するPOS for .NETのプロパティ
CapMACCalculation	CapMacCalculation
DeviceServiceDescription	ServiceObjectDescription
DeviceServiceVersion	ServiceObjectVersion
OutputID	OutputId
POSKeyData	PosKeyData
POSKeyEventType	PosKeyEventType
PhysicalDeviceDescription	DeviceDescription
PhysicalDeviceName	DeviceName
なし	Compatibility
なし	DevicePath
なし	SynchronizingObject

次の表は、UnifiedPOS仕様書とは異なるメソッド名の一部です。

UnifiedPOSのメソッド	対応するPOS for .NETのメソッド
beginEFTTransaction	BeginEftTransaction
checkHealth	CheckHealth
claim	Claim
computeMAC	ComputeMac
DeviceServiceVersion	ServiceObjectVersion
directIO	DirectIO
enablePINEntry	EnablePinEntry
endEFTTransaction	EndEftTransaction
read	Read
resetStatistics	ResetStatistics
verifyMAC	VerifyMac
なし	ResetStatistic
なし	RetrieveStatistic
なし	UpdateStatistic

次の表は、UnifiedPOS仕様書とは異なるイベント名の一部です。

UnifiedPOSのイベントの属性	対応するPOS for .NETのEventArgクラスのプロパティ
OutputID	OutputId
なし	public DateTime TimeStamp {get; }

レガシーなOPOSサービスのために使用可能な暫定的な手順... Shimコードの使用方法　　第1.11版にて更新

.NETアーキテクチャは、現在および将来のWindowsオペレーティングシステムで使用される新しい機能を考慮に入れています。.NETアーキテクチャが提供するすべての機能を利用するには、新しいサービスオブジェクトを作成する必要があります。しかし、既存のOPOSサービスオブジェクトのソースコードをより迅速に.NET環境に移行するために、OPOS技術協議会（OPOS-J）によって、「Shim」と呼ばれる、ソフトウェアの変換をする中間層（ミドルレイヤ）が作成されました。「Shim」は、既存のOPOSベースの命名規則を利用して、.NETのサービスオブジェクトを開発または実装するためのモジュールです。OPOSサービスオブジェクトの提供者が、既存のOPOSサービスオブジェクトをPOS for .NETにポーティングする時、自由に利用できるようになっています。Shimを使用するいくつかの戦略的な理由を以下に挙げます。

POS for .NETでは、UnifiedPOS標準のメソッド定義を拡張しますが、この拡張に対応できるようにサービスオブジェクトを変更する必要が生じます。Shimは、この拡張に対応しており、既存OPOSのサービスオブジェクトの設計手法を利用して、.NETのサービスオブジェクトを作成することができます。

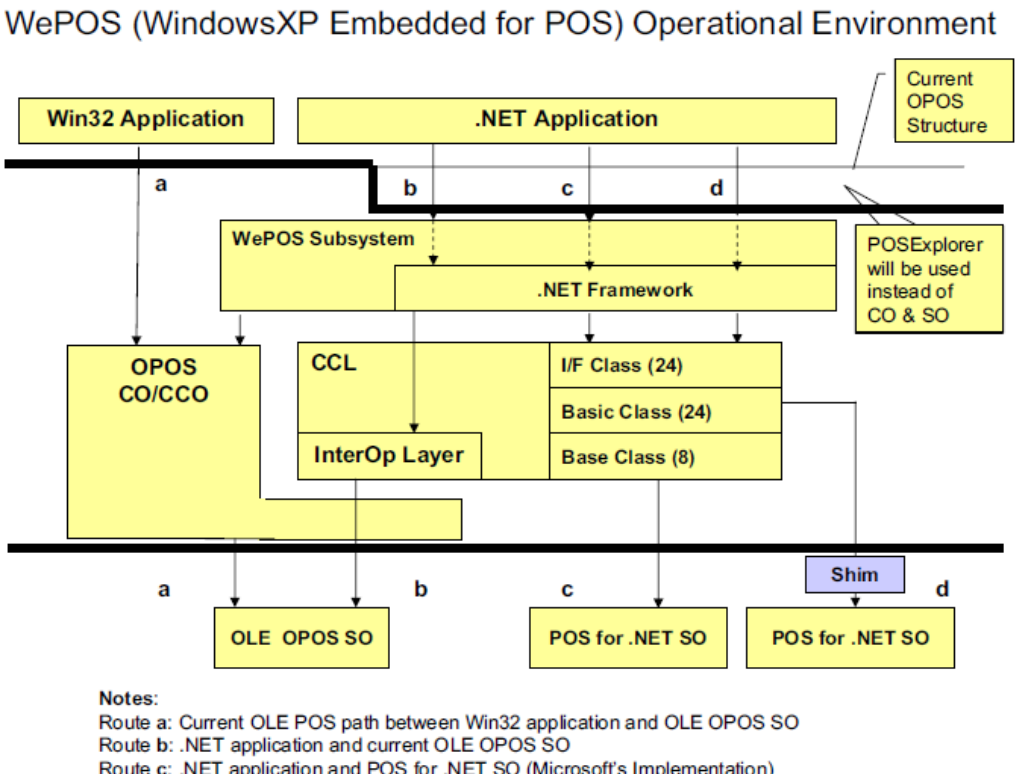
POS for .NETを利用する際は列挙体が要求されますが、列挙体はOPOSのサービスオブジェクトの実装では規定されていません。Shimは、既存のOPOSサービスオブジェクトソースコードからの名前の変更なしでパラメータの定数を列挙体にマップする機構を提供します。

重要な点は、Shimの使用の有無によって.NETアプリケーションを変更する必要はないということです。Shimは、既存OPOSの設計手法を利用したサービスオブジェクトとPOS for .NETのサービスオブジェクトとの差異をアプリケーションから見えなくします。POS for .NETフレームワークのサービスオブジェクトが利用できるならば、アプリケーションを変更せずに、Shimを用いたサービスオブジェクトと置換できます。

POSアプリケーションの開発は、この付録でこれまでに記したリファレンスに従う必要があります。唯一の違いは、UnifiedPOSとPOS for .NET環境をサポートするために使用されるサービスオブジェクトの開発にあります。Shimを使用すると、既存のOPOSのサービスオブジェクトソースコードの設計手法を最大限再利用することで、POS for .NETのサービスオブジェクトをより速く開発できます。

アーキテクチャ図　　第1.11版にて追加

以下の図は、OPOS、POS for.NET、Shim-POS for .NETのアーキテクチャの構造を示します。



実装の方法

Shimコードの命名規則

Shimコードは、以下に記載されているようにPOS for .NETのBasic classを拡張します。

Microsoft.PointOfService.BasicServiceObjects NameSpace.

Shimのクラスの名前は、次のルールに従います。

<*DeviceCategoryName*>+ShimBasic

例えば:
PosPrinterShimBasic

LineDisplayShimBasic

Shimのクラスを定義するファイル名は、次のルールに従います。

<Class Name>.cs

例えば:
PosPrinterShimBasic.cs

LineDisplayShimBasic.cs

Shimのクラスは、次の名前空間で定義されます。

Opos.PointOfService.BasicShimServiceObjects.

特定の列挙体を定義するファイルは、デバイスのカテゴリにより個々のファイルで指定されます。ファイル名は、OPOSのコモンコントロールオブジェクト（CCO）のヘッダーファイルと同じ名前を使用します。

例えば:
POSプリンタのための定数定義は、

OposPtr.cs

LineDisplayのための定数定義は、

OposDisp.cs

列挙体の名前は、定数を使用する関数のパラメータにより名付けられます。

例えば、POSプリンタがサポートするPrintBarCode関数で使用されるalignmentパラメータでは、次のようになります。

OposPtr.cs

```
Enum BarCodeAlignment
{
```

```
        Left = -1,
        Center = -2,
        Right = -3
}
```

列挙型は、次の名前空間で定義されます。

Opos.PointOfService

Shimのメソッドの再定義規則

この付録ですでに述べたように、POS for .NETのメソッドの呼び出しは、UnifiedPOS/ OPOSの実装とは異なる方法で行われます。例えば、OPOSでは情報を取得するのに別のメソッドの呼び出しが必要なのに対して、POS for .NETでは戻り値を使用します。Shimは、この動作の違いをマップする変換コードを提供します。

POS for .NETとOPOSで実装が異なるUnifiedPOS仕様の関数は、protectedレベルでabstract属性を用いて再定義されます。

例えば、**DirectIO**メソッドでは、次のようになります。

```
public override DirectIOData DirectIO (int command, int data, object obj)
    {
        ;
    }
protected abstract void DirectIO (int command, ref int data, ref object obj);
```

注意すべき点は、本例でUnifiedPOSが定義する仮想（abstract）関数である**DirectIO**は、POS for .NETアプリケーションの実装要件を満たす方法で呼ばれることです。一方、Shimコードは、OPOSの**DirectIO**メソッドや他のメソッドの呼び出しを処理するのに必要な関数を実行し、メソッドの機能やデータ交換を実現します。その後、Shimコードは、期待どおり、機能と結果のコードをPOS for .NETアプリケーションに返します。例を挙げます。

```
public override DirectIOData DirectIO (int command, int data, object obj)
    {
        this.DirectIO (command, ref data, ref obj);
        return new DirectIOData (data, obj);
    }

    /** 仮想関数は、Shimクラスを拡張するサービスオブジェクトで実装します。 **/
```

変換の対象と見なされる関数は、Shimクラス側で実装される可能性があります。これが起こるのを防ぐため、sealed属性を付加して、サービスオブジェクトのオーバーライドを防止します。

例えば:

public sealed override DirectIOData **DirectIO** (int command, int data, object obj)

ShimコードのIn/Outパラメータの規則

UnifiedPOS仕様書でIn/Out属性を付けて定義されたOPOSのパラメータは、POS for .NETの実装では異なる方法で扱われます。POS for .NETでは、データは戻り値で渡されることが期待されます。Shimコードは、In/Outパラメータに対して「ref」属性を用いることで、このマッピングを容易にします。この変換は、Shimコードによって自動的に処理され、呼び出し側のアプリケーションが意識する必要はありません。

管理方法

Shimコンポーネントのソースコードは、OPOS技術協議会によって管理されています。Shimのソースコードは、現在、次のWebサイトで公開されています。

http://www.monroecs.com/posfordotnet/shim.htm.

Shimコードのファイル名

次の表は、Shimコードで現在入手可能なファイルの一覧です。命名規則は、最大限のデバイス実使用感覚に基づいて作られています。新しいデバイスがリリースされると、Shimコードは、それを反映して更新されます。

Shimのファイル一覧

Shimクラスのファイル	説明
CashChangerShimBasic.cs	CashChangerのShimクラス
CashDrawerShimBasic.cs	CashDrawerのShimクラス
CatShimBasic.cs	CatのShimクラス
CheckScannerShimBasic.cs	CheckScannerのShimクラス
CoinDispenserShimBasic.cs	CoinDispenserのShimクラス
HardTotalsShimBasic.cs	HardTotalsのShimクラス
KeylockShimBasic.cs	KeylockのShimクラス
LineDisplayShimBasic.cs	LineDisplayのShimクラス
MicrShimBasic.cs	MicrのShimクラス
MsrShimBasic.cs	MsrのShimクラス
PinPadShimBasic.cs	PinPadのShimクラス
PointCardRWShimBasic.cs	PointCardRWのShimクラス
PosKeyboardShimBasic.cs	PosKeyboardのShimクラス
PosPowerShimBasic.cs	PosPowerのShimクラス
PosPrinterShimBasic.cs	PosPrinterのShimクラス
ScaleShimBasic.cs	ScaleのShimクラス
ScannerShimBasic.cs	ScannerのShimクラス
SmartCardRWShimBasic.cs	SmartCardRWのShimクラス
ToneIndicatorShimBasic.cs	ToneIndicatorのShimクラス
列挙体の定義ファイル	**説明**
OposCash.cs	CashDrawerの列挙体
OposCat.cs	Catの列挙体
OposChan.cs	CashChangerの列挙体
OposChk.cs	CheckScannerの列挙体
OposCoin.cs	CoinDispenserの列挙体
OposDisp.cs	LineDisplayの列挙体
OposKbd.cs	PosKeyBoardの列挙体
OposLock.cs	Keylockの列挙体
OposMicr.cs	Micrの列挙体
OposMsr.cs	Msrの列挙体
OposPcrw.cs	PointCardRWの列挙体
OposPpad.cs	PinPadの列挙体
OposPtr.cs	PosPrinterの列挙体
OposPwr.cs	PosPowerの列挙体
OposScal.cs	Scaleの列挙体
OposScan.cs	Scannerの列挙体
OposScrw.cs	SmartCardRWの列挙体
OposTone.cs	ToneIndicatorの列挙体
OposTot.cs	HardTotalsの列挙体
プロジェクトファイル	**説明**
AssemblyInfo.cs	アセンブリ情報ファイル

Opos.PointOfService.BasicShimServiceObjects.csproj	プロジェクトファイル

クラス図

インターフェイスクラス

public abstract DirectIOData DirectIO(int command, int data, object obj)
public abstract void ResetStatistic(string statistic)
public abstract void ResetStatistics()
public abstract void ResetStatistics(StatisticCategories statistics)
public abstract void ResetStatistics(string[] statistics)
public abstract string RetrieveStatistic(string statistic)
public abstract string RetrieveStatistics()
public abstract string RetrieveStatistics(StatisticCategories statistics)
public abstract string RetrieveStatistics(string[] statistics)
public abstract void UpdateStatistic(string name, object value)
public abstract void UpdateStatistics(Statistic[] statistics)
public abstract void UpdateStatistics(StatisticCategories statistics, object value)
:
:

Basic class

public override void ResetStatistic(string statistic)
public override void ResetStatistics()
public override void ResetStatistics(StatisticCategories statistics)
public override void ResetStatistics(string[] statistics)
public override string RetrieveStatistic(string statistic)
public override string RetrieveStatistics()
public override string RetrieveStatistics(StatisticCategories statistics)
public override string RetrieveStatistics(string[] statistics)
public override void UpdateStatistic(string name, object value)
public override void UpdateStatistics(Statistic[] statistics)
public override void UpdateStatistics(StatisticCategories statistics, object value)
:
:

Shimクラス

```
public sealed override DirectIOData DirectIO(int command, int data, object obj)
public sealed override void ResetStatistic(string statistic)
public sealed override void ResetStatistics()
public sealed override void ResetStatistics(StatisticCategories statistics)
public sealed override void ResetStatistics(string[] statistics)
public sealed override string RetrieveStatistic(string statistic)
public sealed override string RetrieveStatistics()
public sealed override string RetrieveStatistics(StatisticCategories statistics)
public sealed override string RetrieveStatistics(string[] statistics)
public sealed override void UpdateStatistic(string name, object value)
public sealed override void UpdateStatistics(Statistic[] statistics)
public sealed override void UpdateStatistics(StatisticCategories statistics, object value)

protected abstract void DirectIO(int command, ref int data, ref object obj)
protected abstract void ResetStatistics(string statistics)
protected abstract void RetrieveStatistics(ref string statistics)
protected abstract void UpdateStatistics(string statistics)
                    :
                    :
```

サービスクラス

```
protected override void DirectIO(int command, ref int data, ref object obj)
protected override void ResetStatistics(string statistics)
protected override void RetrieveStatistics(ref string statistics)
protected override void UpdateStatistics(string statistics)
                    :
                    :
```

第 3 章
ドロワー

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.2	Boolean	R/W	*適用外*
BinaryConversion	1.2	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.0	String	R	Open
Claimed	1.0	Boolean	R	Open
DataCount	1.2	Long	R	*適用外*
DataEventEnabled	1.0	Boolean	R/W	*適用外*
DeviceEnabled	1.0	Boolean	R/W	Open
FreezeEvents	1.0	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.0	Long	R	*適用外*
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.0	Long	R	--
ResultCodeExtended	1.0	Long	R	Open
State	1.0	Long	R	--
ControlObjectDescription	1.0	String	R	--
ControlObjectVersion	1.0	Long	R	--
ServiceObjectDescription	1.0	String	R	Open
ServiceObjectVersion	1.0	Long	R	Open
DeviceDescription	1.0	String	R	Open
DeviceName	1.0	String	R	Open
専用				
CapStatus	1.0	Boolean	R	Open
CapStatusMultiDrawerDetect	1.5	Boolean	R	Open

DrawerOpened	1.0	Boolean	R	Open & Enable

メソッド

共通	*版数*	*使用可能条件*
Open	1.0	--
Close	1.0	Open
ClaimDevice	1.0	Open
ReleaseDevice	1.0	Open & Claim
CheckHealth	1.0	Open & Enable (注)
ClearInput	1.0	*適用外*
ClearInputProperties	1.10	*適用外*
ClearOutput	1.0	*適用外*
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.0	Open
ResetStatistits	1.8	Open & Enable (注)
RetrieveStatistits	1.8	Open & Enable (注)
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistits	1.8	Open & Enable (注)
専用		
OpenDrawer	1.0	Open & Enable (注)
WaitForDrawerClose	1.0	Open & Enable (注)

（注） ドロワーを排他アクセスしているアプリケーションがないこと。

Events

名称	*版数*	*発生条件*
DataEvent	1.0	*適用外*
DirectIOEvent	1.0	Open & Enable
ErrorEvent	1.0	*適用外*
OutputCompleteEvent	1.0	*適用外*
StatusUpdateEvent	1.0	Open & Enable

概説

ドロワーコントロールのOLEプログラマティクIDは“OPOS.CashDrawer”です。

機能

ドロワーは次の機能を持っています。

ドロワーを開けるコマンドのサポート。

また、次の機能はオプションとして実装されます。

ドロワー状態の報告：1台のドロワーが接続されている状態でのドロワーの開閉状態が判断できるものです。

ドロワー特有の状態報告:複数のドロワーが接続されている状態でのドロワーの開閉状態が判断できるものです。

デバイスの共有

ドロワーは共有可能なデバイスです。デバイスを共有する際のルールは以下のとおりです。

デバイスをオープンしイネーブルにすると、アプリケーションはすべてのプロパティとメソッドを使用することができます。また、開閉状態の変化をステータスアップデートイベントにより受け取ることができます。

複数のアプリケーションがデバイスをオープンしイネーブルにすると、それらすべてのアプリケーションはプロパティとメソッドを扱うことができます。また、開閉状態の変化イベントは、それらすべてのアプリケーションに通知されます。

あるアプリケーションがドロワーの排他制御権を獲得すると、そのアプリケーションのみがOpenDrawerとWaitForDrawerCloseメソッドを呼び出すことができます。これによってある程度のセキュリティが保証されます。たとえば、メインとなるPOSアプリケーションが起動時にデバイスの排他アクセス権を獲得すると、これらのメソッドの制御はこのアプリケーションのみに限られます。

使用上の詳細な必要条件は、「一覧」を参照してください。

プロパティ

CapStatus プロパティ

形式　　BOOL CapStatus;

説明　　TRUE：ドロワーの開閉状態を通知することができます。
FALSE：ドロワーの開閉状態を通知することができません。

このプロパティは、**Open**メソッドにより初期化されます。

CapStatusMultiDrawerDetect プロパティ　*第1.5版にて追加*

形式　　BOOL CapStatusMultiDrawerDetect;

説明　　TRUE：複数ドロワー構成*のどのドロワーに対しても開閉状態を通知することができます。
FALSE：以下の通りになります。

DrawerOpenedプロパティがFALSEの場合、ドロワーは閉められています。

DrawerOpenedプロパティがTRUEの場合、少なくとも一つのドロワーは開けられています。特定のドロワーの状態については不明です。つまり、複数ドロワー構成*の場合、通知できるのは a）全てのドロワーが閉められている。 b）一つ以上のドロワーが開けられている。かのいずれかになります。

このプロパティは、**CapStatus**の値が TRUE の時に限り有効になります。

このプロパティは、**Open**メソッドにより初期化されます。

*複数ドロワー構成 － プリンタかターミナルが、一つのポートから複数個のドロワーをサポートしている構成。例えば、Yケーブルを使用して２つのドロワーを接続する構成。

DrawerOpened プロパティ

形式　　BOOL DrawerOpened;

説明　　TRUE：ドロワーは開けられています。
FALSE：ドロワーは閉められています。

CapStatusプロパティがFALSEの場合、デバイスは状態変化の通知を行うことができず、この**DrawerOpened**プロパティは常にFALSEとなっています。

CapStatusMultiDrawerDetectプロパティがFALSEの場合、少なくとも一つのドロワーが開けられているならば、**DrawerOpened**プロパティは TRUEになります。特定のドロワーの対しては不明です。詳細は**CapStatusMultiDrawerDetect**プロパティの項目を参照してください。

このプロパティは、デバイスをイネーブルにしたときに適切な値に初期化されます。

メソッド

OpenDrawer メソッド

形式 LONG OpenDrawer ()

説明 ドロワーを開けます。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	意味
OPOS_SUCCESS	正常終了。
その他	**ResultCode**の項目を参照してください。

WaitForDrawerClose メソッド

形式　LONG WaitForDrawerClose (LONG *BeepTimeout*, LONG *BeepFrequency*, LONG *BeepDuration*, LONG *BeepDelay*);

パラメータ	説明
BeepTimeout	ビープ音を鳴らすまでの待ち時間（ミリ秒）です。0の場合、直ちに鳴り始めます。
BeepFrequency	ビープ音の周波数（ヘルツ）です。
BeepDuration	ビープ音の長さです（ミリ秒）です。
BeepDelay	ビープ音間の空き時間（ミリ秒）です。

説明　ドロワーが閉じられるのを待つために使用されます。*BeepTimeout*ミリ秒経過してもドロワーが開いている場合には、警告としてビープ音が鳴らされます。

エラーが発生しない限り、このメソッドはドロワーが開いている間、アプリケーションに制御を戻しません。ドロワーが閉められると、鳴らされていたビープ音は止められます。

CapStatusMultiDrawerDetectプロパティがFALSEである場合、エラーが発生しない限り、このメソッドは複数ドロワー構成で一つ以上のドロワーが開いている間、アプリケーションに制御を戻しません。全てのドロワーが閉められると、鳴らされていたビープ音は止められます。

CapStatusプロパティがFALSEである場合には、デバイスは状態変化の通知を行うことができず、このメソッドは正常終了として即時に制御を返します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	意味
OPOS_SUCCESS	正常終了。
その他	**ResultCode**の項目を参照してください。

イベント

StatusUpdateEvent イベント　　*第1.13版で更新*

形式　　void StatusUpdateEvent (LONG *Status*);

*Status*パラメータには、最新のドロワー状態が設定されています。

第1.0版 - 第1.2版

ブール値としてのTRUE（0以外）であればドロワーが開けられ、反対に、FALSE（0）であれば閉められたことを意味します。

第1.3版以降

次の値のいずれかが設定されます。

値	意味
CASH_SUE_DRAWERCLOSED (= 0)	ドロワーは閉じています。
CASH_SUE_DRAWEROPEN (= 1)	ドロワーは開いています。 ドロワーが鍵などでロックされていないの場合のみ、通知されます。
*電源状態通知の**StatusUpdateEvent**の値*	「第2章 共通プロパティ、メソッド、イベント」の**StatusUpdateEvent**の記述を参照して下さい。（**PowerNotify**プロパティがOPOS_PN_ENABLEDの場合にのみ設定されます）

説明　　デバイスがイネーブルで、ドロワーの開閉状態が変更されたときに通知されます。デバイスがイネーブルされたとき、**StatusUpdateEvent**は、初期または現在の状態がアプリケーションに通知されます。しかし、この通知は必ず行われるとは限りません。そのため、アプリケーションはこの動作に依存した実装を行ってはいけません。

CapStatusプロパティがFALSEの場合は、デバイスは状態変化の通知を行うことができず、このイベントは通知されません。

CapStatusMultiDrawerDetectプロパティがFALSEの場合には、CASH_SUE_DRAWEROPENの値が(= 1)は、少なくとも一つのドロワーが開いている事を示します。特定のドロワーは不明です。詳細は、**CapStatusMultiDrawerDetect**の項目を参照してください。

第　4 章
ラインディスプレイ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.2	Boolean	R/W	*適用外*
BinaryConversion	1.2	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.0	String	R	Open
Claimed	1.0	Boolean	R	Open
DataCount	1.2	Long	R	*適用外*
DataEventEnabled	1.0	Boolean	R/W	*適用外*
DeviceEnabled	1.0	Boolean	R/W	Open & Claim
FreezeEvents	1.0	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.0	Long	R	*適用外*
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.0	Long	R	無し
ResultCodeExtended	1.0	Long	R	Open
State	1.0	Long	R	無し
ControlObjectDescription	1.0	String	R	無し
ControlObjectVersion	1.0	Long	R	無し
ServiceObjectDescription	1.0	String	R	Open
ServiceObjectVersion	1.0	Long	R	Open
DeviceDescription	1.0	String	R	Open
DeviceName	1.0	String	R	Open

専用

CapBlink	1.0	Long	R	Open
CapBitmap	1.7	Boolean	R	Open
CapBlinkRate	1.6	Boolean	R	Open
CapBrightness	1.0	Boolean	R	Open
CapCharacterSet	1.0	Long	R	Open
CapCursorType	1.6	Long	R	Open
CapCustomGlyph	1.6	Boolean	R	Open
CapDescriptors	1.0	Boolean	R	Open
CapHMarquee	1.0	Boolean	R	Open
CapICharWait	1.0	Boolean	R	Open
CapMapCharacterSet	1.7	Boolean	R	Open
CapReadBack	1.6	Long	R	Open
CapReverse	1.6	Long	R	Open
CapScreenMode	1.7	Boolean	R	Open
CapVMarquee	1.0	Boolean	R	Open
BlinkRate	1.6	Long	R/W	Open
DeviceWindows	1.0	Long	R	Open
DeviceRows	1.0	Long	R	Open
DeviceColumns	1.0	Long	R	Open
DeviceDescriptors	1.0	Long	R	Open
DeviceBrightness	1.0	Long	R/W	Open, Claim, & Enable
CharacterSet	1.0	Long	R/W	Open, Claim, & Enable
CharacterSetList	1.0	String	R	Open
MapCharacterSet	1.7	Boolean	R/W	Open
CurrentWindow	1.0	Long	R/W	Open
Rows	1.0	Long	R	Open
Columns	1.0	Long	R	Open
CursorRow	1.0	Long	R/W	Open
CursorColumn	1.0	Long	R/W	Open
CursorType	1.6	Long	R/W	Open
CursorUpdate	1.0	Boolean	R/W	Open
MarqueeType	1.0	Long	R/W	Open
MarqueeFormat	1.0	Long	R/W	Open
MarqueeUnitWait	1.0	Long	R/W	Open
MarqueeRepeatWait	1.0	Long	R/W	Open
InterCharacterWait	1.0	Long	R/W	Open
CustomGlyphList	1.6	String	R	Open

GlyphHeight	1.6	Long	R	Open
GlyphWidth	1.6	Long	R	Open
MaximumX	1.7	Long	R	Open
MaximumY	1.7	Long	R	Open
ScreenMode	1.7	Long	R/W	Open & Claim
ScreenModeList	1.7	String	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.0	無し
Close	1.0	Open
ClaimDevice	1.0	Open
ReleaseDevice	1.0	Open & Claim
CheckHealth	1.0	Open, Claim, & Enable
ClearInput	1.0	*適用外*
ClearInputProperties	1.10	*適用外*
ClearOutput	1.0	*適用外*
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.0	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用		
DisplayText	1.0	Open, Claim, & Enable
DisplayTextAt	1.0	Open, Claim, & Enable
ClearText	1.0	Open, Claim, & Enable
ScrollText	1.0	Open, Claim, & Enable
SetDescriptor	1.0	Open, Claim, & Enable
ClearDescriptors	1.0	Open, Claim, & Enable
CreateWindow	1.0	Open, Claim, & Enable
DestroyWindow	1.0	Open, Claim, & Enable
RefreshWindow	1.0	Open, Claim, & Enable
ReadCharacterAtCursor	1.6	Open, Claim, & Enable
DefineGlyph	1.6	Open, Claim, & Enable
DisplayBitmap	1.7	Open, Claim, & Enable

SetBitmap	1.7	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.0	*適用外*
DirectIOEvent	1.0	Open & Claim
ErrorEvent	1.0	*適用外*
OutputCompleteEvent	1.0	*適用外*
StatusUpdateEvent	1.3	Open, Claim, & Enable

概説

ラインディスプレイコントロールのOLEプログラマティックIDは“OPOS.LineDisplay”です。

機能

ラインディスプレイコントロールは次の機能を持っています。

テキスト文字の表示をサポートします。ディスプレイのデフォルトのモード（あるいは、唯一のモードとして）は文字表示です。

また、次の機能は、オプションとして実装されます。

マーキー[4]のようなスクロールを持ったウィンドウを提供します。ディスプレイは、水平マーキー、垂直マーキー、あるいは双方をサポートします。

テレタイプ効果を行うために、一文字ごとに時間間隔をおいて表示できます。

文字単位、あるいは、デバイス単位で周期の調節可能なブリンキングができます。

文字単位、あるいは、デバイス単位の反転表示ができます。

一つ以上のディスクリプターをサポートします。ディスクリプターとは、ある決まった名称を持った小さな表示です。これらは、アイテム、合計、釣り銭といったような取り引き状態を示すために用いられるものです。

デバイスの輝度を一つ以上のレベルで表現することにより、デバイスの輝度制御ができます。すべてのデバイスは、標準とブランク（少なくともソフトウェアの制御により）の2つの輝度を持たなければなりません。しかし、複数の輝度が可能なデバイスもあります。

アンダーライン、ブロックや反転表示を含む、複数のカーソルタイプをサポートします。

ピクセルレベルで文字単位にユーザー定義した“Glyphs”をサポートします。

スクリーンモードーデバイスでサポートしている行数や桁数ーの変更ができます。

ビットマップの登録と表示ができます。またこの機能を使うためにピクセルやドット単位の位置指定もサポートします。

4　マーキー：Marquee. ホテルなどの玄関前の歩道の上に差し掛けられたひさし。マーキーのようなスクロールとは、電光掲示板のようにメッセージが流れるスクロールを意味します。

モデル

ラインディスプレイの一般的なモデルは、次のようなものです。

一つ以上の行に文字を表示する一つ以上の桁を含んで成り立っています。デフォルトの文字設定中の文字は、文字設定を定義する性能とともに少なくとも以下のいずれかを含みます。

- 0から9までの数字、および、スペース、マイナス (‘-’)、ピリオド (‘.’)を含みます。
- 上記に加えて、‘A’から‘Z’までの大文字のアルファベットも含みます。
- すべてのASCII文字 (0x20～0x7F)。これらは、スペース、数字、大小文字のアルファベット、および、いくつかの特殊文字を含んでいます。

行と桁の位置は、ウィンドウの左上角を(0, 0)とした数値で割り振られます。

ウィンドウ0は常に以下のように定義されます。

- ビューポート － ウィンドウにより更新される表示領域部分 － は全表示領域となります。
- ウィンドウサイズは、全表示領域と同じです。

したがって、ウィンドウ0 － デバイスウィンドウと呼ばれることもあります － は、直接ディスプレイに対応します。

付加的なウィンドウを作成することができます。作られたウィンドウは次のような性格を持っています。

- そのビューポートは、表示領域のすべて、あるいは、一部分となります。
- デバイスの一部分、もしくは、すべての表示領域をビューポート[5]とした、水平方向、垂直方向のいずれか一方向において物理表示領域よりも大きなウィンドウを作成することができます。また、このウィンドウ内部をマーキーのようにスクロールすることができます。
- 各ウィンドウごとに、行数、桁数、現在のカーソル位置（行、桁）、カーソルタイプ、スクロールタイプ、タイマー値を保持しています。
- すべてのビューポートは透過的に扱われます。二つのビューポートが重なっている場合、各位置に表示される文字はいずれかのウィンドウにより最後に表示された文字となります。

---

[5] ビューポート：Viewport. デバイスの一部分、あるいは、全領域に割り当てられた可視領域のこと。

ディスプレイのモード

ラインディスプレイのモードには次の3つがあります。

イミディエイトモード
MarqueeTypeプロパティがDISP_MT_NONEで、InterCharacterWaitプロパティが0のときのモードです。
ウィンドウがビューポートより大きい場合には、ビューポートに対応する文字のみが表示されます。

テレタイプモード
MarqueeTypeプロパティがDISP_MT_NONEで、InterCharacterWaitプロパティが0以外のときのモードです。
DisplayText メソッドと DisplayTextAt メソッドの要求はキュー管理され、受け付けられた順に処理されます。InterCharacterWait タイマー値は、各文字を表示する間隔時間を意味します。 InterCharacterWait プロパティ値は、ビューポート内に表示される文字に対してのみ適用されます。

マーキーモード
MarqueeTypeプロパティがDISP_MT_NONE以外のときのモードです。
ウィンドウはビューポートよりも大きくなくてはなりません。
マーキースクロールは、MarqueeType プロパティをDISP_MT_INITに設定しマーキー初期化モードに入り、そして、ClearText メソッド、DisplayText メソッド、DisplayTextAt メソッドを使用して初期化されます。その後、MarqueeType プロパティがスクロールを開始する値に設定されたときにマーキー実行モードへと移り、マーキースクロールは、ウィンドウの先頭（あるいは、方向が右、あるいは、下の場合には末尾）からビューポート内で表示され始めます。
モードがマーキー実行モードから非モードへと設定された場合には、マーキーはその表示状態で終了します。その後、引き続きマーキー実行モードへと設定された場合には、その表示状態からスクロールされます。
マーキー実行モードからマーキー初期化モードへと設定された場合には、マーキーは終了します。新しい内容でマーキースクロールを実行するには、ウィンドウ上へ文字列を編集してからマーキー実行モードへと設定します。
マーキー実行モード中に DisplayText、DisplayTextAt、ClearText、RefreshWindow、ScrollText 各メソッドを使用することは禁止されています。

表字データとエスケープシーケンス　　第1.7版にて追加

全てのラインディスプレイのデフォルトの文字設定は、少なくとも0x20から0x7FのASCII文字をサポートすると仮定されます。これは、スペース、数字、大文字、小文字、いくつかの特種な文字を含んでいます。もし、ラインディスプレイが小文字をサポートしていなければ、サービスオブジェクトは小文字を大文字に変換する事があります。

第1.7版からエスケープシーケンスのサポートが開始された

あらゆるエスケープシーケンスは、後ろに垂直バー（'|'）が続くエスケープ文字ESC（この値は10進法で27）で始まります。この後ろには、0またはより大きな数、小文字のアルファベットが続きます。エスケープシーケンスは、大文字のアルファベットで終わります。

DisplayTextと**DisplayTextAt**メソッドの文字列データの中で以下のエスケープシーケンスが認識されます。もし、エスケープシーケンスがラインディスプレイによってサポートされないオペレーションを指定すれば、そのエスケープシーケンスは無視されます。

（1）指定時のみ、動作するエスケープシーケンス

名称	データ	内容
ビットマップ表示	ESC |#B	前もって保存されていたビットマップを表示します。文字'#'はビットマップ番号です。

（2）表示時に、動作するエスケープシーケンス

各表示メソッドの終わりや、あるいは"ノーマル"シーケンスによってリセットされる特徴があります。

名称	データ	内容
反転文字	ESC |rvC	明暗を反対にして表示します。
ブリンク	ESC |kC	ブリンクする文字を表示します。
ノーマル	ESC |N	ラインディスプレイの特徴を通常の状態に回復させます。

デバイスの共有

ラインディスプレイは、以下に示すように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に、排他アクセス権を獲得しなければなりません。

アプリケーションは、プロパティを参照、あるいは、デバイスを更新するメソッドを呼び出す前には、デバイスの排他アクセス権を獲得し、イネーブルにしなければなりません。

使用上の必要条件に関しては、「一覧」を参照してください。

プロパティ

BlinkRate プロパティ　R/W　　第1.6版にて追加

形式　　LONG BlinkRate;

説明　　ブリンク周期をミリ秒で示します。ブリンク周期とは、ブリンク中にテキストがいったん表示されてから非表示を経て次に表示されるまで(on-off-on)の時間です。このプロパティの設定時にディスプレイはブリンク周期を、サポートしている一番近い値に設定し、**BlinkRate**プロパティにもその実際の値を反映させます。ブリンク周期はハードウェアに依存し、正確なミリ秒の粒度で制御できるとは限らないため、こうした調整を必要とします。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値　　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	デバイスは、**BlinkRate**をサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照　　CapBlinkRateプロパティ

CapBitmap プロパティ　　第1.7版にて追加

形式　　BOOL CapBitmap;

説明　　TRUE：ビットマップの表示ができます。
FALSE：できません。

このプロパティは、**Open**メソッドにより初期化されます。

CapBlink プロパティ

形式　　LONG CapBlink;

説明　　デバイスの文字ブリンクの種別を示します。以下の値のいずれかが設定されます。

値	説明
DISP_CB_NOBLINK	ブリンキングはサポートされません。
DISP_CB_BLINKALL	ブリンキングはサポートされます。ディスプレイ上のすべての文字が同時にブリンクするか、またはブリンクしません。
DISP_CB_BLINKEACH	ブリンキングはサポートされます。任意の文字単位にブリンクを行うことが可能です。

このプロパティは、**Open**メソッドにより初期化されます。

CapBlinkRate プロパティ　　第1.6版にて追加

形式　　BOOL CapBlinkRate;

説明　　TRUE：デバイスは**BlinkRate**プロパティを使ってブリンク周期の読み取りと制御ができます。
FALSE：制御できません。

このプロパティは、**Open**メソッドにより初期化されます。

CapBrightness プロパティ

形式　　BOOL CapBrightness;

説明　　TRUE：輝度の制御ができます。
FALSE：制御できません。

このプロパティは、**Open**メソッドにより初期化されます。

CapCharacterSet プロパティ　　第1.5版にて更新

形式　LONG CapCharacterSet;

説明　デバイスがデフォルトで表示できる文字の種類を示します。以下の値のいずれかが設定されます。

値	説明
DISP_CCS_NUMERIC	表示できる文字は、0～9の数字、スペース、マイナス（'-'）、ピリオド（'.'）のみです。
DISP_CCS_ALPHA	DISP_CCS_NUMERICに加えて大文字のアルファベットが表示可能です。
DISP_CCS_ASCII	0x20～0x7FまでのASCII文字が表示可能です。
DISP_CCS_KANA	文字設定は、コードページ932の一部をサポートします。それは、0xA1と0xDFの間の半角のカタカナと、0x20と0x7Fの間の全てのASCII文字を含んでいますが、漢字は含まれていません。
DISP_CCS_KANJI	文字設定は、コードページ932をサポートします。それは、0xA1と0xDFの間の半角のカタカナ、0x20と0x7Fの間の全てのASCII文字をサポートします。また、JIS第一水準、JIS第二水準レベルで定義される文字のシフトJISコード文字をサポートします。
DISP_CCS_UNICODE	文字設定は、ユニコードをサポートします。

デフォルトの文字設定は、これらの範囲以上の文字を含んでいる場合があります。**CharacterSet**プロパティの初期設定値により付加的な情報を調べることができます。

このプロパティは、**Open**メソッドにより初期化されます。

CapCursorType プロパティ　　第1.8版にて更新

形式　　LONG CapCursorType;

説明　　デバイスがサポートしていて、CursorTypeプロパティで選択できるカーソルの種類をビット単位で示します。以下の値が設定されます。

値	説明
DISP_CCT_NONE	カーソルは表示できません。
DISP_CCT_FIXED	特定のカーソルが常に表示されます。
DISP_CCT_BLOCK	ブロック型のカーソルが表示可能です。
DISP_CCT_HALFBLOCK	ハーフブロック型のカーソルが表示可能です。
DISP_CCT_UNDERLINE	アンダーライン型のカーソルが表示可能です。
DISP_CCT_REVERSE	文字反転型のカーソルが表示可能です。
DISP_CCT_BLINK	カーソルのブリンク表示が可能です。
DISP_CCT_OTHER	その他の型のカーソルが表示可能です。

DISP_CCT_NONEが設定されていた場合は、他のビットはセットされません。これはカーソルが表示可能でないためです。

DISP_CCT_FIXEDが設定されていた場合は、DISP_CCT_BLINKが設定可能で、さらに他のビットのうちただ１つだけがセットされています。このビットは常に表示されているカーソルタイプを示します。

このプロパティは、Openメソッドにより初期化されます。

CapCustomGlyph プロパティ　　第1.6版にて追加

形式　　BOOL CapCustomGlyph;

説明　　デバイスのGlyph定義能力を示します。TRUEならばデバイスはCustomGlyphの定義ができます。FALSEならばできません。

このプロパティは、Openメソッドにより初期化されます。

CapDescriptors プロパティ

形式　　BOOL CapDescriptors;

説明　　TRUE：ディスクリプターをサポートします。
FALSE：サポートしません。

このプロパティは、Openメソッドにより初期化されます。

CapHMarquee プロパティ

形式　　BOOL CapHMarquee;

説明　　TRUE：水平方向のマーキースクロールをサポートします。
FALSE：サポートされません。

このプロパティは、Openメソッドにより初期化されます。

CapICharWait プロパティ

形式　BOOL CapICharWait;

説明　TRUE：文字の表示時間遅延をサポートします。
FALSE：サポートしません。

このプロパティは、Openメソッドにより初期化されます。

CapMapCharacterSet プロパティ　第1.7版にて追加

形式　BOOL CapMapCharacterSet;

説明　データ表示の際にアプリケーションの文字を選択された文字セットにマップするサービスオブジェクトの機能を示します。

TRUEならばデバイスは文字をCharacterSetList に定義された文字セットにマップできます。FALSEならばできません。

このプロパティは、Openメソッドにより初期化されます。

参照　CharacterSetプロパティ、MapCharacterSetプロパティ、CharacterSetListプロパティ

CapReadBack プロパティ　第1.6版にて追加

形式　LONG CapReadBack;

説明　デバイスに表示されている文字の読み取り能力の種別を示します。以下の値のいずれかが設定されます。

値	説明
DISP_CRB_NONE	読み取りはサポートされません。
DISP_CRB_SINGLE	１度に１文字の読み取りをサポートします。

このプロパティは、Openメソッドにより初期化されます。

CapReverse プロパティ　　第1.6版にて追加

形式　LONG CapReverse;

説明　デバイスの反転表示の種別を示します。以下の値のいずれかが設定されます。

値	説明
DISP_CR_NONE	反転はサポートされません。
DISP_CR_REVERSEALL	反転はサポートされます。ディスプレイ上のすべての文字が同時に反転するか、または反転しません。
DISP_CR_REVERSEEACH	反転はサポートされます。任意の文字単位に反転を行うことが可能です。

このプロパティは、Openメソッドにより初期化されます。

CapScreenMode プロパティ　　第1.7版にて追加

形式　BOOL CapScreenMode;

説明　TRUEならばデバイスはスクリーンモード（例えばデバイス上のテキスト行数桁数）を変更できます。FALSEならばできません。

このプロパティは、Openメソッドにより初期化されます。

参照　ScreenModeプロパティ、ScreenModeListプロパティ

CapVMarquee プロパティ

形式　BOOL CapVMarquee;

説明　TRUE：垂直方向のマーキースクロールをサポートします。
FALSE：サポートしません。

このプロパティは、Openメソッドにより初期化されます。

CharacterSet プロパティ　R/W　　第1.10版にて更新

形式　　LONG CharacterSet;

説明　　表示している文字のキャラクターセット（文字設定）が設定されます。

以下の値のいずれかが設定されます。

値	説明
101-199までの値	コードページ、または、ASCII、Windows ANSI 文字設定のいずれにも相当しないデバイス特殊文字設定です。
400-990までの値	コードページ、標準の値のいずれかに相当します。
DISP_CS_UNICODE	ユニコードを設定します。この定数値は997です。
DISP_CS_ASCII	ASCIIキャラクタを設定します。0x20と0x7Fの間のASCIIキャラクタをサポートします。この定数値は998です。
DISP_CS_ANSI	DISP_CS_WINDOWSと同じ値を持っています。
DISP_CS_WINDOWS	Windows ANSIキャラクタを設定します。この定数値は999です。これは正確には、Windowsのコードページ1252と等しい。
Range 1000 and higher	Windowsのコードページ；標準値の一つと適合します。

このプロパティを使うための実装に依存する追加情報として「キャラクターセットのマッピング」を参照してください。

このプロパティは、**Open**メソッドを呼び出し、デバイスをイネーブルにしたときに適切な値に初期化されます。この値は、**CapCharacterSet**プロパティで指定される文字設定が少なくともサポートされます。

戻り値　　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に設定されました。

その他	**ResultCode**の項目を参照してください。

参照　　**CharacterSetList**プロパティ、**CapCharacterSet**プロパティ

CharacterSetList プロパティ

形式　BSTR CharacterSetList;

説明　文字設定番号が文字列で設定されます。

このプロパティは、**Open**メソッドにより初期化されます。文字列は、コンマで区切られたASCII数字により成り立っています。

たとえば、文字列が“101,850,999”の場合には、デバイスはデバイス特有の文字設定、コードページ850番、および、Windows ANSI文字設定をサポートします。

参照　CharacterSetプロパティ

Columns プロパティ

形式　LONG Columns;

説明　カレントウィンドウの桁数が設定されています。

ウィンドウ番号0では、**Columns**プロパティは、**DeviceColumns**プロパティと同じです。その他のウィンドウでは、**DeviceColumns**プロパティより小さいか大きい場合があります。

このプロパティは、**Open**メソッドにより**DeviceColumns**プロパティの値に初期化されます。また、**CurrentWindow**プロパティを設定、または、**CreateWindow**メソッド、**DestroyWindow**メソッドを呼び出すことにより更新されます。

参照　Rowsプロパティ

CurrentWindow プロパティ　R/W　第1.6版にて更新

形式　LONG CurrentWindow;

説明　カレントウィンドウの番号が設定されており、該ウィンドウのテキストが表示されます。

いくつかのプロパティが各ウィンドウに関連付けられています。それらには、**Rows**、**Columns**、**CursorRow**、**CursorColumn**、**CursorUpdate**、**CursorType**、**MarqueeFormat**、**MarqueeType**、**MarqueeUnitWait**、**MarqueeRepeatWait**、**InterCharacterWait**があります。

このプロパティが変更されるとカレントウィンドウが変わり、関連するプロパティはこのウィンドウの値へと設定されます。

このプロパティへウィンドウ番号の設定を行ってもそのビューポート内容は更新されません。このウィンドウと他のウィンドウのビューポートがオーバーラップし、他のウィンドウがビューポートの内容を更新した場合に、このウィンドウのビューポート内容を戻すために**RefreshWindow**メソッドが呼び出されます。

このプロパティは**Open**メソッドにより0（デバイスウィンドウ）に初期化されます。また、**CreateWindow**メソッド、または、**DestroyWindow**メソッドの呼び出しにより更新されます。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	無効なウィンドウ値が指定されました。

CursorColumn プロパティ　R/W

形式　LONG CursorColumn;

説明　カレントウィンドウの、次の表示文字が出力される桁が設定されています。

有効な値は、0から(**Columns**)の範囲です。（**DisplayText**メソッドの“**CursorColumn = Columns**“の説明に関する備考を参照のこと。）

このプロパティは、**Open**メソッド、および、**CreateWindow**メソッドにより0に初期化されます。また、**CurrentWindow**プロパティの設定、**ClearText**メソッド、または、**DestroyWindow**メソッドの呼び出しにより更新されます。**CursorUpdate**プロパティがTRUEの場合には、**DisplayText**メソッド、**DisplayTextAt**の呼び出しにより更新されます。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	無効なカーソル桁数値が指定されました。

参照　**CursorRow**プロパティ、**DisplayText**メソッド

CursorRow プロパティ　R/W

形式　　LONG CursorRow;

説明　　カレントウィンドウの、次の表示文字が出力される行が設定されています。

有効な値は、0から(**Rows** - 1)の範囲です。

このプロパティは、**Open**メソッド、および、**CreateWindow**メソッドにより0に初期化されます。また、**CurrentWindow**プロパティの設定、**ClearText**メソッド、または、**DestroyWindow**メソッドの呼び出しにより更新されます。**CursorUpdate**プロパティがTRUEの場合には、**DisplayText**メソッド、**DisplayTextAt**メソッドの呼び出しにより更新されます。

戻り値　　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	無効なカーソル行値が指定されました。

参照　　**CursorColumn**プロパティ、**DisplayText**メソッド

CursorType プロパティ　R/W　　第1.8版にて更新

形式　　LONG CursorType;

説明　　カレントウィンドウの、カーソルタイプを示します。以下の値のいずれかが設定されます。

値	説明
DISP_CT_NONE	カーソルを表示しません。
DISP_CT_BLOCK	ブロック型のカーソルを表示します。
DISP_CT_HALFBLOCK	ハーフブロック型のカーソルを表示します。
DISP_CT_UNDERLINE	アンダーライン型のカーソルを表示します。
DISP_CT_REVERSE	文字反転型のカーソルを表示します。
DISP_CT_BLINK	カーソルのブリンク表示を行います。この値はこのプロパティ用に定義されている他の値の１つとの論理和にて使います。
DISP_CT_OTHER	その他の型のカーソルを表示します。

CapCursorTypeプロパティにDISP_CCT_NONEまたはDISP_CCT_FIXEDが設定されていた場合は、このプロパティは書き込みできません。他の場合は**CapCursorType**に指定されているカーソルタイプの１つを設定できます。そしてカーソルのブリンク表示がサポートされていれば、そのカーソルタイプにDISP_CT_BLINKを論理和することができます。

このプロパティは，各ウィンドウ毎に管理されます。このプロパティへの設定はカレントウィンドウにだけ反映されます。カレントウィンドウだけが表示可能なカーソルを持つからです。

このプロパティは、**Open**メソッド、および、**CreateWindow**メソッドによりDISP_CT_NONE(または**CapCursorType**にDISP_CCT_FIXEDが設定されていた場合は、対応する適切なタイプ)に初期化されます。また、**CurrentWindow**プロパティの設定、または、**DestroyWindow**メソッドの呼び出しにより更新されます。

戻り値　　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	正常に設定されました。

OPOS_E_ILLEGAL	**CapCursorType**にDISP_CCT_NONEかDISP_CCT_FIXEDが設定されているか、無効なカーソルタイプ値が指定されました。

参照　**CapCursorType**プロパティ

CursorUpdate プロパティ　R/W

形式　BOOL CursorUpdate;

説明　TRUE：**DisplayText**メソッド、または、**DisplayTextAt**メソッドで文字が表示されると、**CursorRow**プロパティと**CursorColumn**プロパティは、表示された最後の文字の次を指すように更新されます。

FALSE：文字が表示されてもカーソルプロパティは更新されません。

このプロパティは，各ウィンドウ毎に管理されます。

このプロパティは、**Open**メソッド、および、**CreateWindow**メソッドによりTRUEに初期化されます。また、**CurrentWindow**プロパティの設定、または、**DestroyWindow**メソッドの呼び出しにより更新されます。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	正常に設定されました。

参照　**CursorRow**プロパティ、**CursorColumn**プロパティ

CustomGlyphList プロパティ　　第1.6版にて追加

形式　　BSTR CustomGlyphList;

説明　　Glyphとして定義出来る文字コードが文字列で設定されます。

文字コードは、2桁(ASCII)または4桁(ユニコード)の16進数で表されます。これらの値は ‘-’文字を使った範囲指定表現を含み、コンマで区切られています。

このプロパティは、**Open**メソッドにより初期化されます。

たとえば、文字列が“2D,4D”の場合には、デバイスはそれぞれ“-”,“M”文字のGlyph定義をサポートしています。文字列が“002D-004D”の場合には、デバイスはユニコードで002Dから004Dまでのすべての Glyph定義をサポートしています。また、文字列が“2D-2F,3D-3F”の場合には、デバイスは16進数で2Dから2Fまでと、3Dから3Fまでの範囲のGlyph定義をサポートしています。

参照　　**CapCustomGlyphプロパティ、GlyphWidthプロパティ、GlyphHeightプロパティ、DefineGlyphメソッド**

DeviceBrightness プロパティ　R/W

形式　　LONG DeviceBrightness;

説明　　0から100までのパーセント表現によるデバイスの輝度が設定されています。

すべてのデバイスは、0%（ブランク）と100%（最大の明るさ）をサポートします。最低限、スペース文字を送出することによりブランクがサポートされます。**CapBrightness**プロパティがTRUEであると、デバイスは一つ以上の輝度をサポートします。

デバイスが指定された輝度をサポートできない場合には、サービスオブジェクトにより適切な値に置き換えられます。

このプロパティは、**Open**メソッドの呼出し後にデバイスを最初にイネーブルにしたときに100に初期化されます。

戻り値　　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。0から100の範囲外の値です。

DeviceColumns プロパティ　　第1.7版にて更新

形式　　LONG DeviceColumns;

説明　　デバイスの桁数が設定されています。

このプロパティは、**Open**メソッドにより初期化されます。
ScreenModeプロパティの変更により更新されます。

参照　　DeviceRowsプロパティ、ScreenModeプロパティ

DeviceDescriptors プロパティ

形式　LONG DeviceDescriptors;

説明　デバイスのディスクリプターの個数が設定されています。

CapDescriptorsプロパティがTRUEの場合には、DeviceDescriptorsプロパティは0以外の値が設定されています。そうでない場合には、0となっています。

このプロパティは、Openメソッドにより初期化されます。

参照　SetDescriptorメソッド、ClearDescriptorsメソッド

DeviceRows プロパティ　　第1.7版にて更新

形式　LONG DeviceRows;

説明　デバイスの行数が設定されています。

このプロパティは、Openメソッドにより初期化されます。
ScreenModeプロパティの変更により更新されます。

参照　DeviceColumnsプロパティ、ScreenModeプロパティ

DeviceWindows プロパティ

形式　　LONG DeviceWindows;

説明　　サポートできるウィンドウの最大数が設定されています。0の場合には、デバイスウィンドウのみがサポートされ、新たなウィンドウを作成することはできないことを意味しています。

このプロパティは、Openメソッドにより初期化されます。

参照　　CurrentWindowプロパティ

GlyphHeight プロパティ　　第1.6版にて追加

形式　　LONG GlyphHeight;

説明　　１文字のピクセル数を元にしたGlyphの高さを示します。

このプロパティは、Openメソッドにより初期化されます。

参照　　CapCustomGlyphプロパティ、CustomGlyphListプロパティ、DefineGlyphメソッド

GlyphWidth プロパティ　　第1.6版にて追加

形式　　LONG GlyphWidth;

説明　　１文字のピクセル数を元にしたGlyphの幅を示します。

このプロパティは、Openメソッドにより初期化されます。

参照　　CapCustomGlyphプロパティ、CustomGlyphListプロパティ、DefineGlyphメソッド

InterCharacterWait プロパティ　R/W

形式　LONG InterCharacterWait;

説明　**DisplayText**メソッドと**DisplayTextAt**メソッド使用時に各文字を表示する際の遅延時間が設定されています。このタイマーは、テキスト表示時にテレタイプのような効果を与えます。

このプロパティは、ウィンドウがマーキーモード以外 － つまり、**MarqueeType**プロパティがDISP_MT_NONEに設定されている場合 － のときのみ参照されます。

このプロパティが0以外でウィンドウがマーキーモードでない場合には、ウィンドウはテレタイプモードとなります： **DisplayText**メソッドと **DisplayTextAt**メソッドの要求はキュー管理され、受け付けられた順に処理されます。**InterCharacterWait**タイマー値は、各文字を表示する間隔時間を意味します。この時間はミリ秒単位で指定されます。（タイマー精度により誤差が生じる場合があります。）**CursorUpdate**プロパティがTRUE の場合には、**CursorRow**プロパティと **CursorColumn**プロパティは、**DisplayText**メソッド、**DisplayTextAt**メソッドがリターンする前に適切な値に更新されます。これは、例えすべての文字列が表示されきっていない場合にも適用されます。

タイマー値が0でマーキーモードでない場合には、イミディエイトモードとなります。文字列は即時に表示されます。このプロパティが0に設定されたちょうどそのときにキュー管理されていたいくつかの表示要求がある場合には、それら要求は可能な限り即時に表示されます。

CapICharWaitがFALSEの場合には、表示の遅延時間はサポートされず、また、このプロパティは使用されません。

このプロパティは、**Open**メソッド、および、**CreateWindow**メソッドにより0に初期化されます。また、**CurrentWindow**プロパティの設定、または、**DestroyWindow**メソッドの呼び出しにより更新されます。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

参照　**DisplayText**メソッド

MapCharacterSet プロパティ　R/W　第1.7版にて追加

形式　　BOOL MapCharacterSet;

説明　　TRUEならばサービスオブジェクトはデータ表示の際にアプリケーションから渡された文字を**CharacterSet**プロパティで選択された文字セットにマップします。

FALSEならばマッピングをサポートしません。この場合アプリケーションは**CharacterSet**プロパティで選択された文字セットへのマップを保証する必要があります。

CapMapCharacterSetプロパティがFALSEならば、このプロパティは常にFALSEです。

このプロパティは、**Open**メソッドにより初期化されます。

参照　　CharacterSetプロパティ、CapMapCharacterSetプロパティ

MarqueeFormat プロパティ　R/W

形式　　LONG MarqueeFormat;

説明　　カレントウィンドウのマーキースクロールの型が設定されています。

以下の値のいずれかが設定されます。

値	説明
DISP_MF_WALK	反対側からデータを置き進めることによりマーキースクロールが開始されます。たとえば、マーキータイプが左方向の場合には、ビューポートの右側に文字が置かれ、左方向へスクロールされていきます。
DISP_MF_PLACE	文字を置くようにマーキースクロールを開始します。たとえば、マーキータイプが左方向の場合には、ビューポートは左側から文字が置かれていきます。そして、ビューポートが一杯になった後、スクロールが開始されます。

MarqueeFormatプロパティは、**Open**メソッド、**CreateWindow**メソッドによりDISP_MF_WALKに初期化されます。また、**CurrentWindow**プロパティへの設定、あるいは、**DestroyWindow**メソッドの呼び出しにより更新されます。

MarqueeFormatプロパティは、マーキー実行モードに移行したときに参照されます。また，**MarqueeFormat**プロパティは、マーキーモード以外の場合には参照されません。

MarqueeFormatがDISP_MF_WALKでマーキー初期化モードからマーキー実行モードへと移った場合には、以下のことが行われます。

1. 次のようにウィンドウをビューポートに対応させます。

MarqueeType	ウィンドウ		ビューポート
左方向	第一桁目	=	最終桁
上方向	第一行目	=	最終行
右方向	最終桁	=	第一桁目
下方向	最終行	=	第一行目

   ビューポートをブランクにし、即座に第2ステップへと移ります。

2. ウィンドウの対応領域をビューポートへ表示し、**MarqueeUnitWait**ミリ秒待ちます。マーキー方向に一行、あるいは、一桁分だけビューポートに対応するウィンドウを移動します。これをビューポートが一杯になるまで繰り返します。

3. ビューポートをリフレッシュします。その後、**MarqueeUnitWait**ミリ秒間ウェイトします。ウィンドウの対応領域を一行、あるいは、一桁分移動します。これを最終行、あるいは、最終桁がビューポート内でスクロールされるまで繰り返します。（この場合、**MarqueeUnitWait**ミリ秒間のウェイトは省かれます）。

4. **MarqueeRepeatWait**ミリ秒間ウェイトします。その後、第一ステップへ戻ります。

MarqueeFormatがDISP_MF_PLACEでマーキー初期化モードからマーキー実行モードへと移った場合には、以下のことが行われます。

1. 次のようにウィンドウをビューポートに対応させます。

MarqueeType	ウィンドウ		ビューポート
左方向	第一桁目	=	第一桁目
上方向	第一行目	=	第一行目
右方向	最終桁	=	最終桁
下方向	最終行	=	最終行

ビューポートをブランクにし、即座に第2ステップへと移ります。

2. 一行、あるいは、一桁の文字列をビューポートへ表示し、**MarqueeUnitWait**ミリ秒待ちます。これをビューポートが一杯になるまで繰り返します。

3. マーキー方向に一行、あるいは、一桁分だけビューポートに対応するウィンドウを移動し、ビューポートをリフレッシュします。その後、**MarqueeUnitWait**ミリ秒間ウェイトします。最終行、あるいは、最終桁がビューポート内でスクロールされるまで繰り返します（この場合、**MarqueeUnitWait**ミリ秒間のウェイトは省かれます）。

4. **MarqueeRepeatWait**ミリ秒間ウェイトします。その後、第一ステップへ戻ります。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	無効な値が設定されました。あるいは、ウィンドウ番号0に対して行われました。

参照　**MarqueeType** プロパティ、**MarqueeUnitWait** プロパティ、**MarqueeRepeatWait** プロパティ

例 1　　ウォーク型.

- 2行20桁のラインディスプレイとします。
- VB アプリケーションがLDと名付けられたラインディスプレイオブジェクトを持っているとします。
- そのアプリケーションが次を実行したとします。

```
LD.CreateWindow(0, 3, 2, 3, 2, 5)
LD.DisplayText("0123456789", DISP_DT_NORMAL)
```

ウィンドウ内容は次のようになっています。

	0	1	2	3	4
0	0	1	2	3	4
1	5	6	7	8	9

また、ディスプレイは次のようになっています（ほかのウィンドウはすべてブランク表示とします）。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0				0	1	2														
1				5	6	7														

アプリケーションが次のシーケンスを行ったとすると、

```
LD.MarqueeType = DISP_MT_INIT
LD.MarqueeFormat = DISP_MF_WALK
LD.DisplayTextAt(0, 4, "AB", DISP_DT_NORMAL)
```

ビューポートには変化がなく、ウィンドウは次のようになります。

	0	1	2	3	4
0	0	1	2	3	A
1	B	6	7	8	9

アプリケーションが次を行うと、

```
LD.MarqueeType = DISP_MT_LEFT
```

ウィンドウには変化がなく、ビューポートは次のようになります。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0						0														
1						B														

MarqueeUnitWaitミリ秒後、ビューポートは次のようになります。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0					0	1														
1					B	6														

MarqueeUnitWaitミリ秒後、ビューポートは次のようになります。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0				0	1	2														
1				B	6	7														

MarqueeUnitWaitミリ秒後、ビューポートは次のようになります。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0				1	2	3														
1				6	7	8														

MarqueeUnitWaitミリ秒後、ビューポートは次のようになります。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0				2	3	A														
1				7	8	9														

マーキーはウィンドウの最後までスクロールします。
MarqueeRepeatWaitミリ秒後、マーキー表示は次のような表示に戻って繰り返されます。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0						0														
1						B														

例 2

プレース型

- 2行20桁のラインディスプレイとします。
- VB アプリケーションがLDと名付けられたラインディスプレイオブジェクトを持っているとします。
- そのアプリケーションが次を実行したとします。

```
LD.CreateWindow(0, 3, 2, 3, 2, 5)
LD.DisplayText("0123456789", DISP_DT_NORMAL)
```

ウィンドウ内容は次のようになっています。

	0	1	2	3	4
0	0	1	2	3	4
1	5	6	7	8	9

また、ディスプレイは次のようになっています（ほかのウィンドウはすべてブランク表示とします）。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0				0	1	2														
1				5	6	7														

アプリケーションが次のシーケンスを行ったとすると、
　LD.MarqueeType = DISP_MT_INIT
　LD.MarqueeFormat = DISP_MF_PLACE
　LD.DisplayTextAt(0, 4, “AB”, DISP_DT_NORMAL)
ビューポートには変化がなく、ウィンドウは次のようになります。

	0	1	2	3	4
0	0	1	2	3	A
1	B	6	7	8	9

アプリケーションが次を行うと、
　LD.MarqueeType = DISP_MT_LEFT
ウィンドウには変化がなく、ビューポートは次のようになります。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0				0																
1				B																

MarqueeUnitWaitミリ秒後、ビューポートは次のようになります。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0				0	1															
1				B	6															

MarqueeUnitWaitミリ秒後、ビューポートは次のようになります。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0				0	1	2														
1				B	6	7														

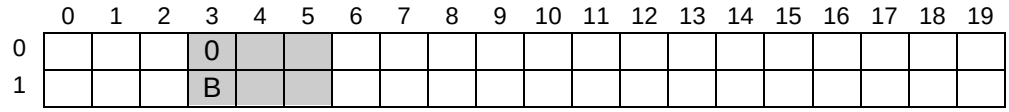
ウィンドウの終端に達してからは、このマーキーの動きはウォーク型
と同じ動作内容となります。
MarqueeUnitWaitミリ秒後、ビューポートは次のようになります。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0				1	2	3														
1				6	7	8														

MarqueeUnitWaitミリ秒後、ビューポートは次のようになります。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0				2	3	A														
1				7	8	9														

マーキーはウィンドウの最後までスクロールします。
MarqueeRepeatWaitミリ秒後、マーキー表示は次のような表示に戻って繰り返されます。



MarqueeRepeatWait プロパティ　R/W

形式　　**LONG MarqueeRepeatWait;**

説明　　マーキースクロールを繰り返す際の待ち時間が設定されています。

この待ち時間は、ミリ秒単位で設定されます。（タイマー精度により誤差が生じる場合があります。）

このプロパティは、**Open**メソッド、および、**CreateWindow**メソッドにより0に初期化されます。また、**CurrentWindow**プロパティの設定、または、**DestroyWindow**メソッドの呼び出しにより更新されます。

マーキーモードでないときには **MarqueeRepeatWait**プロパティは使用されません。

戻り値　　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

参照　　**MarqueeType**プロパティ、**MarqueeFormat**プロパティ、**MarqueeUnitWait**プロパティ

MarqueeType プロパティ　R/W

形式　　LONG MarqueeType;

説明　　カレントウィンドウのマーキースクロールのタイプが設定されています。この値がDISP_MT_NONE以外の場合、ウィンドウはマーキーモードであると表現します。

値	説明
DISP_MT_NONE	スクロールをしません。
DISP_MT_INIT	マーキー初期化モード。**MarqueeType**プロパティがほかの値へ設定されるまでウィンドウへの書き込みは、ビューポートへ影響しません。
DISP_MT_UP	上方向にスクロールします。**Rows**プロパティがウィンドウ作成時の**CreateWindow**メソッドのパラメータ*Height*以下の場合、または、**CapVMarquee**プロパティがTRUEでない場合には無効です。
DISP_MT_DOWN	下方向にスクロールします。**Rows**プロパティがウィンドウ作成時の**CreateWindow**メソッドのパラメータ*Height*以下の場合、または、**CapVMarquee**プロパティがTRUEでない場合には無効です。
DISP_MT_LEFT	左方向にスクロールします。**Columns**プロパティがウィンドウ作成時の**CreateWindow**メソッドのパラメータ*Width*以下の場合、または、**CapHMarquee**プロパティがTRUEでない場合には無効です。
DISP_MT_RIGHT	右方向にスクロールします。**Columns**プロパティがウィンドウ作成時の**CreateWindow**メソッドのパラメータ*Width*以下の場合、または、**CapHMarquee**プロパティがTRUEでない場合には無効です。

マーキーは、**MarqueeType**プロパティにDISP_MT_INITを設定しマーキー初期化モードに入り、**ClearText**メソッド、**DisplayText**メソッド、**DisplayTextAt**メソッドを使用することにより、初期化されます。その後、**MarqueeType**プロパティにスクロールさせる適切な値を設定することにより、マーキー実行モードとなります。

マーキー実行モード時にDISP_MT_NONEを設定すると、マーキースクロールはその場で止まります。また、その後、マーキー実行モードにすると現在の表示状態から引き続いて継続されます。

マーキー実行モードからマーキー初期化モードへ変更した場合には、マーキーは終了します。そして、ウィンドウ内容を変更しその新しいデータ内容でマーキーを再開させるために、マーキー実行モードへと設定します。

ウィンドウ番号0（デバイスウィンドウ）の場合には、**MarqueeType**プロパティは常にDISP_MT_NONEとなっています。

Openメソッド、または、**CreateWindow**メソッドの呼び出しにより**MarqueeType**プロパティは、DISP_MT_NONEに初期化されます。また、**CurrentWindow**プロパティの設定、または、**DestroyWindow**メソッドの呼び出しにより更新されます。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。または、ウィンドウ番号0が指定されています。

参照　**MarqueeUnitWait**プロパティ、**MarqueeFormat**プロパティ、**MarqueeRepeatWait**プロパティ

MarqueeUnitWait プロパティ　R/W

形式　LONG MarqueeUnitWait;

説明　ウィンドウ内で桁単位、あるいは行単位でマーキースクロールする際の表示遅延時間が設定されています。

この表示遅延時間は、ミリ秒単位で設定されます。（タイマー精度により誤差が生じる場合があります。）

MarqueeTypeプロパティがDISP_MT_NONEの場合には、**MarqueeUnitWait**プロパティは使用されません。

このプロパティは、**Open**メソッド、および、**CreateWindow**メソッドにより0に初期化されます。また、**CurrentWindow**プロパティを設定、または、**DestroyWindow**メソッドの呼び出しにより更新されます。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに設定されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

参照　**MarqueeType**プロパティ、 **MarqueeFormat**プロパティ、**MarqueeRepeatWait**プロパティ

MaximumX プロパティ　　第1.7版にて追加

形式　LONG MaximumX;

説明　ビットマップをサポートしていない場合は0です。それ以外ではデバイスがサポートしている水平方向のピクセル数の最大値を65536未満で示します。**MaximumX**を**DeviceColumns**で割ることで、各文字に必要なピクセル数が得られます。

このプロパティは、**Open**メソッドにより初期化されます。**ScreenMode**プロパティの変更で更新されます。

参照　DeviceColumnsプロパティ、ScreenModeプロパティ、MaximumYプロパティ

MaximumY プロパティ　　第1.7版にて追加

形式　LONG MaximumY;

説明　ビットマップをサポートしていない場合は0です。それ以外ではデバイスがサポートしている垂直方向のピクセル数の最大値を65536未満で示します。**MaximumY**を**DeviceRows**で割ることで、各文字に必要なピクセル数が得られます。

このプロパティは、**Open**メソッドにより初期化されます。**ScreenMode**プロパティの変更で更新されます。

参照　DeviceRowsプロパティ、ScreenModeプロパティ、MaximumXプロパティ

Rows プロパティ

形式　　LONG Rows;

説明　　カレントウィンドウの行数を示します。

ウィンドウ番号0の場合には、**Rows**プロパティは**DeviceRows**プロパティと同じです。その他のウィンドウに関しては、**DeviceRows**プロパティより小さい場合もあるし大きい場合もあります。

Openメソッドによりこのプロパティは**DeviceRows**プロパティと同じ値に初期化されます。また、**CurrentWindow**プロパティが設定されたとき、**CreateWindow**メソッド、**DestroyWindow**メソッドの呼び出しによって更新されます。

参照　　Columnsプロパティ

ScreenMode プロパティ　R/W　　第1.7版にて追加

形式　　LONG ScreenMode;

説明　　デバイスのスクリーンモードを示します。**CapScreenMode**プロパティがFALSEならば、0だけが有効です。**CapScreenMode**プロパティがTRUEならば、この値はスクリーンモードの**ScreenModeList**上のインデックス値です。例えば：

0=デフォルトスクリーンモード

1= **ScreenModeList**の最初のモード

2= **ScreenModeList**の2番目のモード

Note:このプロパティは、**Open & Claim**済みで、かつイネーブルでは**ない**状態でのみ更新することができます。

ScreenModeプロパティの変更により、**DeviceColumns**および**DeviceRows**プロパティも新しいスクリーンサイズに変わります。またデバイスによっては**MaximumX**と**MaximumY**プロパティも行数と桁数にもとづく要求で物理的なピクセル数が変わることで変更されます。例えば、ディスプレイが物理的に48x256のピクセルを持ち、2x20、4x32、5x32をサポートしているならば以下のようになります。

Mode	Pixels per Row	Pixels per Column	MaximumY	MaximumX	Unused Vertical Pixels	Unused Horizontal Pixels
2x20	24	12	48	240	0	16
4x32	12	8	48	256	0	0
5x32	8	8	40	256	8	0

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CapScreenModeプロパティ**、**DeviceColumnsプロパティ**、**DeviceRowsプロパティ**、**MaximumXプロパティ**、**MaximumYプロパティ**、**ScreenModeListプロパティ**

ScreenModeList プロパティ　　第1.7版にて追加

形式　**BSTR ScreenModeList;**

説明　デバイスがサポートしている行数桁数の組合せをカンマ区切りのリストで示します。

CapScreenModeプロパティがFALSEならば、1組だけがリストされます。例えばデバイスが2行20桁のみをサポートしているならば、このプロパティには”2x20”が設定されます。

デバイスが2行20桁、4行32桁、5行32桁の各モードで動作可能な場合、このプロパティには、”2x20,4x32,5x32”が設定されます。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CapScreenModeプロパティ**、**ScreenModeプロパティ**

メソッド

ClearDescriptors メソッド

形式　LONG ClearDescriptors ();

説明　すべてのディスクリプターを消灯させます。

CapDescriptorsプロパティがFALSEの場合には、このメソッドは機能しません。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	正常終了です。
OPOS_E_ILLEGAL	デバイスは、ディスクリプターをサポートしていません。
その他	ResultCodeの項目を参照してください。

参照　SetDescriptorメソッド、DeviceDescriptorsプロパティ

ClearText メソッド　　第1.7版にて更新

形式　　LONG ClearText ();

説明　　カレントウィンドウ上のすべての文字を消去します。**CursorRow**プロパティと**CursorColumn**プロパティは、0へ設定されます。ビューポートはウィンドウの先頭部分へ移動します。ウィンドウ内のすべてのビットマップを消去します。

イミディエイトモード、テレタイプモードでは、ビューポートは即時にクリアーされます。

マーキー初期化モードでは、ビューポートへの影響はありません。

マーキー実行モード中では、このメソッドの使用は禁止されています。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納ます。

値	説明
OPOS_SUCCESS	正常終了です。
OPOS_E_ILLEGAL	マーキー実行中です。
その他	**ResultCode**の項目を参照してください。

参照　　**DisplayText**メソッド

CreateWindow メソッド　　第1.6版にて更新

形式　LONG CreateWindow (LONG *Row*, LONG *Column*, LONG *Height*,LONG *Width*,LONG *WindowHeight*, LONG *WindowWidth*);

パラメータ	説明
Row	ビューポートの開始デバイス行位置。
Column	ビューポートの開始デバイス桁位置。
Height	ビューポートのデバイス行数。
Width	ビューポートのデバイス列数。
WindowHeight	ウィンドウの論理的な行数。
WindowWidth	ウィンドウの論理的な列数。

説明　*Row*、*Column*、*Height*、*Width*パラメータで示されるディスプレイの物理位置にビューポートを作成します。ウィンドウのサイズは、*WindowHeight*、*WindowWidth*パラメータによって指定されます。有効なウィンドウの行範囲は(0)から(*Height*-1)で、桁範囲は(0)から(*Width*-1)です。

ウィンドウは、ディスプレイ上に割り当てられた物理的なビューポートサイズ以上の大きさを持っていなければなりません。

ウィンドウは、一つの方向においてビューポートよりも大きくすることができます。ウィンドウのスクロールプロパティ、すなわち、**MarqueeType**プロパティ、**MarqueeFormat**プロパティ、**MarqueeUnitWait**プロパティ、および、**MarqueeRepeatWait**プロパティを使って、このようなウィンドウをマーキー風に連続してスクロールすることができます。

ウィンドウが作成されると、**CreateWindow**メソッドはこのウィンドウに割り当てられたウィンドウ番号を**CurrentWindow**プロパティに設定します。以下のプロパティがそれぞれのウィンドウごとに管理され、また、それぞれの値に初期化されます。

プロパティ	値
Rows	*WindowHeight*
Columns	*WindowWidth*
CursorRow	0
CursorColumn	0

CursorType	DISP_CT_NONE(**CapCursorType**プロパティにDISP_CCT_FIXEDが設定されている場合には、対応する適切な値)
CursorUpdate	TRUE
MarqueeType	DISP_MT_NONE
MarqueeFormat	DISP_MF_WALK
MarqueeUnitWait	0
MarqueeRepeatWait	0
InterCharacterWait	0

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	正常終了です。
OPOS_E_ILLEGAL	無効なパラメータが渡されました。あるいは、すべてのウィンドウが使用中です。
その他	**ResultCode**の項目を参照してください。

参照　**DestroyWindow**メソッド、**CurrentWindow**プロパティ

DefineGlyph メソッド　　第1.6版にて追加

形式　LONG DefineGlyph (LONG *GlyphCode*, BSTR *Glyph*);

パラメータ	説明
GlyphCode	Glyphを定義する文字のコード。
Glyph	Glyphを定義するデータ。このデータのフォーマットは **BinaryConversion**プロパティに依存しています。詳細は**BinaryConversion**プロパティを参照してください。

説明　Glyph文字を定義します。

Glyphは各ピクセルのビット表現をバイト単位に詰め込み、それぞれの行をバイトで区切って定義されます。

GlyphWidthに基づいて、8ビットで1バイトとし、各行のために最小のバイト数が使われます。各行のバイト列は左から右に送られます。もし１行あたり１バイト以上必要な場合は、一番左側のバイトが最初になります。1バイトの中の最下位ビットは一番右のピクセルを表します。ピクセルを表していないビットがあれば、それは最上位ビット側になり、それらの値は無視されます。各行は上から下の順に送られます。

幅10ピクセルのGlyphは、一番左側の2つのピクセルをそれぞれ最初のバイトの1と0のビットに表します。残りの8つのピクセルが2番目のバイトで表されます。

文字全体を定義するのに充分な行が無ければなりません。現在表示されている文字のGlyph定義を変更した場合、現在の表示を変更するか、次に表示された時に変更するかはハードウェア定義となります。

例：1文字が縦7横5ピクセルの場合

ビット位置 76543210	バイト順	16進数値
.*...	0	08
..*..	1	04
...	2	12
.*..*	3	09
..*..	4	04
...*.	5	02
....*	6	01

例：1文字が縦16横12ピクセルの場合

ビット位置 111111 5432109876543210	バイト順	16進数値
....	0,1	00 00
.....*......	2,3	00 40
....***.....	4,5	00 E0
...**.**....	6,7	01 B0
..**...**...	8,9	03 18
..**...**...	10,11	03 18
..*******...	12,13	03 F8
..*******...	14,15	03 F8
..**...**...	16,17	03 18
..**...**...	18,19	03 18
..**...**...	20,21	03 18
....	22,23	00 00
....	24,25	00 00
....	26,27	00 00
....	28,29	00 00
....	30,31	00 00

CapCustomGlyphプロパティがFALSEの場合には、このメソッドは機能しません。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	正常終了です。
OPOS_E_ILLEGAL	デバイスは、Glyph定義をサポートしていません。あるいは、無効な*GlyphCode*が渡されました。
その他	**ResultCode**の項目を参照してください。

参照　**CapCustomGlyphプロパティ、CustomGlyphListプロパティ、GlyphWidthプロパティ、GlyphHeightプロパティ**

DestroyWindow メソッド

形式　LONG DestroyWindow ();

説明　カレントウィンドウを削除します。表示されている内容は変化しません。

CurrentWindowプロパティは、ウィンドウ番号0に設定されます。デバイスウィンドウと関連するプロパティ内容は更新されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	正常終了です。
OPOS_E_ILLEGAL	ウィンドウ番号0が設定されています。このウィンドウを削除することはできません。
その他	**ResultCode**の項目を参照してください。

参照　**CreateWindow**メソッド、**CurrentWindow**プロパティ

DisplayBitmap メソッド　　第1.7版にて追加

形式　　LONG DisplayBitmap (BSTR *FileName*, LONG *Width*, LONG *AlignmentX*, LONG *AlignmentY*);

パラメータ	説明
FileName	Windowsビットマップファイル名。ファイルは圧縮されたフォーマットではいけません。
Width	ビットマップの表示幅。下記の値を参照。
AlignmentX	ビットマップの水平方向位置。下記の値を参照。
AlignmentY	ビットマップの垂直方向位置。下記の値を参照。

*Width*パラメータの値は、

値	意味
DISP_BM_ASIS	ラインディスプレイのドット当たり1ビットマップピクセルで、ビットマップを表示します。
その他	ビットマップ幅。

*AlignmentX*パラメータの値は、

値	意味
DISP_BM_LEFT	ビットマップの左端を**CursorColumn**で示す現在位置の文字の左端に合わせる。
DISP _BM_CENTER	**CursorColumn**で示す現在位置の文字からの水平方向のセンタリング。
DISP _BM_RIGHT	ビットマップの右端を**CursorColumn**で示す現在位置の文字の右端に合わせる。
その他	ビットマップ表示を開始するウィンドウ左端からの距離。

*AlignmentY*パラメータの値は、

値	意味
DISP_BM_TOP	ビットマップの上端を**CursorRow**で示す現在位置の文字の上端に合わせる。

DISP _BM_CENTER　**CursorRow**で示す現在位置の文字からの垂直方向のセンタリング。

DISP _BM_BOTTOM　ビットマップの下端を**CursorRow**で示す現在位置の文字の下端に合わせる。

その他　ビットマップ表示を開始するウィンドウ上端からの距離。

説明　ラインディスプレイにビットマップを表示する時にこのメソッドを呼び出します。ビットマップはカレントウィンドウのビューポートに表示されます。

AlignmentX、*AlignmentY*パラメータにDISP_BM_XXXが指定されていたならば、**CursorRow**と**CursorColumn**で指定された文字位置から表示されます。さらに、**CursorUpdate**がTRUEならば、**CursorRow**と**CursorColumn**は表示されたビットマップに続く最初の文字位置に更新されます。

ビットマップが行桁ちょうどの領域を占めていないのであれば、それらの文字位置の残りの領域はバックグラウンドカラーで埋められます。言い換えると、サービスオブジェクトはビットマップを表示する前に、そのビットマップが表示される全文字領域をバックグラウンドカラーで埋めておきます。

ビットマップ表示には以下の制約があります。

- ビットマップ表示はイミディエートモード時のみできます。
- ウィンドウサイズはそのウィンドウのビューポートサイズと合致していなければなりません。
- ビットマップは、パラメータを適用後でもウィンドウに表示可能でなければなりません。例えば、*AlignmentY*がウィンドウの下端近くに指定され、ビットマップの（必要ならば変換後の）高さが*AlignmentY*からウィンドウの下端までの長さを超える場合、ビットマップは表示されません。

*Width*パラメータはビットマップの変換をコントロールします。もし、*Width*がDISP_BM_ASISならば、変換は実行されません。ビットマップはラインディスプレイドット当たり1ビットマップピクセルで表示されます。このオプションの長所は、以下のことです。

- 高いパフォーマンスのビットマップ表示ができます。
- ビットマップは、水平方向と垂直方向のドット単位がラインディスプレイ固有のアスペクト比に調整され、良好な表示が得られます。

もし*Width*が0でなければ、ビットマップは、その幅が指定された幅でそのアスペクト比が変化しないように、拡大されたり圧縮されたりすることにより、変換されます。このオプションの長所は以下のことです。

- いろいろなラインディスプレイに合うように、ビットマップの大きさを作ります。
- ビットマップのアスペクト比の比率を維持します。

欠点は、

- 変換しないデータよりもパフォーマンスが落ちます。
- オリジナルのビットマップでは良好ないくつかの行とイメージが、いくらかギザギザとなるかもしれません。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	以下のエラーの内の一つが起こりました。 ・デバイスがビットマップ表示をサポートしていません。 ・*Width*が、不正な値であるか、大きすぎます。 ・*AlignmentX*、*AlignmentY*が、不正な値であるか、 大きすぎます。 ・ウィンドウがイミディエートモードではありません。 ・ウィンドウとビューポートのサイズが合っていません。 ・指定された位置から表示するにはビットマップが大きすぎます。
OPOS_E_NOEXIST	*FileName*が見つかりませんでした。
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_EDISP_TOOBIG: ビットマップが、変換無しで表示するにはその幅が広すぎるか、あるいは変換するには大きすぎます。

	ResultCodeExtended = OPOS_EDISP_BADFORMAT: 指定されたファイルが、ビットマップファイルでないか、あるいはサポートされていないフォーマットです。
その他	**ResultCode**の項目を参照してください。

参照　**CapBitmap**プロパティ、**CursorColumn**プロパティ、**CursorRow**プロパティ、**CursorUpdate**プロパティ

DisplayText メソッド　　第1.7版にて更新

形式　LONG DisplayText (BSTR *Data*, LONG *Attribute*);

パラメータ	説明
Data	表示する文字列です。このデータのフォーマットは **BinaryConversion**プロパティに依存しています。詳細は**BinaryConversion**プロパティを参照してください。
Attribute	表示属性。DISP_DT_NORMAL、DISP_DT_BLINK、DISP_DT_REVERSE、あるいは、DISP_DT_BLINK_REVERSEのいずれかを指定します。

説明　**CursorRow**と**CursorColumn**で指定される位置から*Data*パラメータの文字列を表示します。

行末に達した場合には、文字表示は次行に続けられます。ウィンドウの末尾になっても表示すべき文字がある場合には、ウィンドウは1行だけ上へスクロールアップします。**CursorUpdate**プロパティがTRUEの場合には、**CursorRow**プロパティと**CursorColumn**プロパティは引数*Data*の末尾文字の次を指すように更新されます。

備考： *Data*の末尾文字が行末に表示される場合には、スクロールは行われません。この場合、**CursorUpdate**プロパティがTRUEの場合には、**CursorRow**プロパティは末尾文字が表示されている行に設定され、**CursorColumn**プロパティは実行前の**Columns**プロパティ（つまり、その行の末尾文字のひとつ次）に設定されます。

この規定は、文字が末尾位置に書かれても表示はスクロールされないということを保証します。その代わり、コントロールオブジェクトは別の文字が書かれるまでウィンドウのスクロールを遅らせることができます。

DisplayText メソッド（および、**DisplayTextAt** メソッド）の動作はモードによってことなります。

イミディエイトモード（MarqueeType = DISP_MT_NONE かつ InterCharacterWait = 0）： ウィンドウとビューポートを即座に更新します。

テレタイプモード（ MarqueeType = DISP_MT_NONE かつ InterCharacterWait not = 0）： *Data*はキューに管理されます。キュー管理された要求は順に処理され（典型的にはOPOSコントロール内のほかのスレッドにより）InterCharacterWait ミリ秒ごとに一文字ずつビューポートに送られることによりウィンドウとビューポートは更新されます。

マーキー初期化モード（MarqueeType = DISP_MT_INIT）： ウィンドウを更新するのみでビューポートは更新されません。

マーキー実行モード（MarqueeType not = DISP_MT_INIT）： 無効

CapBlinkプロパティがDISP_CB_NOBLINKの場合、*Attribute*がDISP_DT_BLINKのときは無視され、*Attribute*がDISP_DT_BLINK_REVERSEのときはDISP_DT_REVERSEとして扱われます。DISP_CB_BLINKALLの場合には、文字をブリンクするように設定されたときにすべての表示がブリンクします。

DISP_CB_BLINKEACHの場合には、ブリンク属性を持って表示された文字のみがブリンクします。

CapReverseプロパティがDISP_CR_NONEの場合、*Attribute*がDISP_DT_REVERSEのときは無視され、*Attribute*がDISP_DT_BLINK_REVERSEのときはDISP_DT_BLINKとして扱われます。DISP_CR_REVERSEALLの場合には、文字を反転するように設定されたときにすべての表示が反転します。

DISP_CB_REVERSEEACHの場合には、反転属性を持って表示された文字のみが反転します。

*Attribute*は開始時のブリンクと反転の属性を設定します。第1.7版以後は、*Data*の中でエスケープシーケンスを使ってこれらの属性を設定解除できます。

*Data*に含まれる特殊文字には次のようなものがあります。

値	説明
Carriage Return (10進数の13)	次の文字の出力位置を現在行の先頭へ移動します。

Newline / Line Feed (10進数の10)
次の文字の出力位置を次行の先頭へ移動します。現在の行がウィンドウの最終行の場合には、ウィンドウはスクロールします。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	正常終了です。
OPOS_E_ILLEGAL	*Attribute*が無効です。あるいは、マーキー実行中です。
その他	**ResultCode**の項目を参照してください。

参照　**DisplayTextAt**メソッド、**ClearText**メソッド、**CapBlink**プロパティ、**CapReverse**プロパティ、**InterCharacterWait**プロパティ

DisplayTextAt メソッド　　第1.6版にて更新

形式　LONG DisplayTextAt (LONG *Row*, LONG *Column*, BSTR *Data*, LONG *Attribute*);

パラメータ	説明
Row	テキストの開始行。
Column	テキストの開始列。
Data	表示文字列。このデータのフォーマットは**BinaryConversion**プロパティに依存しています。詳細は**BinaryConversion**プロパティを参照してください。
Attribute	表示属性。DISP_DT_NORMAL、DISP_DT_BLINK、DISP_DT_REVERSE、あるいは、DISP_DT_BLINK_REVERSEのいずれかを指定します。

説明　*Data*パラメータの文字列は、*Row*パラメータと*Column*パラメータで指定される位置から表示されます。

このメソッドは、**CursorRow**プロパティに*Row*パラメータを、**CursorColumn**プロパティに*Column*パラメータを設定し、**DisplayText**メソッドを呼び出すことと同じ結果が得られます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	正常終了です。
OPOS_E_ILLEGAL	*Row*あるいは*Column*が範囲外、*Attribute*が無効、あるいは、マーキー実行モードです。
その他	**ResultCode**の項目を参照してください。

参照　**DisplayText**メソッド、**ClearText**メソッド、**CapBlink**プロパティ、**CapReverse**プロパティ、**InterCharacterWait**プロパティ

ReadCharacterAtCursor メソッド　第1.6版にて追加

形式　LONG ReadCharacterAtCursor (LONG* *pChar*);

パラメータ	説明
pChar	ディスプレイから読み込んだ文字を格納する領域。

説明　現在のカーソル位置に表示されている文字を読み込みます。

CapReadBackプロパティがDISP_CRB_NONEの場合には、このメソッドは機能しません。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	正常終了です。
OPOS_E_ILLEGAL	デバイスは、文字読み込みをサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照　**CapReadBack**プロパティ

RefreshWindow メソッド

形式　　LONG RefreshWindow (LONG *Window*);

*Window*パラメータには、リフレッシュするウィンドウ番号を指定します。

説明　　カレントウィンドウを*Window*パラメータで指定されるウィンドウにし、このウィンドウの以前表示されていた内容が再表示されます。ビューポートへのウィンドウ対応領域とウィンドウのカーソル位置には影響がありません。

このメソッドは、ほかのウィンドウによって表示領域上を上書きされた後、現在のウィンドウ内容に戻す場合に使われます。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	正常終了です。
OPOS_E_ILLEGAL	*Window*が**DeviceWindows**プロパティ値以上か作成されていません。あるいは、マーキー実行モードです。
その他	**ResultCode**の項目を参照してください。

ScrollText メソッド　　第1.7版にて更新

形式　LONG ScrollText (LONG *Direction*, LONG *Units*);

*Direction*パラメータには、スクロールの方向を設定します。この値は、以下のいずれかの値です。

値	説明
DISP_ST_UP	上スクロール。
DISP_ST_DOWN	下スクロール。
DISP_ST_LEFT	左スクロール。
DISP_ST_RIGHT	右スクロール。

*Units*パラメータには、スクロールする行数、あるいは、桁数を設定します。

説明　カレントウィンドウをスクロールさせます。また、スクロールしても、 **CursorRow**プロパティと**CursorColumn**プロパティには影響がありません。

ScrollTextメソッドは、イミディエイトモード時のみに使用されます。

スクロール方向に対してウィンドウのサイズがビューポートのサイズと等しい場合には、ウィンドウの内容はスクロールされ、最後の*Units*行目、あるいは、*Units*桁目はスペースになり、ビューポートは更新されます。ウィンドウがビットマップを含むならば、それもスクロールされます。

スクロール方向に対してウィンドウのサイズがビューポートよりも大きい場合には、ウィンドウのデータは変化しません。その代わり、ビューポートへのウィンドウの対応部分が指定方向に移動します。ウィンドウのデータは変化せず、ビューポートが更新されます。*Units*数分スクロールすることによってウィンドウデータの先頭を超える場合には、ビューポートの第―行目、あるいは第１桁目がウィンドウの第―行目、あるいは第１桁目になるようウィンドウはスクロールされます。*Units*数分スクロールすることによってウィンドウデータの末尾を超える場合には、ビューポートの最終行目、あるいは最終桁目がウィンドウの最終行目、あるいは最終桁目になるようウィンドウはスクロールされます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明

OPOS_SUCCESS	正常終了です。
OPOS_E_ILLEGAL	*Direction*が無効です。あるいは、テレタイプモードかマーキー実行モードです。
その他	**ResultCode**の項目を参照してください。

参照　**DisplayTextメソッド**

例 1

- 2行20桁のラインディスプレイを考えます。
- VB アプリケーションがLDと名付けられたラインディスプレイオブジェクトを持っているとします。
- そのアプリケーションが次を実行したとします。

```
LD.CreateWindow(0, 3, 2, 4, 2, 4)
LD.DisplayText("abcdABCD", DISP_DT_NORMAL)
```

ウィンドウ内容は次のようになっています。

	0	1	2	3
0	a	b	c	d
1	A	B	C	D

また、デバイス上のビューポートは次のようになっています。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0				a	b	c	d													
1				A	B	C	D													

次のメソッドを呼ぶと、

```
LD.ScrollText (DISP_ST_LEFT, 2)
```

ウィンドウのデータは次のようになります。

	0	1	2	3
0	c	d		
1	C	D		

また、ビューポートは次のようになります。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0				c	d															
1				C	D															

例 2

- 2行20桁のラインディスプレイを考えます。
- VB アプリケーションがLDと名付けられたラインディスプレイオブジェクトを持っているとします。
- そのアプリケーションが次を実行したとします。

```
LD.CreateWindow(0, 3, 2, 4, 2, 8)
LD.DisplayText("abcdefghABCDEFGH", DISP_DT_NORMAL)
```

ウィンドウ内容は次のようになっています。

	0	1	2	3	4	5	6	7
0	a	b	c	d	e	f	g	h
1	A	B	C	D	E	F	G	H

また、デバイス上のビューポートは次のようになっています。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0				a	b	c	d													
1				A	B	C	D													

次のメソッドを呼ぶと、

```
LD.ScrollText (DISP_ST_LEFT, 2)
```

ウィンドウのデータには変化がなく、ビューポートだけが次のようになります。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0				c	d	e	f													
1				C	D	E	F													

このとき、次のメソッドを呼ぶと、

```
LD.ScrollText (DISP_ST_UP, 1)
```

ウィンドウのデータは次のようになります。

	0	1	2	3	4	5	6	7
0	A	B	C	D	E	F	G	H
1								

また、ビューポートは次のようになります。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0				C	D	E	F													
1																				

SetBitmap メソッド　　第1.7版にて追加

形式　LONG SetBitmap (LONG *BitmapNumber*, BSTR *FileName*, LONG *Width*, LONG *AlignmentX*, LONG *AlignmentY*);

パラメータ	説明
BitmapNumber	このビットマップに割り当てられる番号。有効なビットマップ番号は1から100です。
FileName	Windowsビットマップファイル名。ファイルは圧縮されたフォーマットではいけません。 もし、空文字列が設定されれば、ビットマップは設定解除されます。
Width	ビットマップの表示幅。下記の値を参照。
AlignmentX	ビットマップの水平方向位置。下記の値を参照。
AlignmentY	ビットマップの垂直方向位置。下記の値を参照。

*Width*パラメータの値は、

値	意味
DISP_BM_ASIS	ラインディスプレイのドット当たり1ビットマップピクセルで、ビットマップを表示します。
その他	ビットマップ幅。

*AlignmentX*パラメータの値は、

値	意味
DISP_BM_LEFT	ビットマップの左端を現在位置の文字の左端に合わせる。
DISP _BM_CENTER	センタリング。
DISP _BM_RIGHT	ビットマップの右端を現在位置の文字の右端に合わせる。
その他	ビットマップ表示を開始するウィンドウ左端からの距離。

*AlignmentY*パラメータの値は、

値	意味

DISP_BM_TOP	ビットマップの上端を現在位置の文字の上端に合わせる。
DISP _BM_CENTER	センタリング。
DISP _BM_BOTTOM	ビットマップの下端を現在位置の文字の下端に合わせる。
その他	ビットマップ表示を開始するウィンドウ上端からの距離。

説明　まもなく表示するビットマップに関する情報を保存する時に呼び出します。

ビットマップは、表示データ内にビットマップ表示のエスケープシーケンスをもつ**DisplayText**あるいは**DisplayTextAt**を呼ぶことによって表示されます。ビットマップ表示のエスケープシーケンスは、宣伝、店舗ロゴやアイコンを表示するための文字列を通常含んでいます。格納したビットマップを表示するための制約は**DisplayBitmap**の**説明**を参照して下さい。1つ以上の制約を満たしていなければ、ビットマップは表示されず、メソッドは表示データの次の文字の処理を続けます。

サービスオブジェクトはよりよいパフォーマンスを与えるために、まもなく利用するビットマップをキャッシュするように選びます。ビットマップファイルとパラメータはこのメソッドにより、正確さを増します。

高頻度で使われるビットマップには小さい*BitmapNumber*(1に近い値)を割り当て、時々使用されるビットマップには大きい*BitmapNumber*を割り当ててください。サービスオブジェクトは、ビットマップを保存する場所を決めるためにこの数値を使います。可能な場合にはデバイスにダウンロードしますが、サービスオブジェクト内のメモリに保存することもあります。単純に*FileName*と関連するプロパティを記憶するだけの場合もあります。

アプリケーションは、このメソッドが呼ばれる前に、ビューポート幅、高さのようなライディスプレイウィンドウの諸元が設定されていることを保証しなければなりません。サービスオブジェクトは、現在の値に基づいて、後で表示するための準備として、ビットマップを変換する場合があります。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。

OPOS_E_ILLEGAL	以下のエラーの内の一つが起こりました。 ・*BitmapNumber*が不正な値です。 ・デバイスがビットマップ表示をサポートしていません。（**CapBitmap**がFALSE） ・*Width*が、不正な値であるか、大きすぎます。 ・*AlignmentX*、*AlignmentY*が、不正な値であるか、 大きすぎます。
OPOS_E_NOEXIST	*FileName*が見つかりませんでした。
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_EDISP_TOOBIG: ビットマップが、変換無しで表示するにはその幅が広すぎるか、あるいは変換するには大きすぎます。 **ResultCodeExtended** = OPOS_EDISP_BADFORMAT: 指定されたファイルが、ビットマップファイルでないか、あるいはサポートされていないフォーマットです。
その他	**ResultCode**の項目を参照してください。

参照　**CapBitmap**プロパティ、**DisplayBitmap**メソッド、**DisplayText**メソッド、**DisplayTextAt**メソッド

SetDescriptor メソッド

形式　LONG SetDescriptor (LONG *Descriptor*, LONG *Attribute*);

*Descriptor*パラメータには、状態を変更したいディスクリプターを指定します。その有効範囲は、0から（DeviceDescriptors-1）までです。

*Attribute*パラメータは、ディスクリプター属性を設定します。その値は、下表のいずれかの値です。

値	説明
DISP_SD_ON	ディスクリプターの点灯。
DISP_SD_BLINK	ディスクリプターのブリンク。
DISP_SD_OFF	ディスクリプターの消灯。

説明　一つのディスクリプターの状態を設定します。

CapDescriptorsプロパティがFALSEの場合には、このメソッドは機能しません。

*Descriptor*パラメータで与えられたディスクリプターの物理位置は、デバイスとそのサービスオブジェクト間において決定されます。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	正常終了です。
OPOS_E_ILLEGAL	デバイスは、ディスクリプターをサポートしていません。あるいは、無効な引数が渡されました。
その他	ResultCodeの項目を参照してください。

参照 ClearDescriptorsメソッド、DeviceDescriptorsプロパティ

補足説明

ここでは、ラインディスプレイ仕様について補足説明をします。

ウィンドウとビューポートについて

ラインディスプレイには、ウィンドウとビューポートの概念があります。ビューポートとは、デバイス上の一部分、あるいは、全領域に割り当てられる可視領域のことです。このビューポートに対して、ビューポートの大きさ以上（少なくとも同等の大きさを持つ必要がある）の仮想的な領域を対応付けることができます。この仮想的な領域がウィンドウと呼ばれ、実際に表示される文字が編集される領域となります。また、一つのビューポートに対して一つのウィンドウが定義されます。

ビューポートとウィンドウの関係には、次の3つの場合のみが定義されます。

① ビューポートとウィンドウの大きさが同等な場合

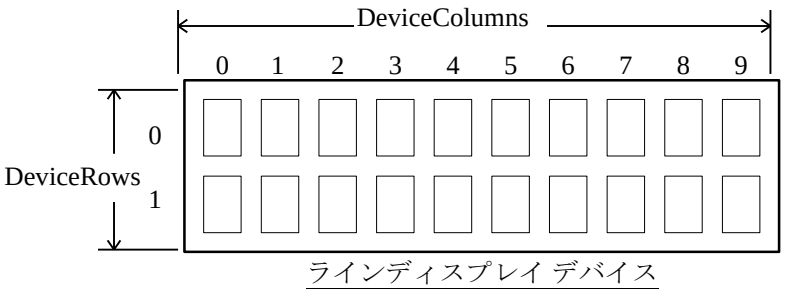
② ビューポートよりもウィンドウが水平方向に大きい場合

③ ビューポートよりもウィンドウが垂直方向に大きい場合

つまり、水平方向と垂直方向がともにビューポートよりも大きなウィンドウを作成することはできません。通常もっとも頻繁に使用されるウィンドウは、上記の①で定義されるウィンドウです。②および③で定義されるウィンドウの主な使用法は、マーキースクロールによって効果的に情報を伝達する場合などにおいてです。

◆　ウィンドウとビューポートの作成

簡単のために、2行10桁の表示領域をもつラインディスプレイ デバイスについて考えてみます（下図参照）。そのデバイスの桁数と行数は、それぞれ**DeviceColumns**プロパティと**DeviceRows**プロパティによって定義されています。



ラインディスプレイ デバイス

ここで、このデバイス上にビューポートを割り当て、ウィンドウを作成する方法は、**CreateWindow**メソッドを用いて行われます（**エラー! ブックマークが定義されていません。**ページを参照）。

このとき、作成可能なビューポートとウィンドウの関係において、**CreateWindow**メソッドに渡されるパラメーターには次の3つの関係式があります（C言語の表記法で記されています）。

① (Height == WindowHeight) && (Width == WindowWidth)

② (Height == WindowHeight) && (Width != WindowWidth)

③ (Height != WindowHeight) && (Width == WindowWidth)

上記の①～③は、先のビューポートとウィンドウの関係①～③にそれぞれ対応します。

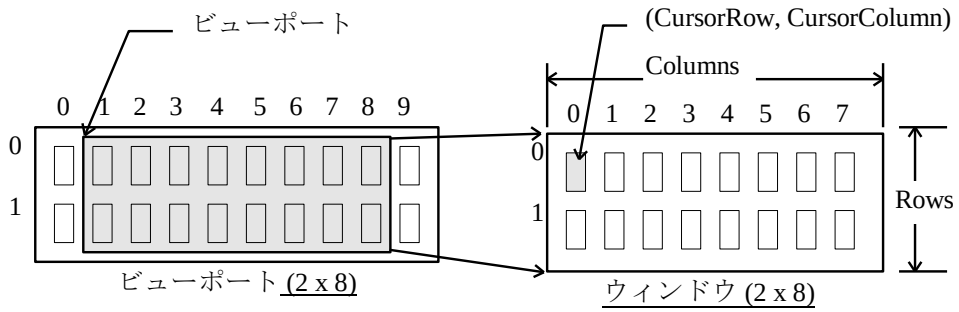
♦ 各ウィンドウ形態の仕様

① (Height == WindowHeight) && (Width == WindowWidth)の場合

これは、ビューポートとウィンドウの大きさが等しい場合で、通常頻繁に使用されるウィンドウ形態です。たとえば、前述のデバイスに対して

CreateWindow(0, 1, 2, 8, 2, 8)

を実行すると次図のような関係が作成されます。



この場合には、ウィンドウの全領域がビューポート上にマッピングされ、領域全体が可視領域となっています。

このウィンドウへの文字列の表示は、すべてウィンドウ上に編集することによって行われます。この方法は、**CursorRow**プロパティと**CursorColumn**プロパティを (0, 0) ～ (1, 7) の範囲内（つまり、ウィンドウの座標軸内）で定義し**DisplayText**メソッドを使うか、前述の範囲内のカーソル位置を指定して**DisplayTextAt**メソッドを使うかのいずれかです。このとき、指定した表示文字列中のある文字がウィンドウの末端 (1, 7) に位置した場合には、まず、上方向に一行分スクロールされ、末尾行が空白行となります。次に、引き続く文字列がその空白行へ表示されます。スクロール時にビューポートからはみ出した表示データはウィンドウ内部からも削除されます。

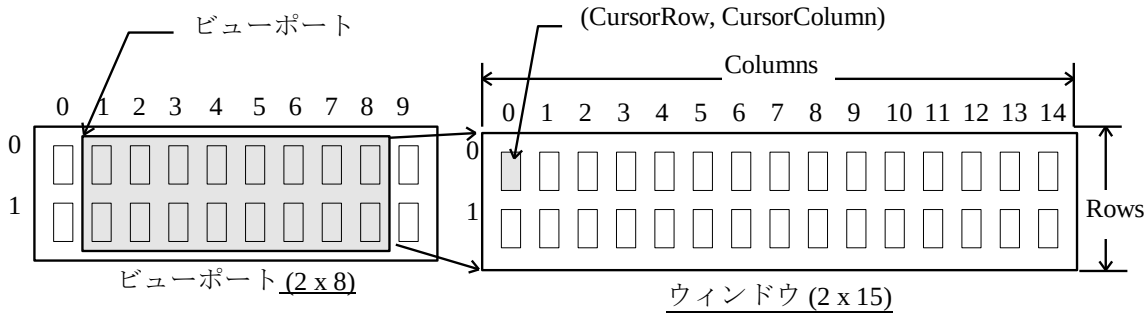
このウィンドウに表示されている文字列のスクロールは、**ScrollText**メソッドを使って行われます。**ScrollText**メソッドで指定された桁（あるいは、行）数分だけ指定された方向（上・下・左・右のいずれか）にスクロールし、このスクロールによって新しく得られた表示領域は、スペース表示となります。また、スクロールによってはみ出した文字は、ウィンドウ上からも削除されます。これは、このウィンドウに対する文字列表示でのスクロールと同等なものです。また、**CursorRow**プロパティと**CursorColumn**プロパティの各値は、このスクロールによっては更新されません。つまり、スクロール後のカーソル位置の値は、スクロール前の値と同等です。

② (Height == WindowHeight) && (Width != WindowWidth)の場合

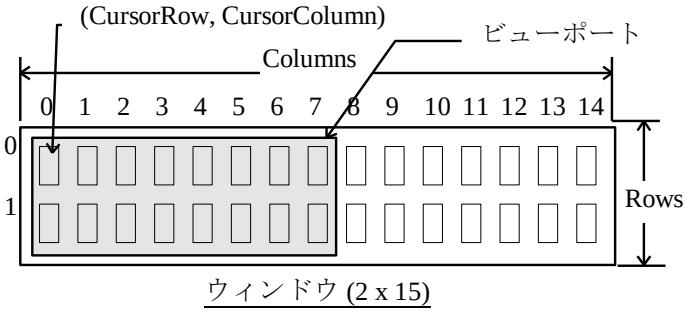
これは、ビューポートに比べてウィンドウが水平方向に大きい場合で、電光掲示板のように水平方向にメッセージを流して宣伝効果を高めるための特殊なウィンドウ形態です。たとえば、前述のデバイスに対して

CreateWindow(0, 1, 2, 8, 2, 15)

を実行すると次図のような関係が作成されます。

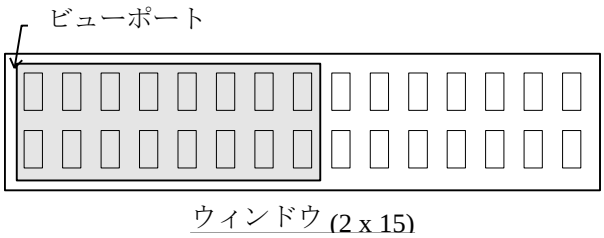


この場合には、ウィンドウの一部分の領域がビューポート上にマッピングされ、その領域のみが可視領域となっています。次図は、ウィンドウ作成直後のウィンドウとビューポートを示しています。

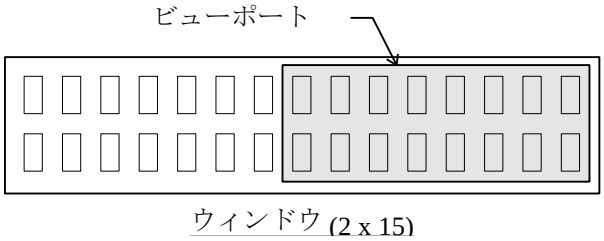


このウィンドウへの文字列の表示も、すべてウィンドウ上に編集することによって行われます。この方法は、**CursorRow**プロパティと**CursorColumn**プロパティを (0, 0) ～ (1, 14) の範囲内（つまり、ウィンドウの座標軸内）で定義し**DisplayText**メソッドを使うか、前述の範囲内のカーソル位置を指定して**DisplayTextAt**メソッドを使うかのいずれかです。また、メソッドに指定された文字列中のある文字がウィンドウの末端 (1, 14) に達した場合には、先のビューポートとウィンドウが同じ大きさの場合と同じ動作をします。まず、上方向に一行分スクロールされ、末尾行が空白行となります。次に、引き続く文字列がその空白行へ表示されます。スクロール時にビューポートからはみ出した表示データはウィンドウ内部からも削除されます。また、ビューポートに対応するウィンドウ上のデータ内容がこのときのスクロールにより更新された場合には、ビューポート上の表示内容は更新されます。

このウィンドウに表示されている文字列のスクロールは、**ScrollText**メソッドを使って行われますが、先のビューポートとウィンドウの大きさが同等な場合とは異なった動きをします。これは、**ScrollText**メソッドで指定された桁（あるいは、行）数分だけ指定された方向（上・下・左・右のいずれか）にスクロールした結果、ウィンドウの領域を超える場合には、そのウィンドウの先頭部分、あるいは、末尾部分の文字列が表示されます。このことを利用することにより、ビューポートをウィンドウの各コーナー（左上角、左下角、右上角、あるいは、右下角）に配置することができます。



たとえば、上図で右に10文字分スクロールすると次のような概念図が描かれます。

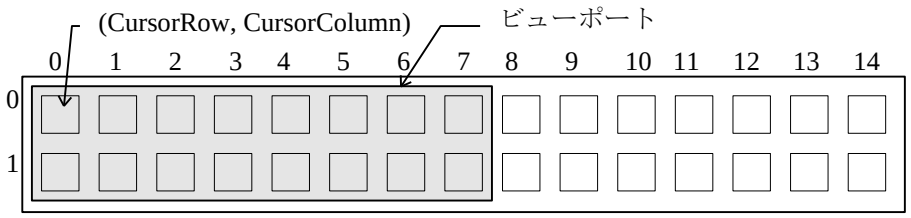


また、**CursorRow**プロパティと**CursorColumn** ロパティの各値は、このスクロールによっては更新されません。つまり、スクロール後のカーソル位置の値は、スクロール前の値と同等です。

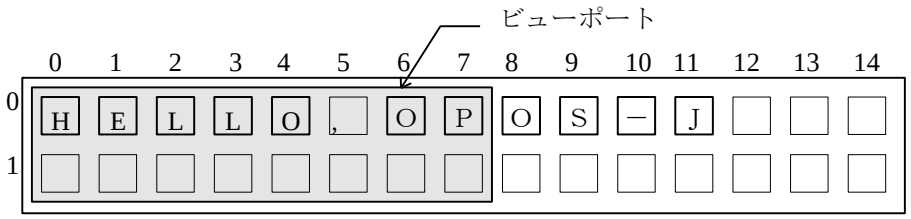
このウィンドウに対するビューポートの移動は、**ScrollText**メソッド、あるいは、**MarqueeType**プロパティの操作によってのみ行われます。つまり、**DisplayText**メソッド（あるいは、**DisplayTextAt**メソッド）での文字列表示がビューポートの領域を超えても、ビューポート内のデータは文字列に応じてスクロールされません。たとえば、図Ａで示される状態時に、

DisplayText(“HELLO,OPOS-J”, DISP_DT_NORMAL)

を実行すると、図Ｂで示される結果が得られます。



図A）　初期状態



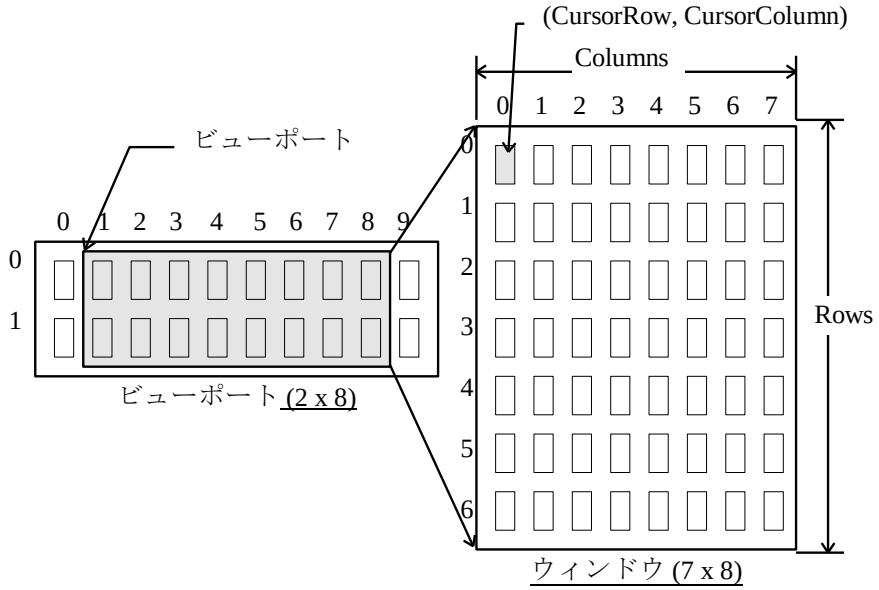
図B）　DisplayText メソッド実行後

③　(Height != WindowHeight) && (Width == WindowWidth)の場合

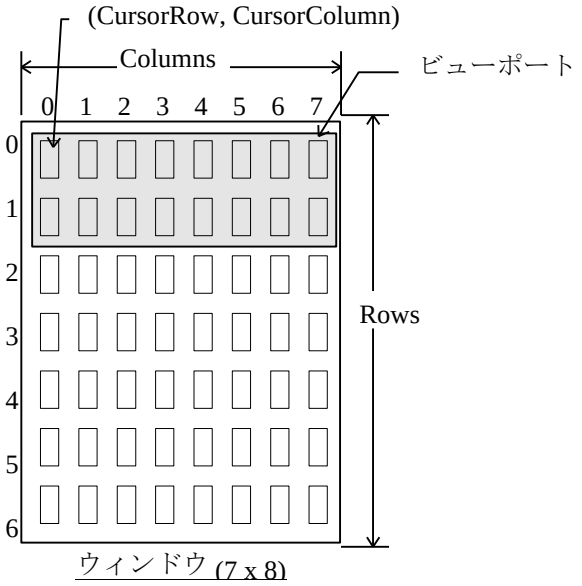
これは、ビューポートに比べてウィンドウが垂直方向に大きい場合で、電光掲示板の一つとして垂直方向にメッセージをスクロールして宣伝効果を高めるための特殊なウィンドウ形態です。たとえば、前述のデバイスに対して

CreateWindow(0, 1, 2, 8, 7, 8)

を実行すると次図のような関係が描かれます。



この場合にも、ウィンドウの一部分の領域がビューポート上にマッピングされ、その領域のみが可視領域となっています。次図は、ウィンドウ作成直後のウィンドウとビューポートを示しています。



このウィンドウへの文字列の表示、スクロール、および、ビューポートの移動に関する仕様は、前のビューポートに比べてウィンドウが水平方向に大きいウィンドウ形態のときと同様に定義されます。

マーキースクロールについて

マーキースクロールとは、電光掲示板のようにメッセージが繰り返し流されるスクロールであり、情報を効果的に伝達するためのものです。

♦　対象となるウィンドウ形態、および、制限事項

このマーキースクロールは、“ウィンドウとビューポートについて”の項目で示されるようなビューポートよりもウィンドウが大きなウィンドウ形態においてのみ定義されます。また、マーキースクロールにおいては、次のプロパティ値は参照されず、特に意味を持ちません。

・ **InterCharacterWait** プロパティ

また、マーキースクロール中（**MarqueeType**プロパティに DISP_MT_UP, DISP_MT_DOWN, DISP_MT_LEFT, あるいは、DISP_MT_RIGHT が設定されている場合）に次のメソッドを使用することは禁止されています。

・**DispalyText**メソッド

・**DispalyTextAt**メソッド

・**ClearText**メソッド

・ **ScrollText** メソッド

もし、マーキースクロール中に上記のメソッドが呼び出されると、メソッドの処理結果としてステータス OPOS_E_ILLEGAL が返されます。

◆ マーキースクロールの種類

次の2つの種類のマーキースクロールが定義されます。

① プレース（Place）型

② ウォーク（Walk）型

上記の各マーキースクロール型間においては、文字列表示のスクロール方法が異なります。また、どちらのマーキースクロール型に従ってスクロールされるかは、**MarqueeFormat** プロパティに設定されている値によって決定されます。

① プレース型マーキースクロール

プレース型のマーキースクロールの流れは、以下のルールに従ってスクロール表示されます。

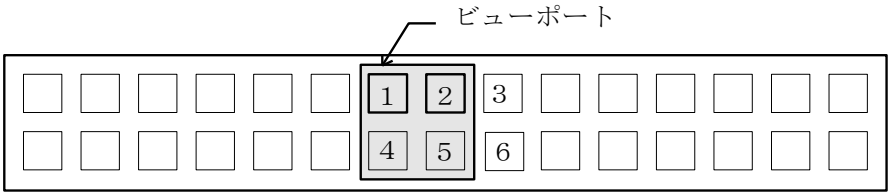
1. ビューポートはいったんブランク表示になります。**MarqueeType**プロパティが DISP_MT_LEFT と DISP_MT_UP の場合には、ウィンドウをビューポートの左上角に位置するように設定します。あるいは、**MarqueeType**プロパティが DISP_MT_RIGHT と DISP_MT_DOWN の場合には、ウィンドウをビューポートの右下角に位置するように設定します。

2. ビューポートが一杯になるまで、**MarqueeUnitWait**プロパティで示される単位時間ごとに一桁ずつ、あるいは、一行ずつビューポートに文字(列)を表示していきます。

3.ウィンドウ内の最終桁の文字、あるいは、最終行の文字列がビューポートに表示されるまで、ウィンドウをスクロールし文字(列)を表示していきます。このときには、**MarqueeUnitWait**プロパティで示される単位時間ごとに一桁ずつ、あるいは、一行ずつ文字(列)が表示されます。

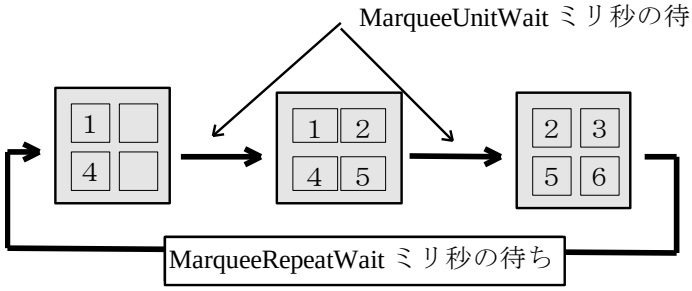
4. **MarqueeRepeatWait**プロパティで示される単位時間後、1.のシーケンスに戻ります。

たとえば、図C）のように2行3桁のウィンドウに対して2行2桁のビューポートが割り当てられているとします。



図C）

このとき、プレース型で左方向にマーキースクロールを行うように設定するとビューポートには図D）のような表示の流れが得られます。



図D）

② ウォーク型マーキースクロール

ウォーク型のマーキースクロールの流れは、以下のルールに従ってスクロール表示されます。

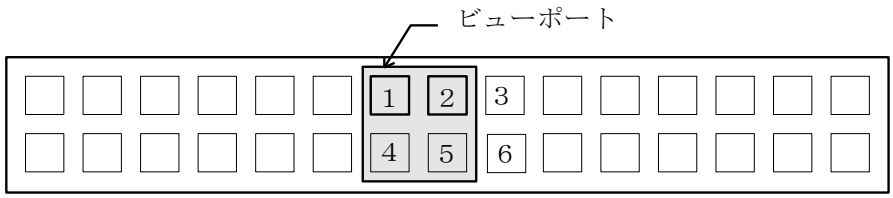
1. ビューポートを消去（スペース表示）します。次に、**MarqueeType**プロパティが DISP_MT_LEFT、DISP_MT_RIGHT、DISP_MT_UP、あるいは、DISP_MT_DOWN の各場合に対応して、ビューポートの右端の桁、左端の桁、最下位行、あるいは、最上位行に文字(列)を表示します。

2. ビューポートに表示しきれるまで、**MarqueeUnitWait**プロパティで設定される単位時間ごとに一桁ずつ、あるいは、一行ずつスクロールし文字(列)を表示していきます。

3. ウィンドウの最終桁の文字、あるいは、最終行の文字列がちょうどスクロールされるまで、ウィンドウをビューポート内でスクロールします。

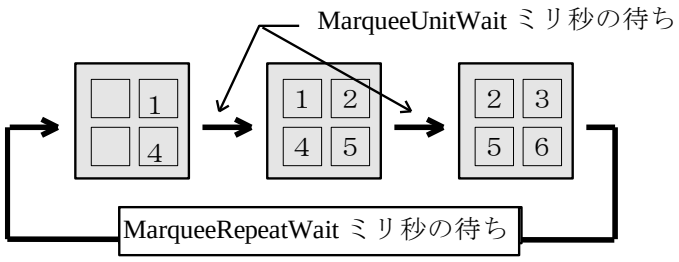
4. **MarqueeRepeatWait**プロパティで示される単位時間後、1.のシーケンスに戻ります。

たとえば、図E）のように2行3桁のウィンドウに対して2行2桁のビューポートが割り当てられているとします。



図E）

このとき、ウォーク型で左方向にマーキースクロールを行うように設定するとビューポートには図F）のような表示の流れが得られます。



図F）

♦ **MarqueeType**プロパティの設定について

マーキースクロールは、**MarqueeType**プロパティに DISP_MT_NONE 以外の適切な値を設定することによりマーキースクロールが開始され、反対に、DISP_MT_NONE を設定することによりスクロールは終了します。このときの終了状態は、 DISP_MT_NONE が設定されたときの表示状態となります。また、再度 DISP_MT_NONE 以外の適切な値を設定することによりそのときの終了状態からマーキースクロールが再開されます。

また、**MarqueeType**プロパティへの値の設定においては、以下の規則に則ります。

1. DISP_MT_INIT が設定されているときは、マーキースクロールは行われず、**DisplayText**メソッドなどを使用してウィンドウ上に文字列の設定ができます。しかし、ビューポートには何も反映されません。

2. マーキースクロール中に DISP_MT_INIT が設定された場合には、そのときの表示状態でスクロールは止まります。

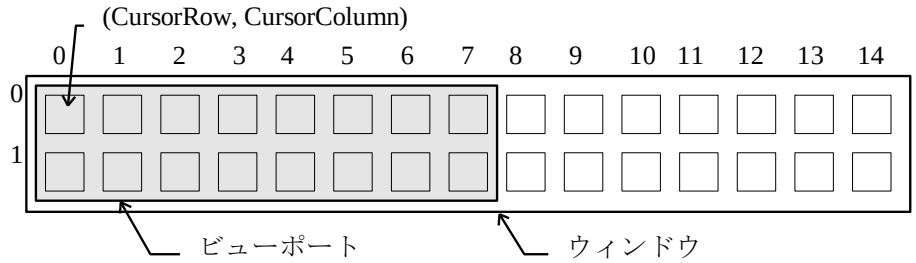
3. DISP_MT_INIT から マーキースクロールを開始する値（ DISP_MT_INIT、DISP_MT_NONE 以外の値）へと設定された場合には、ビューポートがブランク表示にされてからマーキースクロールが開始されます。

4. DISP_MT_INIT から DISP_MT_NONE へと設定された場合には、ウィンドウの先頭部分がビューポートに表示されます。

カーソル、文字列表示とスクロールの関係について

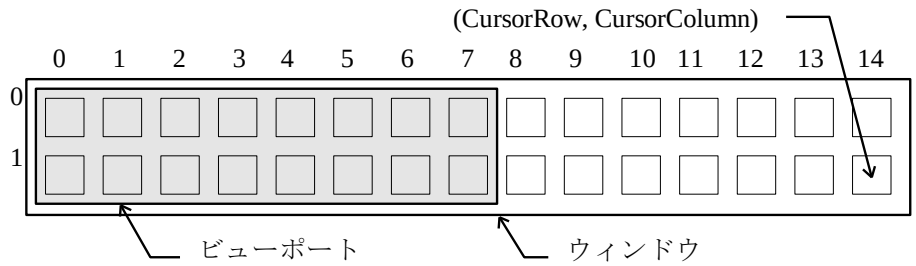
♦　カーソルの設定

カーソルとは、ウィンドウ上の各文字（半角文字単位）位置を行と桁の座標軸によって表現するプロパティであり、行位置を示す**CursorRow**プロパティと桁位置を示す**CursorColumn**プロパティにより構成されます。ウィンドウ上の文字位置は、一般的に座標（**CursorRow**、**CursorColumn**）により表現され、これらの値は各ウィンドウごとに管理されています。また、X行Y桁の領域をもつウィンドウにおいては、ウィンドウの左上角は (0, 0) であり右下角は (X-1, Y-1) となり、カーソル値の有効範囲は、(0, 0) ～ (X-1, Y-1) の範囲内となります。



図G）作成直後

図G）は、2行8桁のビューポートを持つ2行15桁のウィンドウを作成した直後を示したものです。作成直後のカーソル位置は、常に (0, 0) です。カーソル位置を設定するには、**CursorRow**プロパティあるいは **CursorColumn**プロパティに適切な値を設定することにより行われます。また、カーソル位置の設定によってビューポートの移動は発生しません。つまり、カーソル位置を現在のビューポート外へ設定したとしてもビューポートの表示内容には何の影響もありません。たとえば、上図の状態においてカーソル位置を (1, 14) に設定したとすると、設定後のウィンドウとカーソルの関係は図H）のように描かれます。



図H）カーソル設定後

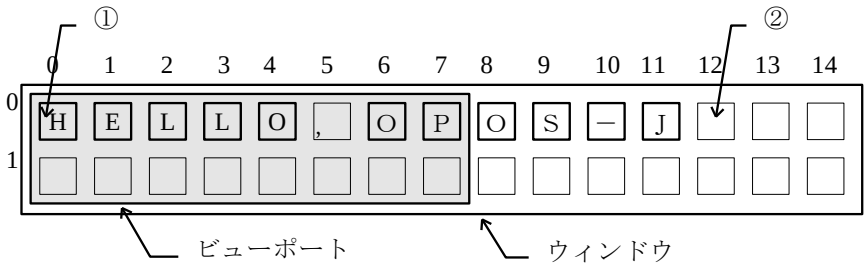
♦　文字列の表示とカーソルの移動

文字列を表示するには、**DisplayText**メソッド、あるいは、**DisplayTextAt**メソッドを用いて文字列をウィンドウ上に編集することにより行われます。このとき

に指定された文字列は、**DisplayText**メソッドの場合にはそのウィンドウで設定されているカーソル位置を基点として、また、**DisplayTextAt**メソッドの場合には指定されたカーソル位置を基点としてウィンドウ上に編集されます。ここで、そのウィンドウで管理されている**CursorUpdate**プロパティ値がTRUEの場合には、カーソル位置は編集された文字列の末尾文字の次を指すように自動で更新されます。たとえば、図G）の状態で、

DisplayText(“HELLO,OPOS-J”, DISP_DT_NORMAL)

を実行したすると図I）の結果が得られ、そのときのカーソル位置は、**CursorUpdate**プロパティ値がFALSEのときには①を指し、反対にTRUEのときには②を指すものとなります。



図I）　DisplayText メソッド実行後

◆　ビューポートのスクロール

ウィンドウ上でビューポートのスクロール表示は、次のいずれかの方法により行われます。

① **ScrollText**メソッドの呼び出しによる行、あるいは、桁単位のスクロール。

② **MarqueeType**プロパティに適切な値を設定することによるマーキースクロール。

③ **DisplayText**メソッド、あるいは、**DisplayTextAt**メソッドにより文字列がウィンドウの領域を超えることによるスクロール。

第 5 章

ハードトータル

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.2	Boolean	R/W	*適用外*
BinaryConversion	1.2	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.0	String	R	Open
Claimed	1.0	Boolean	R	Open
DataCount	1.2	Long	R	*適用外*
DataEventEnabled	1.0	Boolean	R/W	*適用外*
DeviceEnabled	1.0	Boolean	R/W	Open
FreezeEvents	1.0	Boolean	R/W	Open
OpenResult	1.5	LONG	R	無し
OutputID	1.0	Long	R	*適用外*
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.0	Long	R	無し
ResultCodeExtended	1.0	Long	R	Open
State	1.0	Long	R	無し
ControlObjectDescription	1.0	String	R	無し
ControlObjectVersion	1.0	Long	R	無し
ServiceObjectDescription	1.0	String	R	Open
ServiceObjectVersion	1.0	Long	R	Open
DeviceDescription	1.0	String	R	Open
DeviceName	1.0	String	R	Open
専用				
CapErrorDetection	1.0	Boolean	R	Open
CapSingleFile	1.0	Boolean	R	Open

CapTransactions	1.0	Boolean	R	Open
FreeData	1.0	Long	R	Open & Enable
TotalsSize	1.0	Long	R	Open & Enable
NumberOfFiles	1.0	Long	R	Open & Enable
TransactionInProgress	1.0	Boolean	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.0	無し
Close	1.0	Open
ClaimDevice	1.0	Open
ReleaseDevice	1.0	Open & Claim
CheckHealth	1.0	Open & Enable; (注1)
ClearInput	1.0	*適用外*
ClearInputProperties	1.10	*適用外*
ClearOutput	1.0	*適用外*
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.0	Open
ResetStatistics	1.8	Open & Enable; (注1)
RetrieveStatistics	1.8	Open & Enable; (注1)
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open & Enable; (注1)
専用		
ClaimFile	1.0	Open & Enable; (注2)
ReleaseFile	1.0	Open & Enable
Read	1.0	Open & Enable; (注2)
Write	1.0	Open & Enable; (注2)
SetAll	1.0	Open & Enable; (注2)
ValidateData	1.0	Open & Enable; (注2)
RecalculateValidationData	1.0	Open & Enable; (注2)
Create	1.0	Open & Enable; (注1)
Find	1.0	Open & Enable; (注1)
FindByIndex	1.0	Open & Enable; (注1)
Delete	1.0	Open & Enable; (注2)
Rename	1.0	Open & Enable; (注2)
BeginTrans	1.0	Open & Enable
CommitTrans	1.0	Open & Enable
Rollback	1.0	Open & Enable

(注1) ハードトータルデバイスを排他アクセスしているアプリケーションがいないこと。

(注2) ハードトータルデバイスを排他アクセスしているか、そのファイルに対するメソッドを実行中のアプリケーションがいないこと。

イベント

名称	*版数*	*発生条件*
DataEvent	1.0	*適用外*
DirectIOEvent	1.0	Open
ErrorEvent	1.0	*適用外*
OutputCompleteEvent	1.0	*適用外*
StatusUpdateEvent	1.3	Open,Claim,&Enable

概説

ハードトータルコントロールのOLEプログラマティクIDは“OPOS.Totals”です。

機能

ハードトータルは、次の機能を基本機能として実装しています。

トータルメモリの領域に、“”（空の文字列）の名称を持つファイルを少なくとも1つサポートします。各々のトータルファイルは連続したバイトデータのように読み書きされます。

各々のトータルファイルは固定長で生成され、削除、初期化、排他アクセス要求ができます。

トータルメモリは保護されていますが、制約つきのリソースです。たとえば、数千バイトの大きさしかないこともあります。

以下の機能は、オプションとして実装されます。

ファイル名を付けたトータルファイルを作成する機能をサポートします。これは、ルートディレクトリレベルのみのファイルシステムの特性と共通します。上記の基本性能に加えて、各トータルファイルのファイル名の変更が可能となります。

次のトランザクション処理機能をサポートします。

begin：トランザクション処理の開始

commit：トランザクション処理を終了し、変更データを保存する

rollback：トランザクション処理を終了し、変更データを廃棄する

高度なエラー検知をサポートします。このエラー検知はハードウェアまたはソフトウェアにより実現されます。

モデル

以下にハードトータルの一般的なモデルを示します。

ハードトータルデバイスは論理的に連続したバイトデータとして扱われ、アプリケーションがその内容を「トータルファイル」として細分します。これは、Createにより実行され、名称、サイズ、エラー検出レベルが割り当てられます。トータルファイルはCreate時に設定された固定長のファイルです。

基本機能のみの場合は、“”（空の文字列）の名前を持つ1つのトータルファイルを作成でき、操作することができます。オプション機能が実装されている場合は、任意の名前を持つトータルファイルを作成できます。

トータルファイルは従来のファイルシステムに似ていますが、強力なファイルシステムの提供を意図していません。どちらかといえば、データ保護やアクセスの簡便化を目的としています。

リソースの使用時の不必要なオーバヘッドを少なくするため、ディレクトリの階層構造はサポートせず、ファイルアトリビュートは最低限にし、ファイルの動的なサイズ変更はできません。

次のような操作がトータルファイルで実行できます。

- Read：一連のデータを読み取ります。
- Write：一連のデータを書き込みます。
- SetAll：トータルファイル内のすべてのデータを指定値に設定します。
- Find：既存のトータルファイルを名前で検索し、ファイルハンドルとサイズを返します。
- FindByIndex：ハードトータル内のすべてのファイルを列挙するために使用します。
- Delete：名前を指定してトータルファイルを削除します。
- Rename：トータルファイルの名前を変更します。
- ClaimFile：アプリケーションが特定のファイルの排他アクセス権を獲得します。タイムアウト値はほかのアプリケーションがアクセス中である場合の最大待ち時間を指定します。ClaimDeviceメソッドはハードトータルデバイスの排他アクセス権を獲得する場合に使用します。
- ReleaseFile：ファイルの排他アクセス権を解放します。

FreeDataプロパティは空きデータバイト数を示します。

TotalsSizeプロパティはハードトータルデバイスの総バイト数を示します。

NumberOfFilesプロパティはハードトータルデバイス内に存在するトータルファイルの数を示します。

トランザクション処理機能はオプションでサポートされます。トランザクションは一連のデータ書き込みで定義され、一つ以上のハードトータルファイルへの操作単位として適用されます。

トランザクション処理中、そのデータ書き込みはcommitまたはrollbackまでメモリに保持されます。FreeDataはトランザクション処理中、顕著に減少します。それは、操作単位としてのcommitを成立させるために、テンポラリなトータル領域が使用されるためです。

- BeginTrans：トランザクション処理の始まりを意味します。
- CommitTrans: トランザクション処理の終了と変更データの保存を行います。ソフトウェアまたはハードウェアメソッドが、完全にトランザクションが保存されるか、何もアップデートがされないことを保証するのに使用されます。このことは一般的に、トランザクションをテンポラリトータルスペースに書き込み、commitが進行中であることを示すステータス情報をデバイス内に設定し、トータルファイルにデータを書き込んだ後、テンポラリトータルスペースを解放するという一連の動きを必要とします。システムパワーダウンかリセットなどによりcommitに割込がかかった場合には、ハードトータルサービスオブジェクトが再ロードと初期化されたときに、テンポラリスペースからトータルファイルにデータをコピーすることでcommitを完了することができます。これにより関連しているトータルデータの正当性を確保できます。
- Rollback : 現在のトランザクション処理を終了し、更新内容を破棄します。Rollbackは更新内容をキャンセルするために、ユーザが介入する場合に利用します。また、高度なエラー検知機能により、トータル領域のデータが、更新用に正しく読めないことが検出されたなら、そのときはトランザクション処理をアボートする必要があります。
- TransactionInProgress：当プロパティには、トランザクション処理の現行の状態が設定されます。

アプリケーションはCommitTrans前にほかのハードトータルコントロールがファイルの排他アクセス権を獲得し、commitが失敗したりすでに排他アクセスされているステータスが返されることのないように、トランザクション処理中に使用されるファイルを排他アクセスする必要があります。

高度なエラー検知機能はオプションとして、次のようにサポートされます。

- **Read**または**Write**はバリデーションエラーを通知することがあります。データは通常バリデーションブロックに分割され、このブロック全体にサムチェックやＣＲＣが付けられています。バリデーションデータブロックのサイズはサービスオブジェクトにより決定されます。

  バリデーションエラーはアプリケーションに、読み取りまたは書き込み対象のデータを含むバリデーションブロックが、ハードウェアエラーにより無効であることを通知します。（**Write**のエラーはバリデーションブロックの一部を変更しなければならないときに起こる可能性があります。その一部分の変更前には、必ずバリデーションブロックの読み出しとブロック自体のバリデーションが行わなければなりません。）

  バリデーションエラーを通知されたときは、アプリケーションでトータルファイル内のすべてのデータを読み出すべきです。アプリケーションにて、どの部分のデータが無効であるかを決め、読み出しの結果により処置を行うべきです。

- RecalculateValidationDataはトータルファイル内のすべてのバリデーションデータの再計算を行うときに、呼び出します。RecalculateValidationDataは前項で説明したようなリカバリをした後で呼び出すものです。
- ValidateDataはトータルファイル内のすべてのデータをバリデーションチェックにより比較する場合に、呼び出します。
- データのWriteは、書き込まれたデータが存在する一個か複数のバリデーションブロックに対するバリデーションデータの再計算を自動的に行います。
- 高度なエラー検知は通常パフォーマンスの低下をもたらすため、トータルファイルを作成するごとに、アプリケーションで本機能を使用するかどうかを選択できます。

デバイスの共有

ハードトータルは共有可能なデバイスです。デバイスを共有する際のルールは以下のとおりです。

デバイスオープンの後、大部分のプロパティはリード可能となります。

デバイスをオープンしイネーブルにした後、アプリケーションはすべてのプロパティとメソッドを使用することができます。

複数のアプリケーションがデバイスをオープンし、イネーブルにした場合、そのすべてのアプリケーションはプロパティとメソッドを扱うことができます。

ひとつのアプリケーションが、ハードトータルデバイスを排他アクセスすることもできます。それはすべてのほかのアプリケーションに対し、デバイス上のどのようなファイルの読み込み、更新、または排他アクセス権の獲得要求に対しても排他を行うことになります。

ひとつのアプリケーションが、ハードトータルファイルを排他アクセスできます。それは、すべてのほかのアプリケーションに対し、そのファイルに対する読み込み、更新、または排他アクセス権の獲得要求とハードトータルデバイスに対する排他アクセス権の獲得要求に対して排他を行うことになります。

プロパティ

CapErrorDetection プロパティ

形式　　BOOL CapErrorDetection;

説明　　TRUE：高度なエラー検知機能をサポートしています。

FALSE：サポートしていません。

このプロパティはOpenメソッドにより初期化されます。

CapSingleFile プロパティ

形式　　BOOL CapSingleFile;

説明　　TRUE：空の文字列(“”)により識別されるただ1つのファイルのみサポートしています。

FALSE：ファイル名を付けたトータルファイルをサポートしています。

このプロパティはOpenメソッドにより初期化されます。

CapTransactions プロパティ

形式　　BOOL CapTransactions;

説明　　TRUE：トランザクション処理機能がサポートされています。

FALSE：サポートされていません。

このプロパティはOpenメソッドにより初期化されます。

FreeData プロパティ

形式　LONG FreeData;

説明　ハードトータルデバイス内の空き領域のバイト数です。

この値はデバイスがイネーブルになったときに適切な値に初期化され、ファイルが**Create**されるか**Delete**されるたびに更新されます。ファイルの生成時にファイル情報のためにいくらかの領域が必要ですが、**FreeData**はこの領域分の値が差し引かれています。このことは**FreeData**の値のサイズの新しいファイルが生成できることを保証します。

トランザクション処理では、commitの準備のために、ハードトータル領域を割り付ける必要があるため、トランザクション処理中のデータ書き込み時は、一時的に**FreeData**を減少させます。このため、アプリケーションは最大サイズのトランザクションが実行されるのに十分な**FreeData**が確保されていることを保証しなければなりません。

参照　Createメソッド、Writeメソッド

NumberOfFiles プロパティ

形式　LONG NumberOfFiles;

説明　現在のハードトータルデバイス内のトータルファイルの数です。

このプロパティは、デバイスをイネーブルにしたときに初期化されます。

参照　FreeDataプロパティ

TotalsSize プロパティ

形式　LONG TotalsSize;

説明　ハードトータルエリアのサイズです。このサイズはファイルが何も存在しない場合に作成できる最大のファイルサイズです。

このプロパティは、デバイスをイネーブルにしたときに初期化されます。

参照　FreeDataプロパティ

TransactionInProgress プロパティ

形式　　BOOL TransactionInProgress;

説明　　TRUE：アプリケーションはトランザクション処理中です。
FALSE：トランザクション処理中ではありません。

このプロパティはOpenメソッドによりFALSEに初期化されます。

参照　　BeginTransメソッド

メソッド

BeginTrans メソッド

形式　LONG BeginTrans ();

説明　ハードトータルへの一連の書き込みの開始をマークします。この書き込みは1つのグループとして適用されるか、またはまったく適用されないかでなければなりません。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	トランザクション機能がサポートされていないデバイスです。
その他	**ResultCode**の項目を参照してください。

参照　**CommitTrans**メソッド、**Rollback**メソッド

ClaimDevice メソッド (共通)

形式　　LONG ClaimDevice (LONG *Timeout*);

*Timeout*パラメータには、排他アクセス権を獲得するまでの最大待ち時間をミリ秒単位で設定します。
0が設定されている場合は、排他アクセス権の獲得ができなかった場合でも、その結果を直ちに戻します。
OPOS_FOREVER(-1)が設定されている場合は、メソッドは排他アクセス権が獲得できるまで必要なだけ待ちます。

説明　　デバイスに対して排他的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。

もしほかのアプリケーションが**ClaimFile**を使うことによって、ハードトータルファイルのいずれかに対して排他的アクセスが行われていれば、**ClaimDevice**メソッドは**ReleaseFile**メソッドによって、それらのファイルの排他アクセス権が解放されるまで待ち合わせます。

正常終了時は**Claimed**プロパティがTRUEになります。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	排他アクセスが認められました。**Claimed**プロパティはTRUEになります。アプリケーションがすでにデバイスを排他アクセスしていた場合にも返されます。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されています。
OPOS_E_TIMEOUT	ほかのアプリケーションがデバイスや1つ以上のファイルへの排他アクセスを行なっているため、*Timeout*ミリ秒経過するまでに、排他アクセス権を獲得できなかったことを示します。

参照　　**ReleaseDevice**メソッド、**ClaimFile**メソッド、**ReleaseFile**メソッド

ClaimFile メソッド　　*第1.8版にて更新*

形式　LONG ClaimFile (LONG *HTotalsFile*, LONG *Timeout*);

パラメータ	説明
HTotalsFile	要求するトータルファイルのハンドル
Timeout	ファイルの排他アクセス権が獲得できるまでの、最大待ち時間（ミリ秒単位） 0が設定されている場合は、排他アクセス権の獲得ができなかった場合でも、その結果を直ちに戻します。 OPOS_FOREVER(-1)が設定されている場合は、メソッドは排他アクセス権が獲得できるまで必要なだけ待ちます。

説明　指定したファイルへの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。一度許可されるとそのアプリケーションが明らかに排他アクセス権を解放するかデバイスがクローズされるまで排他アクセスは継続されます。

もしほかのアプリケーションが、**ClaimFile**メソッドを使ってこのファイルに対して排他アクセスを行っていたり、**ClaimDevice**メソッドを使用してトータルデバイスに対して排他アクセスを行っていれば、この**ClaimFile**は、それらの排他アクセス権が解放されるまで待ち合わせます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	ファイルハンドルが無効かまたは、無効な*Timeout*パラメータが指定されています。
OPOS_E_TIMEOUT	ほかのアプリケーションがトータルデバイスやトータルファイルへの排他アクセスを行なっているため、*Timeout*ミリ秒経過するまでに、排他アクセス権を獲得できなかったことを示します。

参照　**ClaimDevice**メソッド、**ReleaseFile**メソッド

CommitTrans メソッド

形式 LONG CommitTrans ();

説明 現在のトランザクション処理を終了します。直前のBeginTransメソッドと、本メソッド間のすべての書き込み要求が、ハードトータルエリアに保存されます。

戻り値 次の値のいずれかが戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	このデバイスがトランザクション処理機能をサポートしていないか、実行中のトランザクションがありません。
その他	ResultCodeの項目を参照してください。

参照 BeginTransメソッド、Rollbackメソッド

Create メソッド

形式　LONG Create (BSTR *FileName*, LONG* *pHTotalsFile*, LONG *Size*, BOOL *ErrorDetection*);

パラメータ	説明
FileName	ファイルに割り付けられる名前です。 10文字以内。すべて表示可能な文字（文字コード≧0x20）でなければなりません。
pHTotalsFile	新しく作成されるトータルファイルのハンドルへのポインタ。メソッドにより設定されます。
Size	ファイルのバイト長。一度作成されるとファイルサイズは変更できません。トータルファイルは固定長ファイルです。
ErrorDetection	このファイルに対するエラー検知レベル。 TRUEであれば、サービスオブジェクトが高度なエラー検知をサポートしていればこれを有効にします。 FALSEであれば高速なアクセスが要求されているのでこのファイルに対する高度のエラー検知は有効にしません。

説明　指定された名前、サイズ、エラー検知レベルでトータルファイルが作成されます。データエリアはバイナリの0で初期化されます。

CapSingleFileがTRUEであれば1ファイルだけが作成され、そのファイル名は空の文字列(“”)になります。それ以外であれば、作成されるハードトータルファイルの数は唯一ハードトータルエリアの有効空容量にのみ制限されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_CLAIMED	トータルデバイスがほかのアプリケーションによって排他アクセスされており、作成できません。
OPOS_E_ILLEGAL	*FileName*が長すぎるか無効な文字を含んでいます。

OPOS_E_EXISTS	*FileName*がすでに存在します。
OPOS_E_EXTENDED	**ResultCodeExtended**=OPOS_ETOT_NOROOM: ファイルを作成するのに十分なエリアがありません。
その他	**ResultCode**の項目を参照してください。

参照　　**Find**メソッド、**Delete**メソッド、**Rename**メソッド

Delete メソッド

形式　　LONG Delete (BSTR *FileName*);

*FileName*パラメータには、削除するトータルファイル名を指定します。

説明　　指定された名前のファイルを削除します。

戻り値　　次の値のいずれかが戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_CLAIMED	ほかのアプリケーションがトータルファイルまたはトータルデバイスへの排他アクセスを行っているため削除できません。
OPOS_E_ILLEGAL	*FileName*が長すぎるか無効な文字を含んでいます。
OPOS_E_NOEXIST	*FileName*が見つかりません。
その他	ResultCodeの項目を参照してください。

参照　　Createメソッド、Findメソッド、Renameメソッド

Findメソッド

形式 LONG Find (BSTR *FileName*, LONG* *pHTotalsFile*, LONG* *pSize*);

パラメータ	説明
FileName	トータルファイルの名前です。
pHTotalsFile	トータルファイルのハンドルへのポインタ。メソッドにより設定されます。
pSize	ファイルサイズへのポインタ（バイトサイズ）メソッドにより設定されます。

説明 トータルファイルを検索します。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_CLAIMED	トータルデバイスがほかのアプリケーションによって排他アクセスされているため検索できません。
OPOS_E_ILLEGAL	*FileName*が長すぎるか、無効な文字を含んでいます。
OPOS_E_NOEXIST	*FileName*が見つかりません。
その他	**ResultCode**の項目を参照してください。

参照 **Create**メソッド、**Delete**メソッド、**Rename**メソッド

FindByIndex メソッド

形式　LONG FindByIndex (LONG *Index*, BSTR* *pFileName*);

パラメータ	説明
Index	検索するトータルファイルのインデックスです。
pFileName	トータルファイル名へのポインタ。メソッドにより設定されます。

説明　与えられたインデックスで、現在示されるファイル名へのポインタが返されます。

このメソッドは、現在定義されている全トータルファイルを列挙する手間をはぶきます。インデックスが0の場合は先頭のファイル位置が返され、それに続くインデックスで以降のファイル名が返されます。もっとも大きなインデックスの有効値はNumberOfFilesの値から1引いた値です。

ファイルの作成・削除を行うと、インデックスとファイルの関係も変わります。また、このときコントロールはファイル名と属性を管理するためのデータ領域を縮めるか再調整します。このため、すべてのファイル名を正常に取り出すためには、アプリケーションはデバイスを排他アクセスする必要があります。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_CLAIMED	トータルデバイスがほかのアプリケーションによって排他アクセスされているため検索できません。
OPOS_E_ILLEGAL	*Index*が現在定義されている最大のインデックス値を超えています。
その他	ResultCodeの項目を参照してください。

参照　Createメソッド、Findメソッド

Read メソッド

形式 LONG Read (LONG *HTotalsFile*, BSTR* *pData*, LONG *Offset*, LONG *Count*);

パラメータ	説明
HTotalsFile	**Create**または**Find**メソッドから返されたトータルファイルのハンドル。
pData	トータルデータが読み出されるバッファへのポインタ。 このデータのフォーマットは**BinaryConversion**プロパティの値に依存します。詳細は**BinaryConversion**プロパティを参照してください。
Offset	読みだすデータの先頭オフセット値。
Count	読みだすデータのバイト数。

説明 トータルファイルからデータを読み出します。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_CLAIMED	トータルデバイスまたはトータルファイルがほかのアプリケーションによって排他アクセスされているため検索できません。
OPOS_E_ILLEGAL	ファイルハンドルが不正か、データレンジの一部がトータルファイルの限度を越えています。
OPOS_E_EXTENDED	**ResultCodeExtended**=OPOS_ETOT_VALIDATION: バリデーションエラーがデータの読み出し中に発生しました。
その他	**ResultCode**の項目を参照してください。

参照 **Write**メソッド

RecalculateValidationData メソッド

形式　LONG RecalculateValidationData (LONG *HTotalsFile*);

パラメータ	説明
HTotalsFile	トータルファイルのハンドル。

説明　指定されたトータルファイルのバリデーションデータを再計算します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_CLAIMED	トータルデバイスまたはトータルファイルがほかのアプリケーションによって排他アクセスされているため再計算できません。
OPOS_E_ILLEGAL	ハンドルが無効、またはサービスオブジェクトまたはこのファイルが高度なエラー検知機能をサポートしていません。
その他	**ResultCode**の項目を参照してください。

Release メソッド（共通）

形式 LONG ReleaseDevice ();

説明 デバイスに対する排他アクセス権を解放します。

アプリケーションは、**ClaimDevice**メソッドによりハードトータルデバイスを、**ClaimFile**メソッドにより特定のファイルを排他アクセス要求するが、**ReleaseDevice**メソッドの呼び出しではハードトータルデバイスに対する排他アクセス権を解放します。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	排他アクセス権を解放しました。**Claimed**プロパティはFALSEとなります。
OPOS_E_ILLEGAL	アプリケーションはデバイスに対して排他アクセスを行っていません。

参照 「**デバイス共有モデル**」、**ClaimDevice**メソッド、**ClaimFile**メソッド

ReleaseFile メソッド

形式　LONG ReleaseFile (LONG *HTotalsFile*);

パラメータ	説明
HTotalsFile	解放するトータルファイルのハンドル。

説明　指定したファイルに対する排他アクセスを解放します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	ハンドルが不正、または指定したファイルがこのアプリケーションにより排他アクセスされていません。

参照　**ClaimDevice**メソッド、**ClaimFile**メソッド

Rename メソッド

形式 LONG Rename (LONG *HTotalsFile*, BSTR *FileName*);

パラメータ	説明
HTotalsFile	ファイル名変更するトータルファイルのハンドル。
FileName	ファイルに割り付ける新しい名前、10文字以内。すべて表示可能な文字（文字コード≧0x20）でなければなりません。

説明 トータルファイルのファイル名を変更します。

CapSingleFileプロパティがTRUEであれば、このメソッドは失敗します。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_CLAIMED	トータルデバイスまたはトータルファイルがほかのアプリケーションによって排他アクセスされているため、変更できません。
OPOS_E_ILLEGAL	ファイルハンドルが不正か、または*FileName*が長すぎるか不正な文字を含んでいます。または、**CapSingleFile**プロパティがTRUEです。
OPOS_E_EXISTS	*FileName*がすでに存在します。
その他	**ResultCode**の項目を参照してください。

Rollback メソッド

形式　LONG Rollback ();

説明　現在のトランザクションを終了します。直前のBeginTransからこのメソッドまでのすべての書き込みは破棄され、ハードトータル領域には保存されません。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	トランザクション処理機能がサポートされていないか、トランザクション処理を実行中ではありません。
その他	ResultCodeの項目を参照してください。

参照　BeginTransメソッド、CommitTransメソッド

SetAll メソッド

形式 LONG SetAll (LONG *HTotalsFile*, LONG *Value*);

パラメータ	説明
HTotalsFile	トータルファイルのハンドル。
Value	設定するデータ（最下位バイト）

説明 トータルファイル内のすべてのデータを指定した値に設定します。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_CLAIMED	トータルデバイスまたはトータルファイルがほかのアプリケーションによって排他アクセスされているため、指定した値に設定できません。
その他	**ResultCode**の項目を参照してください。

ValidateData メソッド

形式　LONG ValidateData (LONG *HTotalsFile*);

パラメータ	説明
HTotalsFile	トータルファイルのハンドル。

説明　バリデーションチェックによりトータルファイルのすべてのデータを比較します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_CLAIMED	トータルデバイスまたはトータルファイルがほかのアプリケーションによって排他アクセスされているためバリデートできません。
OPOS_E_ILLEGAL	ハンドルが無効、もしくは、サービスオブジェクトかこのファイルが高度なエラー検知機能をサポートしていません。
その他	**ResultCode**の項目を参照してください。

Write メソッド

形式 LONG Write (LONG *HTotalsFile*, BSTR *Data*, LONG *Offset*, LONG *Count*);

パラメータ	説明
HTotalsFile	**Create**または**Find**メソッドから返されたトータルファイルのハンドル。
Data	書き込むデータの格納されているバッファ。このデータのフォーマットは**BinaryConversion**プロパティの値に依存します。詳細は**BinaryConversion**プロパティを参照してください。
Offset	書き込みデータの先頭オフセット。
Count	書き込みバイト数。

説明 トータルファイルにデータを書き込みます。

トランザクション処理が実行中の場合には**CommitTrans**または**Rollback**メソッドが呼び出されるまでバッファリングします。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_CLAIMED	トータルデバイスまたはトータルファイルがほかのアプリケーションによって排他アクセスされているため書き込みできません。
OPOS_E_ILLEGAL	ハンドルが不正、またはデータ列の一部がトータルファイルの範囲外です。
OPOS_E_EXTENDED	**ResultCodeExtended**=OPOS_ETOT_NOROOM: トランザクション用の動作領域が不十分のため書き込みできません。 **ResultCodeExtended**=OPOS_ETOT_VALIDATION: データの読み込みでバリデーションエラーが発生しました。
その他	**ResultCode**の項目を参照してください。

参照　　Readメソッド、BeginTransメソッド、CommitTransメソッド、RollbackメソッドFreeDataプロパティ

第 6 章
キーロック

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.2	Boolean	R/W	*適用外*
BinaryConversion	1.2	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.0	String	R	Open
Claimed	1.0	Boolean	R	Open
DataCount	1.2	Long	R	*適用外*
DataEventEnabled	1.0	Boolean	R/W	*適用外*
DeviceEnabled	1.0	Boolean	R/W	Open
FreezeEvents	1.0	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.0	Long	R	*適用外*
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.0	Long	R	*無し*
ResultCodeExtended	1.0	Long	R	Open
State	1.0	Long	R	*無し*
ControlObjectDescription	1.0	String	R	*無し*
ControlObjectVersion	1.0	Long	R	*無し*
ServiceObjectDescription	1.0	String	R	Open
ServiceObjectVersion	1.0	Long	R	Open
DeviceDescription	1.0	String	R	Open
DeviceName	1.0	String	R	Open
専用				
CapKeylockType	1.11	Long	R	Open
ElectronicKeyValue	1.11	String	R	Open & Enable

KeyPosition	1.0	Long	R	Open & Enable
PositionCount	1.0	Long	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.0	*無し*
Close	1.0	Open
ClaimDevice	1.0	Open
ReleaseDevice	1.0	Open & Claim
CheckHealth	1.0	Open & Enable
ClearInput	1.0	*適用外*
ClearInputProperties	1.10	*適用外*
ClearOutput	1.0	*適用外*
CompareFirmwareVersion	1.9	Open & Enable
DirectIO	1.0	Open
ResetStatistics	1.8	Open & Enable
RetrieveStatistics	1.8	Open & Enable
UpdateFirmware	1.9	Open & Enable
UpdateStatistics	1.8	Open & Enable
専用		
WaitForKeylockChange	1.0	Open & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.0	*適用外*
DirectIOEvent	1.0	Open
ErrorEvent	1.0	*適用外*
OutputCompleteEvent	1.0	*適用外*
StatusUpdateEvent	1.0	Open & Enable

概説

キーロックコントロールのOLEプログラマティクIDは
“OPOS.Keylock”です。

機能　*第1.11版で更新*

キーロックは最小限以下の機能を持ちます。

少なくとも3つのキーロック位置をサポートします。

キーロック位置が変化したときにハードウェア的に通知するか、ソフトウェアで検出する機能を持ちます。

第1.11版で以下の機能が追加になりました。

電子キーをサポートします。

キーロックは、3つのキーロック位置とそれぞれの定数を定義します。キーロックは、「ロック」、「通常」、「スーパバイザ」の位置を持ち、それぞれの位置と値は、次のとおりです。

LOCK_KP_LOCK1

LOCK_KP_NORM　2

LOCK_KP_SUPR 3

KeyPositionプロパティには、1から**PositionCount**プロパティで示されるキーロック位置の総数までの範囲内で、キーロック位置の値が設定されます。

電子キー値をストアする**ElectronicKeyValue**プロパティには、デバイスで定義された固有の値が、数値列としてセットされます。

デバイスの共有

キーロックは共有可能です。共有する際のルールは次のとおりです。

デバイスをオープンし、イネーブルにした後、アプリケーションはすべてのプロパティとメソッドを使用することができます。また、StatusUpdateEventイベントを受け取ることができます。

複数のアプリケーションがデバイスをオープンし、イネーブルにした場合、そのすべてのアプリケーションはプロパティとメソッドを扱うことができます。また、それらのアプリケーションのすべてにStatusUpdateEventイベントが通知されます。

キーロックは排他制御をサポートしません。ClaimDeviceメソッドがアプリケーションから呼ばれたならば、これらのメソッドはいつもOPOS_E_ILLEGALを返し、コントロールはほかの処理を何もしません。

使用上の必要条件については「一覧」を参照してください。

プロパティ

CapKeylockType プロパティ　*第1.11版で追加*

形式　LONG CapKeylockType;

説明　キーロック種類を示す値が設定されます。

KeyPositionプロパティは、次のいずれかの値が設定されます。

値	意味
LOCK_KT_STANDARD	標準的なキーロックです。 OPOSVersion1.10以前との互換性が維持されます。
LOCK_KT_ELECTRONIC	電子キーロックを示します。

このプロパティの値がLOCK_KT_ELECTRONICであると電子キーが使用されており、ElectronicKeyValueプロパティによってキーロックのポジションが提供されます。この場合、PositionCountとKeyPositionプロパティには、意味のない値が格納されます。

このプロパティはOpenメソッドによって初期化されます。

参照　ElectoronicKeyValueプロパティ、KeyPositionプロパティ、PositionCountプロパティ、StatusUpdateEventイベント

ElectroicKeyValue プロパティ　*第1.11版で追加*

形式　BSTR ElectronicKeyValue;

説明　電子キーから読み取られた値がストアされます。

このプロパティは、CapKeylockTypeプロパティがLOCK_KT_ELECTRONICのときに有効に作用します。電子キーが接続されている時、通常電子キーは、POSにローフォーマットで固有のキー番号を送信します。したがって、典型的な例として、電子キーが接続されている場合は、「0x00,0x00,0x01,0x52,0x27,0xaf」であり、接続されていない場合は「0x00,0x00,0x00,0x00,0x00,0x00」になります。このデータのフォーマットは、BinaryConversion プロパティの値に依存します。詳細は、BinaryConversion を参照してください。

このプロパティは、DeviceEnabledプロパティがTrueで初期化され値が保持されます。

参照　　**CapKeylockType**プロパティ、**StatusUpdateEvent**イベント

KeyPosition プロパティ　　*第1.11版で変更*

形式　　LONG KeyPosition;

説明　　キーロック位置を示す値が設定されます。

キーロック位置が変わると、この値はコントロールによって変更されます。アプリケーションが**StatusUpdateEvent**を受けるのに加えて、この値は、新しいキーロック位置を示すように変更されます。

KeyPositionプロパティは、次のいずれかの値が設定されます。

値	意味
LOCK_KP_LOCK	キーロックが「ロック」位置にあります。値は1です。
LOCK_KP_NORM	キーロックが「通常」位置にあります。値は2です。
LOCK_KP_SUPR	キーロックが「スーパバイザ」位置にあります。値は3です。
その他	キーロックが上記以外の位置にあります。この値は、4から**PositionCount**プロパティによって示されるキーロック位置の総数までの範囲内です。

CapKeylockTypeプロパティがLOCK_KT_ELECTRONICならば、このプロパティは常に０の値がセットされ意味を持ちません。

このプロパティはデバイスをイネーブルにしたときに初期化されます。

参照　　**ElectoronicKeyValue**プロパティ、**PositionCount**プロパティ、**StatusUpdateEvent**イベント

PositionCount プロパティ　*第1.11版で変更*

形式　LONG PositionCount;

説明　キーロックデバイスに存在する位置の総数を示します。

CapKeylockTypeプロパティがLOCK_KT_ELECTRONICならば、このプロパティは常に０の値がセットされ意味を持ちません。

このプロパティは、Openメソッドにより初期化されます。

参照　ElectoronicKeyValueプロパティ

メソッド

WaitForKeylockChange メソッド　*第1.11版で変更*

形式　LONG WaitForKeylockChange (LONG *KeyPosition*, LONG *Timeout*);

パラメータ	説明
KeyPosition	要求するキーロック位置です。以下の値を参照してください。
Timeout	この値で指定された時間内にキーロック位置が変化しなければ、制御をアプリケーションに返します。単位はミリ秒です。-1、0、正の値が指定できます。

*KeyPosition*パラメータは、次のいずれかの値を設定します。

値	意味
LOCK_KP_ANY	キーロック位置が変化するのを待ちます。値は0です。
LOCK_KP_LOCK	キーロック位置が「ロック」に設定されるのを待ちます。値は1です。
LOCK_KP_NORM	キーロック位置が「通常」に設定されるのを待ちます。値は2です。
LOCK_KP_SUPR	キーロック位置が「スーパバイザ」に設定されるのを待ちます。値は3です。
その他	キーロック位置が上記以外の位置のいずれかひとつに設定されるのを待ちます。この値は、4から**PositionCount**プロパティによって示されるキーロック位置の総数までの範囲です。

説明　指定されたキーロック位置に設定されるのを待ちます。

*KeyPosition*によって指定した位置が、現在のキーロック位置と同じならば、メソッドは直ちに戻ります。

CapKeylockTypeプロパティがLOCK_KT_ELECTRONICの場合、唯一、LOCK_KP_ANYが引数*KeyPosition*の値として指定されます。

*Timeout*が0の場合、キーロック位置が*KeyPosition*で指定された位置にあればOPOS_SUCCESSを返し、指定された位置になければOPOS_E_TIMEOUTを返します。また、*Timeout*が-1の場合は、タイムアウトは発生せず、*KeyPosition*で指定された位置に設定されるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	キーロックが指定された位置にあります。*KeyPosition*がLOCK_KP_ANYならばキーロック位置は変化しました。
OPOS_E_TIMEOUT	指定されたキーロック位置に設定される前に、タイムアウトが発生しました。
OPOS_E_ILLEGAL	無効なパラメータ値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**CapKeylockType**プロパティ、**PositionCount**プロパティ

イベント

StatusUpdateEvent イベント　　*第1.11版で変更*

形式　　void StatusUpdateEvent (LONG *State*);

*State*パラメータは更新後のキーロック位置を示します。以下の値が設定されます。

値	意味
LOCK_KP_ELECTRONIC	電子キーの値。常に「０」が返されます。
LOCK_KP_LOCK	キーロックが「ロック」位置にあります。値は1です。
LOCK_KP_NORM	キーロックが「通常」位置にあります。値は2です。
LOCK_KP_SUPR	キーロックが「スーパバイザ」位置にあります。値は3です。
その他	キーロックがほかの位置のいずれかにあります。この値は4から**PositionCount**プロパティによって示されるキーロック位置の総数までの範囲内です。

*電源通知の**StatusUpdateEvent**の値*

「第2章 共通プロパティ、メソッド、イベント」の**StatusUpdateEvent**の記述を参照。

説明　　キーロック位置が変わったとき、イベントが通知されます。

CapKeylockTypeプロパティがLOCK_KT_ELECTRONICの場合、電子キーの値は、StatusUpdateEventがアプリケーションに通知される時に、ElectronicKeyValueプロパティに格納され、イベントの引数のStatusには、LOCK_KP_ELECTRONICがセットされます。

参照　　**CapKeylockType**プロパティ、**ElectronicKeyValue**プロパティ、**PositionCount**プロパティ

第　7 章
磁気ストライプリーダ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.2	Boolean	R/W	Open
BinaryConversion	1.2	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.0	String	R	Open
Claimed	1.0	Boolean	R	Open
DataCount	1.2	Long	R	Open
DataEventEnabled	1.0	Boolean	R/W	Open
DeviceEnabled	1.0	Boolean	R/W	Open & Claim
FreezeEvents	1.0	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.0	Long	R	*適用外*
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.0	Long	R	無し
ResultCodeExtended	1.0	Long	R	Open
State	1.0	Long	R	無し
ControlObjectDescription	1.0	String	R	無し
ControlObjectVersion	1.0	Long	R	無し
ServiceObjectDescription	1.0	String	R	Open
ServiceObjectVersion	1.0	Long	R	Open
DeviceDescription	1.0	String	R	Open
DeviceName	1.0	String	R	Open

専用				
CapCardAuthentication	1.12	String	R	Open
CapDataEncryption	1.12	Long	R	Open
CapDeviceAuthentication	1.12	Long	R	Open
CapISO	1.0	Boolean	R	Open
CapJISOne	1.0	Boolean	R	Open
CapJISTwo	1.0	Boolean	R	Open
CapTrackDataMasking	1.12	String	R	Open
CapTransmitSentinels	1.5	Boolean	R	Open
CapWritableTracks	1.10	Long	R	Open
TracksToRead	1.0	Long	R/W	Open
TracksToWrite	1.10	Long	R/W	Open, Claim, & Enable
AdditionalSecurityInformation	1.12	String	R	Open
CardAuthenticationData	1.12	String	R	Open
CardAuthenticationDataLength	1.12	Long	R	Open
CardPropertyList	1.12	String	R	Open
CardType	1.12	String	R	Open
CardTypeList	1.12	String	R	Open
DataEncryptionAlgorithm	1.12	Long	R/W	Open & Claim
DecodeData	1.0	Boolean	R/W	Open
DeviceAuthenticated	1.12	Boolean	R	Open, Claim, & Enable
DeviceAuthenticationProtocol	1.12	Long	R	Open
EncodingMaxLength	1.10	Long	R	Open, Claim, & Enable
ParseDecodeData	1.0	Boolean	R/W	Open
ErrorReportingType	1.2	Long	R/W	Open
AccountNumber	1.0	String	R	Open
ExpirationDate	1.0	String	R	Open
Title	1.0	String	R	Open
FirstName	1.0	String	R	Open
MiddleInitial	1.0	String	R	Open
Surname	1.0	String	R	Open
Suffix	1.0	String	R	Open
ServiceCode	1.0	String	R	Open

Track1Data	1.0	String	R	Open
Track1DiscretionaryData	1.0	String	R	Open
Track1EncryptedData	1.12	String	R	Open
Track1EncryptedDataLength	1.12	Long	R	Open
Track2Data	1.0	String	R	Open
Track2DiscretionaryData	1.0	String	R	Open
Track2EncryptedData	1.12	String	R	Open
Track2EncryptedDataLength	1.12	Long	R	Open
Track3Data	1.0	String	R	Open
Track3EncryptedData	1.12	String	R	Open
Track3EncryptedDataLength	1.12	Long	R	Open
Track4Data	1.5	String	R	Open
Track4EncryptedData	1.12	String	R	Open
Track4EncryptedDataLength	1.12	Long	R	Open
TransmitSentinels	1.5	Boolean	R/W	Open
WriteCardType	1.12	String	R/W	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.0	無し
Close	1.0	Open
ClaimDevice	1.0	Open
ReleaseDevice	1.0	Open & Claim
CheckHealth	1.0	Open, Claim, & Enable
ClearInput	1.0	Open & Claim
ClearInputProperties	1.10	Open & Claim
ClearOutput	1.0	*適用外*
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.0	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用		
AuthenticateDevice	1.12	Open, Claim, & Enable
DeauthenticateDevice	1.12	Open, Claim, & Enable
RetrieveCardProperty	1.12	Open, Claim,
RetrieveDeviceAuthenticationData	1.12	Open, Claim, & Enable
UpdateKey	1.12	Open, Claim, & Enable
WriteTracks	1.10	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.0	Open, Claim, & Enable
DirectIOEvent	1.0	Open & Claim
ErrorEvent	1.0	Open, Claim, & Enable
OutputCompleteEvent	1.0	*適用外*
StatusUpdateEvent	1.3	Open, Claim, & Enable

概説

磁気ストライプリーダ(MSR)コントロールのOLEプログラマティックIDは、“OPOS.MSR”です。

機能　*第1.12版で更新*

磁気ストライプリーダコントロールは、次の機能を持っています。

磁気ストライプからコード化されたデータを読み取ります。データは、トラック1、2、3、および、JIS-IIのあらゆる組み合わせから読み取ることができます。

アルファニューメリックのデータバイトは、それらに対応する英数文字のコードにデコードすることができます。さらに、このデコードされたアルファニューメリックのデータは、デバイスプロパティとしてアクセスできる特定のフィールドに分けられます。

また、以下の機能がオプションとして実装されます。

次のカードタイプをサポートします。

ISO、JIS Type-I、かつ、または、JIS Type-II。

注：この仕様の目的に合わせて、次の規則を想定しています。

- トラック1はISOまたはJIS-Iトラック1。
- トラック2はISOまたはJIS-Iトラック2。
- トラック3はISOまたはJIS-Iトラック3。
- トラック4はJIS-II。
- カードタイプの判定はカードトラックが保持している内容の種類に基づいて行います。

トラックデータに制御情報を含めて通知する機能。

ＭＳＲトラックへのデータ書き込み機能。

AAMVA 規格カードの読み取り機能。
AAMVA規格は、以下のウェブサイトからダウンロードできます。
http://www.aamva.org/

顧客情報の暗号化されているトラックデータを戻す機能

データ暗号化時、アプリケーションへマスクしたトラックデータを戻す機能。

読み取ったカードが複製品ではないかを判断するためのカード認証データを戻す機能。

フィッシングや中間者攻撃を防ぐためのデバイス／ホスト間の相互認証機能。

JIS-II に対する補足

Ver1.4以前では**JIS-Ⅱ**データの扱いが明確に定義されておらず、**TrackXData**のどのプロパティにデータが格納されるかはSOに依存していました。Ver1.5以降では**JIS-II** データは、**Track4Data** プロパティに格納されますが、バージョン1.4 以前のSOとの互換性を保つために、他の**TrackXData** にも格納されることがあります。（通常は、**Track1Data**または**Track3Data**。）この場合、**DataEvent** の *Status*、**ErrorEvent** の**ErrorCodeExtended** プロパティにも二重にセットされます。ただし、アプリケーションが**Track4Data**以外のプロパティを使用すると、格納されるプロパティがSOによって異なる可能性が有るため、アプリケーションの互換性が失われる可能性があります。

モデル　*第1.12版で更新*

4つの書き込み可能なプロパティは、磁気ストライプリーダのデータ処理を制御します。

TracksToReadプロパティは、4種類のトラックからどの組み合わせでカードを読むかを制御します。すべての指定トラックをカードから読み取れなくてもエラーではありません。むしろ、このプロパティはアプリケーション処理で必要なトラックを指定します。

DecodeDataプロパティは、元のフォーマットから表示可能データへの変換を制御します。

ParseDecodeDataプロパティは、一般の磁気ストライプリーダ規定に基づき、変換済みのデータからフィールドへの展開を制御します。

ErrorReportingTypeプロパティは、無効なデータを含むトラックが読まれたときに発生するエラーイベントのハンドリング方法を制御します。

DataEncryptionAlgorithmプロパティは、そのデバイスが使うべきである暗号化（もしあれば）種類を制御します。

MSRからの入力　*第1.12版で更新*

磁気ストライプリーダコントロールは、イベント駆動入力の一般的な入力モデルに従います。

カード操作によるデータ入力を受け取ると、コントロールは**DataEvent**をキューイングします。

AutoDisableプロパティがTRUEならば、**DataEvent**のキューイング時にコントロールは自動的にディセーブルとなります。

DataEventEnabledプロパティがTRUEならば、キューイングした**DataEvent**はアプリケーションに通知されます。イベントを通知する直前にコントロールはデータをプロパティにコピーし、**DataEventEnabled**プロパティをFALSEに設定して次のデータイベントを抑止します。これにより、アプリケーションが現在の入力と関連プロパティを処理している間、以降の入力データをコントロールによってキューイングします。アプリケーションは、現在の入力データの処理を終えて次のデータ処理の準備ができたとき、**DataEventEnabled**プロパティをTRUEに設定してイベント通知を再び許可します。

入力データの読込み中または処理中に、コントロールにおいてエラーが発生した場合、**ErrorEvent**が発行され、**DataEventEnabled**プロパティがTRUEならばアプリケーションに通知されます。

DataCountプロパティを読むことでコントロールによってキューイングされたデータイベント数が得られます。

キューイングされたすべての入力データは、**ClearInput**メソッドを呼び出して削除できます。

DataEventもしくは**ErrorEvent**発生により入力された全てのデータプロパティは、**ClearInputProperties**メソッドを呼ぶ事によりデフォルト値に戻す事が出来ます。

CapDeviceAuthenticationプロパティがMSR_DA_REQUIREDに設定された場合、認証された状態（**DeviceAuthenticated**=TRUE)の時、カードリーダからの入力が可能となります。カードリーダは、**AuthenticateDevice**メソッドを呼ぶことによって認証された状態になります。

MSRへの出力　*第1.10版で追加*

トラックにデータを書き込む場合、アプリケーションは**WriteTracks**メソッドを呼びます。データを書き込む能力はデバイスの能力によります。

WriteTracksメソッドは、常に同期動作します。

MSRの暗号化と、認証　*第1.13版で追加*

暗号化 – MSR

増加するカードの不正利用行為からカード利用者を保護するために、カードイシュアは、カード利用者の情報を扱う流通小売業者、アクワイアラ、カードプロセッサ、ベンダーなどへ要求事項を設けました（PCI-DSSなど）。

流通小売業者がこれらの要求事項を満たしてカードの不正利用を防止するために、カードデータを暗号化したり、カードを認証する機能を持ったMSRがあります。MSRによってカードデータを暗号化することで、カードデータを不正利用目的で取得される可能性がなくなります。復号するための鍵がなければ、暗号化された機密のアカウントデータは利用することができません。

デバイス認証により、アプリケーションとMSRの相互が、正規の組み合わせで接続されていることを確認することができます。

- デバイスがサポートするカードデータの暗号化によって異なるいくつかの実装をサポートしています。
- 暗号化されたデータのみ返却
- 暗号化されたデータと、復号化されマスクされたデータを返却
- 暗号化されたデータと、復号化されパースしてマスクされたデータを返却

カードデータの暗号化をサポートするため、以下のプロパティとメソッドを追加しています。

Capプロパティ	プロパティ	メソッド
CapDataEncryption	AdditionalSecurityInformation	UpdateKey
CapTrackDataMasking	DataEncryptionAlgorithm	
	Track1EncryptedData	
	Track1EncryptedDataLength	
	Track2EncryptedData	
	Track2EncryptedDataLength	
	Track3EncryptedData	
	Track3EncryptedDataLength	
	Track4EncryptedData	
	Track4EncryptedDataLength	

暗号化の利用モデル

暗号化は、サービス（ソフトウェア）による暗号化とデバイス（ハードウェア）による暗号化の両方をサポートします。暗号化を行う場所は、アプリケーションからは意識されません。

- データの暗号化
- アプリケーションがデバイスをClaimし、CapDataEncryptionプロパティにMSR_DE_NONE以外がセットされ、DataEncryptionAlgorithmプロパティにデバイスがサポートしている暗号化アルゴリズムがセットされると、MSRデバイスは暗号化モードになります。トラックデータの暗号化がされない場合でも、アカウント情報のマスキングは必ずサポートしなければなりません。この要求は、ハードウェアによる暗号化、ソフトウェアによる暗号化、どちらにも適用します。
- 注意：ハードウェアによる暗号化デバイスに対して、暗号化されていないデータを要求することはありません。
-
- パースされたトラックデータ
- アプリケーションが利用するために、暗号化されていないデータを提供することができます。アプリケーションはパースされた情報をリクエストすることができます。ParseDecodeDataプロパティがTRUEにセットされると、Track1Dataプロパティと Track2Dataプロパティにセットされている復号化されたデータは、さらに分割され、それぞれのプロパティにセットされます。
-
- マスキングのサポート
- デバイスが暗号化されていないデータの返却をサポートする場合、CapTrackDataMaskingプロパティがTRUEにセットされます。TrackXDataプロパティには、暗号化されず、マスクされたトラックデータがセットされます。マスキングされるフィールドやマスキングのレベルが、実装に依存します。これにより、既存のアプリケーションが暗号化に対応したデバイスへ軽微な変更で対応することができます。
-
- UpdateKeyメソッドは、デバイスの暗号鍵を更新する手段を提供します。ホストからの一般のメッセージのフィールドに、端末に送信する新しい鍵を持つような暗号アルゴリズムで利用されます。
-

認証 – MSR

デバイスもしくはアプリケーションのなりすましを防ぐために、相互認証を行います。認証によって、アプリケーションとデバイスは、置き換わったりなりすましをしていない、正しい組み合わせで接続されていることを確かめます。

認証の機能をサポートするため、以下のプロパティとメソッドを追加しています。

Capプロパティ	プロパティ	メソッド
CapDeviceAuthentication	**DeviceAuthenticated**	**AuthenticateDevice**
	DeviceAuthenticationProtocol	**DeauthenticateDevice**
		RetrieveDeviceAuthenticationData

認証の利用モデル

RetrieveDeviceAuthenticationDataメソッドは、デバイスを証明するためのトークンをデバイスから取得するために、アプリケーションから呼び出されます。このトークンは、レスポンストークンを生成することができる信頼された第三者機関に渡すチャレンジトークンとなります。その後、アプリケーションはレスポンストークンを指定してAuthenticateDeviceメソッドを呼び出します。デバイスはレスポンスを検証し、活性化状態へ遷移するか、もしくはレスポンストークンが不正としてエラーを返します。認証を必要とするデバイス（CapDeviceAuthenticationプロパティがその能力を示します）は、認証状態にならなければ、機能しません。

MSRの場合、認証状態にならなければ、カードデータをアプリケーションに通知しません。認証状態ではないときにカードがデバイスにスワイプされても、DataEventイベントはキューイングされません。

デバイスの共有

磁気ストライプリーダは、以下に示すように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、デバイスの読み込みが始まる前か、デバイスを操作するメソッドを呼び出す前か、いくつかのプロパティにアクセスする前に、デバイスをイネーブルにしなければなりません。

使用上の必要条件については、「**一覧**」を参照のこと。

プロパティ

AccountNumber プロパティ　*第1.13版で更新*

形式　BSTR AccountNumber;

説明　最後に通されたカードデータから得られたアカウント番号です。

次の場合は、空の文字列が設定されます。

- そのフィールドが読まれたデータトラックに含まれなかった。
- トラックデータフォーマットがParseDecodeDataプロパティでリストされたものの1つでなかった。
- DataEncryptionAlgorithmがMSR_DE_NONE以外でかつCapTrackDataMaskingがFALSEであった。
- ParseDecodeDataプロパティがFALSEであった。

このプロパティは、DataEncryptionAlgorithmプロパティがMSR_DE_NONE以外でCapTrackDataMaskingプロパティがTRUEの時、マスクされたデータを含んでいます。AccountNumberプロパティがマスクされたデータを含んでいる場合、データの全体がマスクされるのか、一部がマスクされるのかは、デバイスによって決定されます。最後の４桁の数字がマスクされず、アプリケーションがレシートや取引データにこの情報を利用することで、利用されたカードの識別に有益なものとなります。また、最初の４桁の数字をマスクしないことで、ルーティングやカードの認証・登録等を行うソフトウェアにとって有益なものとなる場合があります。そのほかの数字は通常、不正利用を防ぐためにマスクされます。

参照　DataEncryptionAlgorithm プロパティ、ParseDecodeData プロパティ、
CapTrackDataMasking プロパティ

AdditionalSecurityInformationプロパティ　*第1.14.1版にて更新*

形式　BSTR AdditionalSecurityInformation;

説明　DataEventイベントによって通知されるセキュリティ／暗号情報です。
このプロパティのデータフォーマットや情報は、暗号化アルゴリズムにより変化します。
例えば暗号化アルゴリズムがDUKPTなら、このプロパティは、”DUKPT シーケンスナンバー”を含むことになります。もし選択された暗号化アルゴリズムがプロパティを要求しない場合、空の文字列にセットされます。このデータのフォーマットは、BinaryConversion プロパティの値に依存します。詳細は、BinaryConversion を参照してください。

参照　CapDataEncryption プロパティ、DataEncryptionAlgorithm プロパティ

CapCardAuthenticationプロパティ　*第1.12版で追加*

形式　BSTR CapCardAuthentication;

説明　デバイスによりサポートされているカード認証データ種類です。空の文字列が含まれていた場合、デバイスは認証データをサポートせず、**CardAuthenticationData**プロパティは空の文字列になります。
DataEventイベント通知後、**CardAuthenticationData**プロパティにて、認証データの取得ができます。
このプロパティは、**Open**メソッドにより初期化されます。

参照　CardAuthenticationData プロパティ

CapDataEncryptionプロパティ　*第1.12版で追加*

形式　LONG CapDataEncryption;

説明　**DataEncryptionAlgorithm**プロパティによって選択できるデバイスがサポートする暗号化アルゴリズムを示します。

値	意味
MSR_DE_NONE	暗号化されません。**DataEncryptionAlgorithm**プロパティにもこの値がセットされている場合、**TrackXData**プロパティと変換されたプロパティには暗号化されていないデータが含まれます。
MSR_DE_3DEA_DUKPT	Ans X9.24-2004に基づくTriple DEA暗号化技術を用いたUnique Key Per Transaction。
他	0x01000000以上の値は今後追加される暗号化アルゴリズムのために予約されています。

MSR_DE_NONEは、データ暗号化をサポートしていない意味ではなく、暗号化せずにデータを戻す機能であることを意味します。

このプロパティは、**Open**メソッドにより初期化されます。

参照　DataEncryptionAlgorithm プロパティ、TrackXEncryptedData プロパティ、UpdateKey メソッド

CapDeviceAuthenticationプロパティ　*第1.12版で追加*

形式　LONG CapDeviceAuthentication;

説明　デバイス認証のサポートレベルを示します。
デバイス認証がサポートされているかは、デバイスイネーブルにて DeviceAuthenticatedプロパティに設定されます。

もし、デバイス認証がサポートされていれば、現在デバイスが利用可能なDeviceAuthenticatedプロパティの値を保持しなければなりません。
また、認証ステータスの値が変更された場合、StatusUpdataEventイベントにて、MSR_SUE_DEVICE_AUTHENTICATEDもしくは MSR_SUE_DEVICE_DEAUTHENTICATEDが通知されます。

値	意味
MSR_DA_NOT_SUPPORTED	デバイス認証をサポートしません。
MSR_DA_OPTIONAL	要求しない限り認証をサポートしません。
MSR_DA_REQUIRED	デバイス認証を要求します

このプロパティは、Openメソッドにより初期化されます。

参照　DeviceAuthenticationProtocol プロパティ、DeviceAuthenticated プロパティ、AuthenticateDevice メソッド、DeauthenticateDevice メソッド、 RetrieveDeviceAuthenticationData メソッド

CapISO プロパティ

形式　　BOOL CapISO;

説明　　TRUE：磁気ストライプリーダはISOカードをサポートしています。
FALSE：サポートしていません。

このプロパティは、**Open**メソッドにより初期化されます。

CapJISOne プロパティ

形式　　BOOL CapJISOne;

説明　　TRUE：磁気ストライプリーダはJIS Type-Iカードをサポートしています。

FALSE：サポートしていません。

JIS-IカードはISOカードのスーパーセットです。したがって、**CapJISOne**がTRUEであれば、同じく**CapISO**がTRUEであることを意味します。

このプロパティは、**Open**メソッドにより初期化されます。

CapJISTwo プロパティ

形式　　BOOL CapJISTwo;

説明　　TRUE：磁気ストライプリーダは**JIS Type-II** カードをサポートしています。
FALSE：サポートしていません。

このプロパティは、**Open**メソッドにより初期化されます。

CapTrackDataMaskingプロパティ　　*第1.13版で追加*

形式　　BOOL CapTrackDataMasking;

説明　TRUE：暗号化されたトラックデータをマスクしています。
FALSE：暗号化されたトラックデータをマスクしていません。

この値がTRUEで（**DataEncryptionAlgorithm**プロパティによって）暗号化ができる場合はトラックデータをマスクすることができるため、**TrackXData**プロパティと変換されたトラックデータプロパティにはマスクされたデータが含まれます。データマスキングの範囲と結果はサービスに依存します。サービスは、**DirectIO**メソッドでマスキングの設定を選択できますが、最小限のマスキングが指定されてもトラックデータやアカウント番号の復元ができないものであることが推奨されます。デバイスによっては、**FirstName**プロパティ、**MiddleInitial**プロパティ、**Title**プロパティ、**Surname**プロパティ、**ExpirationDate**プロパティなどの情報を、アプリケーションの処理のためにマスクせずに提供します。さらに、デバイスによってはアカウント番号の一部だけをマスクします。**CapTrackDataMasking**プロパティはデバイスがデータ暗号化をサポートしている場合のみ、すなわち、**CapDataEncryption**プロパティがMSR_DE_NONEでない場合のみTRUEになります。
このプロパティは、**Open**メソッドにより初期化されます。

参照　**CapDataEncryption** プロパティ、**DataEncryptionAlgorithm** プロパティ、**TrackXData** プロパティ、**ParseDecodeData** プロパティ、**DirectIO** メソッド

CapTransmitSentinels プロパティ　*第1.5版で追加*

形式　BOOL CapTransmitSentinels;

説明　TRUE：磁気ストライプリーダは開始符号、終了符号を読み取りデータに付加して返す機能をサポートしています。
FALSE：サポートしていません。

このプロパティは**Open**メソッドにより初期化されます。

参照　**TransmitSentinels** プロパティ

CapWritableTracks プロパティ　*第1.10版で追加*

形式　LONG CapWritableTracks;

説明　ＭＳＲの書き込み可能トラックを示します。

例として、デバイスがTrack1、2と3のトラックへの書込みをサポートしている場合MSR_TR_1_2_3にセットされます。MSRトラックへの書込みがサポートされていない場合MSR_TR_NONEにセットされます。

値	意味
MSR_TR_NONE	Trackの書き込みはサポートされません。
MSR_TR_1	Track 1の書き込み。
MSR_TR_2	Track 2の書き込み。
MSR_TR_3	Track 3の書き込み。
MSR_TR_1_2	Track 1と2の書き込み。
MSR_TR_1_3	Track 1と3の書き込み。
MSR_TR_2_3	Track 2と3の書き込み。
MSR_TR_1_2_3	Track 1、2、と3の書き込み。
MSR_TR_4	Track 4の書き込み。
MSR_TR_1_4	Track 1と4の書き込み。
MSR_TR_2_4	Track 2、と4の書き込み。
MSR_TR_3_4	Track 3、と4の書き込み。
MSR_TR_1_2_4	Track 1、2、と4の書き込み。
MSR_TR_1_3_4	Track 1、3、と4の書き込み。
MSR_TR_2_3_4	Track 2、3、と4の書き込み。
MSR_TR_1_2_3_4	Track 1、2、3と4の書き込み。

このプロパティはOpenメソッドにより初期化されます。

参照　TracksToWrite プロパティ

CardAuthenticationDataプロパティ　*第1.14.1版にて更新*

形式　BSTR CardAuthenticationData;

説明　カード認証情報で、DataEventイベント通知により格納されます。認証情報の内容および内部フォーマットは、CapCardAuthenticationプロパティの値により変化します。CapCardAuthenticationが空文字であった場合、このプロパティは空の文字列にセットされます。このプロパティの値はDataEncryptionAlgorithmプロパティに含まれる暗号化アルゴリズムによって暗号化されます。このプロパティの暗号化されていない値の長さはCardAuthenticationDataLengthプロパティに含まれます。このデータのフォーマットは、BinaryConversion プロパティの値に依存します。詳細は、BinaryConversion を参照してください。

参照　CapCardAuthentication プロパティ、CardAuthenticationDataLength プロパティ、DataEncryptionAlgorithm プロパティ

CardAuthenticationDataLengthプロパティ　*第1.13版で追加*

形式　LONG CardAuthenticationDataLength;

説明　暗号化される前のCardAuthenticationDataプロパティの長さが入ります。CapCardAuthenticationプロパティが空文字列の場合、0がセットされます。多くの暗号化アルゴリズムは暗号化される前の入力データのパディングを必要とします。このプロパティは暗号化される前のオリジナル（非パディング）なデータの長さを含みます。復号後にオリジナルデータに戻す際に必要となります。

参照　CapCardAuthentication プロパティ、CardAuthenticationData プロパティ

CardPropertyListプロパティ　*第1.12版で追加*

形式　BSTR CardPropertyList;

説明　最後に通されたカードから順にカードプロパティ名がカンマで区切られ格納されます。

次の場合、空の文字列に設定されます。

- カードの種類が認識できなかった。
- ParseDecodeDataがFALSEであった。

このプロパティは**Open**メソッドにより初期化されます

参照　　**ParseDecodeData** プロパティ

CardTypeプロパティ　　*第1.12版で追加*

形式　　**BSTR CardType;**

説明　　最後に通されたカードを識別するカードの種類が格納されます。カードのフォーマットが認識されない場合、プロパティには空文字列がセットされます。このプロパティの値が、アンダースコア"_"で始まっている場合はベンダー仕様、アンダースコアで始まっていない場合は標準カード仕様を意味します。
現在、標準的なカードとして定義されているものを以下に示します。

"BANK" 銀行のクレジット／デビットカード
"AAMVA" AAMVA規格カード

このプロパティは**Open**メソッドにより空の文字列に初期化されます。

CardTypeListプロパティ　　*第1.12版で追加*

形式　　**BSTR CardTypeList;**

説明　　サポートされるカードの種類がコンマで区切られて格納されます。ベンダーが非標準カードを用いる場合、アンダースコアで始まるカードタイプのみサポートされます。アンダースコアで始まらないカードタイプは全て標準的なカードタイプです。
現在、標準的なカードとして定義されているものを以下に示します。

"BANK" 銀行のクレジット／デビットカード

"AAMVA" AAMVA規格カード

銀行のカードでは次のプロパティに変換され、**RetrieveCardProperty**メソッドを通してアクセスされます。

"AccountNumber"

"ExpirationDate"

"FirstName"

"MiddleInitial"

"ServiceCode"

“Suffix”

“Surname”

“Title”

AAMVA規格では、次のプロパティに変換され、
RetrieveCardPropertyメソッドを通してアクセスされます。

“Address”

“BirthDate”

“City”

“Class”

“Endorsements”

“ExpirationDate”

“EyeColor”

“FirstName”

“Gender”

“HairColor”

“Height”

“LicenseNumber”

“PostalCode”

“Restrictions”

“State”

“Suffix”

“Surname”

“Weight”

このプロパティは**Open**メソッドにより初期化されます。

参照　　**ParseDecodeData** プロパティ

DataEncryptionAlgorithmプロパティ　R/W　　*第1.12版で追加*

形式　　LONG DataEncryptionAlgorithm;

説明　　トラックデータを暗号化するのに使われる暗号化アルゴリズムを示します。このプロパティには**CapDataEncryption**プロパティで定義されている暗号化アルゴリズムの中からセットされます。しかし、セキュリティの理由で、デバイスサービスはアプリケーションが選択した値にセットされない事があります。
注意：このプロパティは、デバイスがOpened＆Claimed（非Enabled）時に、更新可能です。
このプロパティは、**Open**メソッドにより初期化されます。暗号化をサポートするデバイスのために、プロパティは**CapDataEncryption**によって与えられるどの値にも初期化することができます。

戻り値　　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_E_ILLEGAL	指定した暗号化アルゴリズムをサポートしません。もしくはデバイスが現在Enable状態です。

参照　　**CapDataEncryption** プロパティ、**TrackXEncryptedData** プロパティ、**UpdateKey** メソッド

DecodeData プロパティ　R/W　　*第1.12版で更新*

形式　　BOOL DecodeData;

説明　　FALSEならば、**Track1Data**、**Track2Data**、**Track3Data**、および**Track4Data**プロパティは、元のエンコードされたビット列「コード変換しないフォーマット(Raw Format)」を含みます。

TRUEならば、**Track1Data**、**Track2Data**、**Track3Data**、および**Track4Data**プロパティ内のトラックデータの各バイトは、コード変換していないフォーマットから、ASCIIビット列に対応して変換されます。7ビットデータは、それに対応するアルファニューメリック、または、カタカナキャラクタを判読するのにデコードを必要としないので、この変換は主として7ビットフォーマット以外のデータに適用されます。

各カードタイプ、トラックそしてトラックデータフォーマットに対して行うデコーディングは次のとおりです。

カードタイプ	TrackDataプロパティ	データフォーマット	変換されていないバイト	デコードされたバイト
ISO	**Track1Data**	6-ビット	0x00 － 0x3F	0x20 － 0x5F
	Track2Data	4-ビット	0x00 － 0x0F	0x30 － 0x3F
	Track3Data	4-ビット	0x00 － 0x0F	0x30 － 0x3F
JIS-I	**Track1Data**	6-ビット	0x00 － 0x3F	0x20 － 0x5F
	Track1Data	7-ビット	0x00 － 0x7F	データ変化なし
	Track2Data	4-ビット	0x00 － 0x0F	0x30 － 0x3F
	Track3Data	4-ビット	0x00 － 0x0F	0x30 － 0x3F
	Track3Data	7-ビット	0x00 － 0x7F	データ変化なし
JIS-II	**Track4Data**	7-ビット	0x00 － 0x7F	データ変化なし
AAMVA	**Track1Data**	6-ビット	0x00 － 0x3F	0x20 － 0x5F
	Track2Data	4-ビット	0x00 － 0x0F	0x30 － 0x3F
	Track3Data	6 -ビット	0x00 － 0x3F	0x30 － 0x5F

このプロパティは**Open**メソッドによりTRUEに初期化されます。

このプロパティをFALSEに設定することにより、自動的に**ParseDecodeData**プロパティがFALSEに設定されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

参照　**ParseDecodeData** プロパティ

DeviceAuthenticatedプロパティ　*第1.12版で追加*

形式　BOOL DeviceAuthenticated;

説明　デバイス認証サポート（CapDeviceAuthenticationプロパティがMSR_DA_NOT_SUPPORTED以外）の時、デバイスイネーブルにて本プロパティは更新されます。
デバイス認証状態が変化すると、DeviceAuthenticated プロパティが更新され、StatusUpdateEventイベントにて
MSR_SUE_DEVICE_AUTHENTICATED または
MSR_SUE_DEVICE_DEAUTHENTICATEDが通知されます。

この大きな理由は、認証タイムアウトイベントやアプリケーションによって起動されていない他のアクションをアプリケーションに通知するためです。

参照　CapDeviceAuthentication プロパティ、AuthenticateDevice メソッド DeauthenticateDevice メソッド、
RetrieveDeviceAuthenticationData メソッド

DeviceAuthenticationProtocolプロパティ　*第1.12版で追加*

形式　LONG DeviceAuthenticationProtocol;

説明　デバイスがサポートする認証プロトコルが格納されます。

値	意味
MSR_AP_NONE	デバイス認証をサポートしません。
MSR_AP_CHALLNGERESPONSE	チャレンジレスポンス（誰何－応答）プロトコル。

このプロパティは、Openメソッドにより初期化されます。

参照　CapDeviceAuthentication プロパティ、DeviceAuthenticated プロパティ、AuthenticateDevice メソッド、DeauthenticateDevice メソッド、RetrieveDeviceAuthenticationData メソッド

EncodingMaxLength プロパティ　　*第1.10版で追加*

形式　　LONG EncodingMaxLength;

説明　　TracksToWriteプロパティで定義されているトラックの書き込み可能最大データ長を示します。
複数のトラックがTracksToWriteプロパティで選択されている場合、トラックの中で最小のデータ長がこのプロパティに反映されます。

このプロパティはOpenメソッドにより初期化されます。

参照　　TracksToWrite プロパティ、WriteTracks メソッド

ErrorReportingType プロパティ　R/W　　*第1.12版にて更新*

形式　　LONG ErrorReportingType;

説明　　カードを通した場合や、**TracksToRead**プロパティによって指定された1つまたはそれ以上のトラックにデータとエラーが含まれる場合、**ErrorEvent**によってエラーが報告されます 。

アプリケーションに**ErrorEvent**が通知されたとき、2種類のエラー報告タイプがサポートされます。

次の場合は、空の文字列が設定されます。

- カードレベル：一般的なエラーステータスが与えられ、カードデータは得られません。カードレベルは、単純にカードデータが正しく読めたかどうかを調査するために使用します。
- トラックレベル：*ErrorLocus*がOPOS_EL_INPUTで*ResultCode*がOPOS_E_EXTENDEDのとき、*ResultCodeExtended*にそれぞれのトラックの状態が格納され、トラックのプロパティが**DataEvent**と共に更新されます。無効なデータを持つトラックはトラックのプロパティが空にセットされます。このレベルはアプリケーションが他のトラックにエラーがある時、読み取りに成功したトラックのデータを活用出来るときに使用されるべきです。例として、**TracksToRead**がMSR_TR_1_2_3で、読み取られたカードのデータがトラック1と2が正常で、トラック3に「ランダムノイズ」がありMSRよりエラーとして判定されているとします。トラックレベルエラー報告では**ErrorEvent**はトラック１と２のプロパティに正常データとしてセットし、トラック３のプロパティに空をセットし、エラーコードにそれぞれのトラックの状態をセットします。

  トラックレベルエラー報告を処理するための流れは、以下の様になります。

  - カード読み取りが発生しトラックエラーが検知された場合：
    - **DataEvent** が通知される場合、最古の **DataEvent** の前に **ErrorEvent** を *ErrorLocus*=OPOS_EL_INPUT_DATA として通知されます。
    - イベントキューの最後に、常に **ErrorEvent** を *ErrorLocus*=OPOS_EL_INPUT として通知されます。カードが取得したデータをこのイベントと関連付けます。
  - **ErrorEvent**が*ErrorLocus*=OPOS_EL_INPUT_DATAで通知された場合：
    他のプロパティは変更されません。

- **ErrorEvent**が*ErrorLocus*=OPOS_EL_INPUTで通知された場合：**TrackXData**プロパティまたは**TrackXEncryptedData**プロパティはカード読み取りデータにセットされます。読み取りエラーが発生したトラックはプロパティが空になります。
-

値	意味
MSR_ERT_CARD	カードレベルでエラーを報告します。
MSR_ERT_TRACK	トラックレベルでエラーを報告します。

このプロパティは**Open**メソッドによりMSR_ERT_CARDに初期化され、この場合、第1.2版以前と同等の機能がサポートされます。

戻り値 本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

参照 **TracksToRead** プロパティ、**TrackXData** プロパティ 、**TrackXEncryptedData** メソッド、**DataEvent** イベント、**ErrorEvent** イベント

ExpirationDate プロパティ *第1.12版にて更新*

形式 **BSTR ExpirationDate;**

説明 最後に通されたカードデータから得られた満期日がASCII４桁の10進数 “YYMM”の形式で入ります。例えば、1998年2月は、”9802”であり、2018年8月は、”1808”となります。

次の場合は、空の文字列が設定されます。

- そのフィールドが読まれたデータトラックに含まれなかった。
- トラックデータフォーマットが**ParseDecodeData**プロパティの項目で記載されているものでなかった。
- **DataEncryptionAlgorithm**プロパティがMSR_DE_NONE以外でかつ**CapTrackDataMasking**プロパティがFALSEであった。
- **ParseDecodeData**プロパティがFALSEであった。

このプロパティは、**DataEncryptionAlgorithm**プロパティがMSR_DE_NONE以外で**CapTrackDataMasking**プロパティがTRUEの時、マスクされたデータを含んでいるかもしれません。

参照　**ParseDecodeData** プロパティ、**CapTrackDataMasking** プロパティ

FirstName プロパティ　*第1.12版にて更新*

形式　**BSTR FirstName;**

説明　最後に通されたカードデータから得られたファーストネームです。

次の場合は、空の文字列が設定されます。

- そのフィールドが読まれたデータトラックに含まれなかった。
- トラックデータフォーマットが**ParseDecodeData**プロパティの項目で記載されているものでなかった。
- **DataEncryptionAlgorithm**プロパティがMSR_DE_NONE以外でかつ**CapTrackDataMasking**プロパティがFALSEであった。
- **ParseDecodeData**プロパティがFALSEであった。

このプロパティは、**DataEncryptionAlgorithm**プロパティがMSR_DE_NONE以外で**CapTrackDataMasking**プロパティがTRUEの時、マスクされたデータを含んでいるかもしれません。

参照　**ParseDecodeData** プロパティ、**CapTrackDataMasking** プロパティ

MiddleInitial プロパティ　　*第1.12版にて更新*

形式　　BSTR MiddleInitial;

説明　　最後に通されたカードデータから得られたミドルイニシャルです。

次の場合は、空の文字列が設定されます。

- そのフィールドが読まれたデータトラックに含まれなかった。
- トラックデータフォーマットが**ParseDecodeData**プロパティの項目で記載されたものでなかった。
- **DataEncryptionAlgorithm**プロパティがMSR_DE_NONE以外でかつ**CapTrackDataMasking**プロパティがFALSEであった。
- **ParseDecodeData**プロパティがFALSEであった。

このプロパティは、**DataEncryptionAlgorithm**プロパティがMSR_DE_NONE以外で**CapTrackDataMasking**プロパティがTRUEの時、マスクされたデータを含んでいるかもしれません。

参照　　**ParseDecodeData** プロパティ、**CapTrackDataMasking** プロパティ

ParseDecodeData プロパティ　R/W　　*第1.12版にて更新*

形式　　BOOL ParseDecodeData;
BOOL ParseDecodedData;　　(ParseDecodeData.6と同義)

説明　　TRUEであれば、**Track1Data**および**Track2Data**プロパティに含まれるデコードされたデータは、さらにフィールドに分割されて対応するプロパティに設定されます。**Track3Data**は、そのデータ内容がカード発行人によって定義された独自のフォーマットであるので変換されません。JIS-I Track 1フォーマットCおよび、ISO Track 1フォーマットCデータは、同様の理由で変換されません。また、**Track4Data**も変換されません。

フィールドに対応するプロパティは、次のフォーマットからなるデータを持つカードのために定義されたフィールドです。

- JIS-I / ISO Track 1フォーマットA
- JIS-I / ISO Track 1フォーマットB
- JIS-I / ISO Track 1 VISAフォーマット(デファクトスタンダード)
- JIS-I / ISO Track 2フォーマット

このプロパティは、**Open**メソッドによりTRUEに初期化されます。

このプロパティをTRUEへ設定することによって、自動的に**DecodeData** プロパティがTRUEに設定されます。

注意：**DataEncryptionAlgorithm**プロパティがMSR_DE_NONEであり**CapTrackDataMasking**プロパティがTRUEの場合、**TrackXData**プロパティと、マスクされたトラックデータは変換されて各プロパティにセットされます。プロパティを変換した結果は、**TrackXData**プロパティと同じマスクされた値が含まれます。

戻り値　　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
E_ILLEGAL	**DataEncryptionAlgorithm**プロパティがMSR_DE_NONE、もしくは**CapTrackDataMasking**プロパティがTRUEの場合にのみTRUEがセットされます。

参照 DecodeData プロパティ、Surname プロパティ、Suffix プロパティ、AccountNumber プロパティ、FirstName プロパティ、MiddleInitial プロパティ、Title プロパティ、ExpirationDate プロパティ、ServiceCode プロパティ、Track1DiscretionaryData プロパティ、Track2DiscretionaryData プロパティ

ServiceCode プロパティ *第1.12版にて更新*

形式 BSTR ServiceCode;

説明 最後に通されたカードデータから得られたサービスコードです。

次の場合は、空の文字列が設定されます。

- そのフィールドが読まれたデータトラックに含まれなかった。
- トラックデータフォーマットがParseDecodeDataプロパティの項目で記載されているものでなかった。
- DataEncryptionAlgorithmプロパティがMSR_DE_NONE以外でかつCapTrackDataMaskingプロパティがFALSEであった。
- ParseDecodeDataプロパティがFALSEであった。

このプロパティは、DataEncryptionAlgorithmプロパティがMSR_DE_NONE以外で、CapTrackDataMaskingプロパティがTRUEの時、マスクされたデータを含んでいるかもしれません。

参照 ParseDecodeData プロパティ、CapTrackDataMasking プロパティ

---

[6] MSRコントロールオブジェクトは**ParseDecodeData**プロパティをサポートしなければなりません。加えてAPG第1.1版以前での誤記に従い、同等の機能をもつ**ParseDecodedData**プロパティもサポートすることをお薦めします。

Suffix プロパティ　*第1.12版にて更新*

形式　BSTR Suffix;

説明　最後に通されたカードデータから得られたサフィックスです。

次の場合は、空の文字列が設定されます。

- そのフィールドが読まれたデータトラックに含まれなかった。
- トラックデータフォーマットが**ParseDecodeData**プロパティの項目で記載されているものでなかった。
- **DataEncryptionAlgorithm**プロパティがMSR_DE_NONE以外でかつ**CapTrackDataMasking**プロパティがFALSEであった。
- **ParseDecodeData**プロパティがFALSEであった。

このプロパティは、**DataEncryptionAlgorithm**プロパティがMSR_DE_NONE以外で**CapTrackDataMasking**プロパティがTRUEの時、マスクされたデータを含んでいるかもしれません。

参照　**ParseDecodeData** プロパティ、**CapTrackDataMasking** プロパティ

Surname プロパティ　*第1.12版にて更新*

形式　BSTR Surname;

説明　最後に通されたカードデータから得られた姓です。

次の場合は、空の文字列が設定されます。

- そのフィールドが読まれたデータトラックに含まれなかった。
- トラックデータフォーマットが**ParseDecodeData**プロパティの項目で記載されているものでなかった。
- **DataEncryptionAlgorithm**プロパティがMSR_DE_NONE以外でかつ**CapTrackDataMasking**プロパティがFALSEであった。
- **ParseDecodeData**プロパティがFALSEであった。

このプロパティは、**DataEncryptionAlgorithm**プロパティがMSR_DE_NONE以外で**CapTrackDataMasking**プロパティがTRUEの時、マスクされたデータを含んでいるかもしれません。

参照　**ParseDecodeData** プロパティ、**CapTrackDataMasking** プロパティ

Title プロパティ　　*第1.12版にて更新*

形式　　BSTR Title;

説明　　最後に通されたカードデータから得られた肩書きです。

次の場合は、空の文字列が設定されます。

- そのフィールドが読まれたデータトラックに含まれなかった。
- トラックデータフォーマットがParseDecodeDataプロパティの項目で記載されているものでなかった。
- DataEncryptionAlgorithmプロパティがMSR_DE_NONE以外でかつCapTrackDataMaskingプロパティがFALSEであった。
- ParseDecodeDataプロパティがFALSEであった。

このプロパティは、DataEncryptionAlgorithmプロパティがMSR_DE_NONE以外でCapTrackDataMaskingプロパティがTRUEの時、マスクされたデータを含んでいるかもしれません。

参照　　ParseDecodeData プロパティ、CapTrackDataMasking プロパティ

Track1Data プロパティ　*第1.14.1版にて更新*

形式　BSTR Track1Data;

説明　最後に通されたカードのトラック1の内容、あるいは空の文字列です。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

TransmitSentinels が FALSE なら、このプロパティには開始符号と終了符号を含まないトラックデータが入ります。

TransmitSentinels が TRUE なら、開始符号と終了符号を含みます。

DecodeDataプロパティがTRUEならば、“生(Raw)”フォーマットから変換されてこのプロパティにデータが入ります。**ParseDecodeData** プロパティがTRUEに設定されていれば、このプロパティによって返されたデータは解析されて、ほかのプロパティにも設定されます。

DataEncryptionAlgorithmプロパティがMSR_DE_NONEでない場合、以下のルールが追加で適用されます。

- **CapTrackDataMasking**プロパティがTRUEの場合、トラックデータに含まれる有意義と思われる全ての情報をマスクもしくは隠蔽しようと試みます。例えば、アカウントナンバーだけでなく、任意データまでをマスクもしくは隠蔽しようとします。可能であればトラックデータのオリジナルフォーマットを維持しようとする間に文字をマスクした文字に置換されます。
- **CapTrackDataMasking**プロパティがFALSEの場合、このプロパティには空の文字列が設定されます。

空の文字列は、トラックがアクセス可能でなかったことを示します。

参照　**TracksToRead プロパティ**、**TransmitSentinels** プロパティ 、**ParseDecodeData** プロパティ、**CapTrackDataMasking** プロパティ

Track1DiscretionaryData プロパティ　*第1.14.1版にて更新*

形式　BSTR Track1DiscretionaryData;

説明　最後に通されたカードデータから得られたトラック1の任意データです。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

次の場合は、空の文字列が設定されます。

- そのフィールドが読まれたデータトラックに含まれなかった。
- トラックデータフォーマットが**ParseDecodeData**プロパティの項目で記載されているものでなかった。
- **DataEncryptionAlgorithm**プロパティがMSR_DE_NONE以外でかつ**CapTrackDataMasking**プロパティがFALSEであった。
- **ParseDecodeData**プロパティがFALSEであった。

このプロパティは、**DataEncryptionAlgorithm**プロパティがMSR_DE_NONE以外で**CapTrackDataMasking**プロパティがTRUEの時、マスクされたデータを含んでいるかもしれません。

このプロパティに含まれるデータ量は、トラック1データのフォーマットに応じて変化します。

参照　**ParseDecodeData** プロパティ、**CapTrackDataMasking** プロパティ

Track1EncryptedDataプロパティ　*第1.14.1版にて更新*

形式　BSTR Track1EncryptedData;

説明　最後に通されたカードの暗号化されたトラック1のデータが格納されます。このプロパティは、**DataEncryptionAlgorithm**プロパティがMSR_DE_NONEである場合、空の文字列が格納されます。暗号化後のこのプロパティの長さは、**Track1EncyptedDataLength**プロパティに格納されます。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

参照　**DataEncryptionAlgorithm** プロパティ、**Track1EncryptedDataLength** プロパティ

Track1EncryptedDataLengthプロパティ　*第1.12版にて追加*

形式　LONG Track1EncryptedDataLength;

説明　暗号化される前の**Track1Data**プロパティの長さが格納されます。多くの暗号化アルゴリズムは暗号化される前の入力データのパディングを必要とします。このプロパティは暗号化される前のオリジナルの非パディングなデータの長さを含み、復号後、オリジナルデータに戻す際に必要となります。

参照　**DataEncryptionAlgorithm** プロパティ、**Track1EncryptedData** プロパティ

Track2Data プロパティ　*第1.14.1版にて更新*

形式　BSTR Track2Data;

説明　最後に通されたカードデータのトラック2の内容、あるいは空の文字列です。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

TransmitSentinels が FALSE なら、このプロパティには開始符号と終了符号を含まないトラックデータが入ります。

TransmitSentinels が TRUE なら、開始符号と終了符号を含みます。

DecodeDataプロパティがTRUEならば、“生(Raw)”フォーマットから変換されてこのプロパティにデータが入ります。**ParseDecodeData**プロパティがTRUEに設定されていれば、このプロパティによって返されたデータは解析されて、ほかのプロパティにも設定されます。

DataEncryptionAlgorithmプロパティがMSR_DE_NONEでない場合、以下のルールが追加で適用されます。

- **CapTrackDataMasking**プロパティがTRUEの場合、トラックデータに含まれる有意義と思われる全ての情報をマスクもしくは隠蔽しようと試みます。例えば、アカウントナンバーだけでなく、任意データまでをマスクもしくは隠蔽しようとします。可能であればトラックデータのオリジナルフォーマットを維持しようとする間に文字をマスクした文字に置換するため、正常に解析されます。
- **CapTrackDataMasking**プロパティがFALSEの場合、このプロパティには空の文字列が設定されます。

空の文字列は、トラックがアクセス可能でなかったことを示します。

参照　TracksToRead プロパティ、TransmitSentinels プロパティ、ParseDecodeData プロパティ、CapTrackDataMasking プロパティ

Track2DiscretionaryData プロパティ　*第1.14.1版にて更新*

形式　BSTR Track2DiscretionaryData;

説明　最後に通されたカードデータから得られたトラック2の任意データです。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

次の場合は、空の文字列が設定されます。

- そのフィールドが読まれたデータトラックに含まれなかった。
- トラックデータフォーマットが**ParseDecodeData**プロパティの項目に記載されているものでなかった。
- **DataEncryptionAlgorithm**プロパティがMSR_DE_NONE以外でかつ**CapTrackDataMasking**プロパティがFALSEであった。
- **ParseDecodeData**プロパティがFALSEであった。

このプロパティは、**DataEncryptionAlgorithm**プロパティがMSR_DE_NONE以外で**CapTrackDataMasking**プロパティがTRUEの時、マスクされたデータを含んでいるかもしれません。

参照　**ParseDecodeData** プロパティ、**CapTrackDataMasking** プロパティ

Track2EncryptedDataプロパティ　*第1.14.1版にて更新*

形式　BSTR Track2EncryptedData;

説明　最後に通されたカードの暗号化されたトラック2データが格納されます。このプロパティは**DataEncryptionAlgorithm**プロパティがMSR_DE_NONEである場合、空の文字列が格納されます。暗号化後のこのプロパティの長さは**Track2EncyptedDataLength**プロパティに格納されます。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

参照　**DataEncryptionAlgorithm** プロパティ、**Track2EncryptedDataLength** プロパティ

Track2EncryptedDataLengthプロパティ　　*第1.12版にて追加*

形式　　LONG Track2EncryptedDataLength;

説明　　暗号化される前のTrack2Dataプロパティの長さが格納されます。多くの暗号化アルゴリズムは暗号化される前の入力データのパディングを必要とします。このプロパティは暗号化される前のオリジナルの非パディングなデータの長さを含み、復号後オリジナルデータに戻す際に必要となります。

参照　　DataEncryptionAlgorithm プロパティ、Track2EncryptedData プロパティ

Track3Data プロパティ　*第1.14.1版にて更新*

形式　BSTR Track3Data;

説明　最後に通されたカードのトラック3の内容、あるいは空の文字列です。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

TransmitSentinels が FALSE なら、このプロパティには開始符号と終了符号を含まないトラックデータが入ります。

TransmitSentinels が TRUE なら、開始符号と終了符号を含みます。

DecodeDataプロパティがTRUEならば、“生(Raw)”フォーマットから変換されてこのプロパティにデータが入ります。

DataEncryptionAlgorithmプロパティがMSR_DE_NONEでない場合、以下のルールが追加で適用されます。

- **CapTrackDataMasking**プロパティがTRUEの場合、トラックデータに含まれる有意義と思われる全ての情報をマスクもしくは隠蔽しようと試みます。例えば、アカウントナンバーだけでなく、任意データまでをマスクもしくは隠蔽しようとします。可能であればトラックデータのオリジナルフォーマットを維持しようとする間に文字をマスクした文字に置換するため、正常に変換されます。
- **CapTrackDataMasking**プロパティがFALSEの場合、このプロパティには空の文字列が設定されます。

空の文字列は、トラックがアクセス可能でなかったことを示します。

参照　**TracksToRead** プロパティ、**TransmitSentinels** プロパティ，**ParseDecodeData** プロパティ、**CapTrackDataMasking** プロパティ

Track3EncryptedDataプロパティ　*第1.14.1版にて更新*

形式　BSTR Track3EncryptedData;

説明　最後に通されたカードの暗号化されたトラック3データが格納されます。このプロパティは**DataEncryptionAlgorithm**プロパティがMSR_DE_NONEである場合、空の値が何も格格納されます。暗号化後のこのプロパティの長さは**Track3EncyptedDataLength**プロパティに格納されます。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

参照　**DataEncryptionAlgorithm** プロパティ、**Track3EncryptedDataLength** プロパティ

Track3EncryptedDataLengthプロパティ　*第1.12版にて追加*

形式　LONG Track3EncryptedDataLength;

説明　暗号化される前の**Track3Data**プロパティの長さが格納されます。多くの暗号化アルゴリズムは暗号化される前の入力データのパディングを必要とします。このプロパティは暗号化される前のオリジナルの非パディングなデータの長さを含み、復号後オリジナルデータに戻す際に必要となります。

参照　**DataEncryptionAlgorithm** プロパティ、**Track3EncryptedData** プロパティ

Track4Data プロパティ　*第1.14.1版にて更新*

形式　BSTR Track4Data;

説明　最後に通されたカードのトラック4（JIS-II）の内容、あるいは空の文字列です。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

TransmitSentinels が FALSE なら、このプロパティには開始符号と終了符号を含まないトラックデータが入ります。

TransmitSentinels が TRUE なら、開始符号と終了符号を含みます。

DecodeDataプロパティがTRUEならば、“生(Raw)”フォーマットから変換されてこのプロパティにデータが入ります。

DataEncryptionAlgorithmプロパティがMSR_DE_NONEでない場合、以下のルールが追加で適用されます。

- **CapTrackDataMasking**プロパティがTRUEの場合、トラックデータに含まれる有意義と思われる全ての情報をマスクもしくは隠蔽しようと試みます。例えば、アカウントナンバーだけでなく、任意データまでをマスクもしくは隠蔽しようとします。可能であればトラックデータのオリジナルフォーマットを維持しようとする間に文字をマスクした文字に置換するため、正常に解析されます。
- **CapTrackDataMasking**プロパティがFALSEの場合、このプロパティには空の文字列が設定されます。

空の文字列は、トラックがアクセス可能でなかったことを示します。

以前のバージョンとの整合性を保つため、JIS-II データは、他の**TrackXData**プロパティにも格納されている可能性がありますが、アプリケーションの互換性を保つために、できるだけ**Track4Data**プロパティを使用することを推奨します。

参照　**Track1Data** プロパティ、**Track2Data** プロパティ、**Track3Data** プロパティ、**TransmitSentinels** プロパティ、**CapTrackDataMasking** プロパティ

Track4EncryptedDataプロパティ　*第1.14.1版にて更新*

形式　BSTR Track4EncryptedData;

説明　最後に通されたカードの暗号化されたトラック4データが格納されます。このプロパティは**DataEncryptionAlgorithm**プロパティがMSR_DE_NONEである場合、空の文字列が格納されます。暗号化後のこのプロパティの長さは**Track4EncyptedDataLength**プロパティに格納されます。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

参照　**DataEncryptionAlgorithm** プロパティ、**Track4EncryptedDataLength** プロパティ

Track4EncryptedDataLengthプロパティ　*第1.12版にて追加*

形式　LONG Track4EncryptedDataLength;

説明　暗号化される前の**Track4Data**プロパティの長さが格納されます。多くの暗号化アルゴリズムは暗号化される前の入力データのパディングを必要とします。このプロパティは暗号化される前のオリジナルの非パディングなデータの長さを含みます。復号後にオリジナルデータに戻す際に必要となります。

参照　**DataEncryptionAlgorithm** プロパティ、**Track4EncryptedData** プロパティ

TracksToRead プロパティ　R/W　　*第1.5版で更新*

形式　　LONG TracksToRead;

説明　　アプリケーションがカード操作によって**Track1Data**、**Track2Data**、**Track3Data**、および**Track4Data**プロパティに設定したいトラックデータを示します。

値	意味
MSR_TR_1	Track 1の内容。
MSR_TR_2	Track 2の内容。
MSR_TR_3	Track 3の内容。
MSR_TR_1_2	Track 1と2の内容。
MSR_TR_1_3	Track 1と3の内容。
MSR_TR_2_3	Track 2と3の内容。
MSR_TR_1_2_3	Track 1、2、と3の内容。
MSR_TR_4	Track 4.の内容。
MSR_TR_1_4	Track 1と4の内容。
MSR_TR_2_4	Track 2、と4の内容。
MSR_TR_3_4	Track 3、と4の内容。
MSR_TR_1_2_4	Track 1、2、と4の内容。
MSR_TR_1_3_4	Track 1、3、と4の内容。
MSR_TR_2_3_4	Track 2、3、と4の内容。
MSR_TR_1_2_3_4	Track 1、2、3と4の内容。

要求するトラック数を減らすと不要データのアクセス処理が無くなり、カードの読み取り成功率が上がり、多少応答性能がよくなります。

TracksToReadは、磁気ストライプリーダの読み取り能力を示すものではなく、その代わり、どのトラックを採取し、デコードし、返してほしいかをアプリケーションから指定できるプロパティです。たとえば、ISOタイプのカードをJIS-IIの読み取りヘッドで読ませる場合、カードに指定トラックがないというだけでなく、望みのデータ以外の採取を排除します。

このプロパティは**Open**メソッドでMSR_TR_1_2_3に初期化されます。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効なトラックの値が指定されました。

TracksToWrite プロパティ　R/W　　*第1.10版で追加*

形式　　LONG TracksToWrite;

説明　　**WriteTracks**メソッドが呼ばれMSRカードが通された時にMSRトラックに書き込まれるデータを保持します。有効な値は**CapWritableTracks**で定義されているものと同等かサブセットになります。
もし、**CapWritableTracks**にMSR_TR_NONEが設定されている場合、MSRトラックへの書込みはサポートされておらずこのプロパティを更新しようとするとOPOS_E_ILLEGALエラーが発生します。
CapWritableTracksで書込み可能と定義されていないトラックに対し設定を行おうとすると、プロパティは変更されずOPOS_E_ILLEGALエラーが発生します。このプロパティを設定する事により、それぞれのトラックが異なるエンコード限度があるため**EncodingMaxLength**も更新される可能性があります。

値	意味
MSR_TR_1	Track 1の書き込み。
MSR_TR_2	Track 2の書き込み。
MSR_TR_3	Track 3の書き込み。
MSR_TR_1_2	Track 1と2の書き込み。
MSR_TR_1_3	Track 1と3の書き込み。
MSR_TR_2_3	Track 2と3の書き込み。
MSR_TR_1_2_3	Track 1、2、と3の書き込み。
MSR_TR_4	Track 4の書き込み。
MSR_TR_1_4	Track 1と4の書き込み。
MSR_TR_2_4	Track 2、と4の書き込み。
MSR_TR_3_4	Track 3、と4の書き込み。
MSR_TR_1_2_4	Track 1、2、と4の書き込み。
MSR_TR_1_3_4	Track 1、3、と4の書き込み。
MSR_TR_2_3_4	Track 2、3、と4の書き込み。
MSR_TR_1_2_3_4	Track 1、2、3と4の書き込み。

このプロパティはOpenメソッドでMSR_TR_NONEに初期化されます。

戻り値 本プロパティ設定時、次の値のいずれかがResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効なトラックの値が指定されました。

参照 CapWritableTracks プロパティ、EncodingMaxLength プロパティ、 WriteTracks メソッド

TransmitSentinels プロパティ R/W *第1.5版で追加*

形式 BOOL TransmitSentinels;

説明 TRUEなら、Track1Data, Track2Data, Track3Data, と Track4Data に開始符号、終了符号を含みます。

FALSEなら、トラックデータプロパティには開始符号、終了符号は含みません。

このプロパティの値は、Openメソッドにより FALSEに初期化されます。

戻り値 このプロパティが設定されると、下記の値がResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_ILLEGAL	CapTransmitSentinels プロパティが FALSE である。

参照 CapTransmitSentinels プロパティ、Track1Data プロパティ 、Track2Data プロパティ、Track3Data プロパティ、Track4Data プロパティ

WriteCardTypeプロパティ　R/W　　*第1.12版にて追加*

形式　　BSTR WriteCardType;

説明　　**WriteTracks** メソッドが呼び出された後に用いられるカードの型が格納されます。このプロパティの値がアンダースコア"_"で始まっているならばベンダー仕様です。アンダースコアで始まっていないものは標準的なカードの型です。
以下に、現在標準的なカードと定義されているものを示します。

"BANK" 銀行のクレジット／デビットカード

"AAMVA"　AAMVA規格カード

このプロパティは、**Open**メソッドにより"BANK"に初期化されます。

参照　　**WriteTracks** メソッド

メソッド

AuthenticateDeviceメソッド *第1.14.1版にて更新*

形式 LONG AuthenticateDevice (BSTR *Rresponse*);

パラメータ	説明
Rresponse	以前呼び出したRetrieveDeviceAuthenticationDataメソッドから得られるチャレンジトークンから生成されたレスポンストークン。このデータのフォーマットは、BinaryConversion プロパティの値に依存します。詳細は、BinaryConversion を参照してください。

説明 デバイス認証するために、RetrieveDeviceAuthenticationDataメソッドを呼び出してチャレンジトークンをデバイスから得ます。アプリケーションは、このトークンを外部モジュールに渡します。その外部モジュールは秘密情報の特別な能力を持ち、適切なレスポンストークンを生成できるものです。このレスポンストークンは*Response*パラメータとしてこのメソッドに渡され、認証要求を検証するために使われます。このメソッドが成功すると、デバイスは認証状態に入り、DeviceAuthenticatedプロパティはTRUEにセットされます。

戻り値 次の値のいずれかが戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは成功しました。
OPOS_E_ILLEGAL	以下のいずれかの原因が考えられます。 ・デバイス認証をサポートしていません。 （CapDeviceAuthentication = MSR_DA_NOT_SUPPORTED） ・デバイスが既に認証状態です。
OPOS_E_ EXTENDED	ResultCodeExtended= EMSR_DEVICE_AUTHENTICATION_FAILED *Response*パラメータが無効のため、認証要求が失敗しました。

参照　　CapDeviceAuthentication プロパティ、DeviceAuthenticated プロパティ、RetrieveDeviceAuthenticationData メソッド

DeauthenticateDeviceメソッド　　*第1.14.1版にて更新*

形式　LONG DeauthenticateDevice (BSTR *Rresponse*);

パラメータ	説明
Rresponse	以前呼び出したRetrieveDeviceAuthenticationDataメソッドから得られるチャレンジトークンから生成したレスポンストークン。このデータのフォーマットは、BinaryConversion プロパティの値に依存します。詳細は、BinaryConversion を参照してください。

説明　このメソッドは既に認証状態（DeviceAuthenticated = TRUE）であるデバイスの認証状態を解除するために用いられます。レスポンストークンは、RetrieveDeviceAuthenticationDataメソッドにて得られたチャレンジトークンを外部モジュールに渡すことにより生成されます。このメソッドが成功したなら、DeviceAuthenticatedプロパティにはFALSEがセットされ、ステータスの値にMSR_SUE_DEVICE_DEAUTHENTICATEDがセットされ、StatusUpdateEventイベントが通知されます。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは成功しました。
OPOS_E_ILLEGAL	以下のいずれかの原因が考えられます。 ・デバイス認証をサポートしていません。（CapDeviceAuthentication = MSR_DA_NOT_SUPPORTED） ・デバイスが認証状態ではありません。
OPOS_E_ EXTENDED	ResultCodeExtended= EMSR_DEVICE_AUTHENTICATION_FAILED *Response*パラメータが無効のため、認証要求が失敗しました。

参照　CapDeviceAuthentication プロパティ、DeviceAuthenticated プロパティ、RetrieveDeviceAuthenticationData メソッド

RetrieveCardPropertyメソッド　*第1.12版で追加*

形式　LONG RetrieveCardProperty (BSTR *Name,* BSTR* *Value*);

パラメータ	説明
Name	値を取得するプロパティの名前です。**CardPropertyList**プロパティにより最後に通されたカードの有効なプロパティの集合を決定することができます。
Value	*Name*パラメータにより得られるプロパティの値が戻ります。*Name*パラメータが認識されない、もしくは、現在のカードの型にサポートされない場 合、値には何も返しません。

説明　最後に通されたカードから変換されたプロパティの値を取得します。カードが読み取られるまで全ての定義されたプロパティは空文字を返します。
注意：**DataEncryptionAlgorithm**プロパティがMSR_DE_NONEであり、かつ**CapTrackDataMasking**プロパティがTRUEならば戻り値にはマスクされた情報が含まれます。銀行のカードでは以下のプロパティに変換され、**RetrieveCardProperty**メソッドを通してアクセスされます。

銀行カードでは、次のプロパティに変換され、**RetrieveCardProperty**メソッドを通してアクセスされます。

“AccountNumber”

“ExpirationDate”

“FirstName”

“MiddleInitial”

“ServiceCode”

“Suffix”

“Surname”

“Title”

AAMVA規格カードは、次のプロパティに変換され、**RetrieveCardProperty**メソッドを通してアクセスされます。

“Address”

“BirthDate”

“City”

“Class”

“Endorsements”

“ExpirationDate”

“EyeColor”

“FirstName”

“Gender”

“HairColor”

“Height”

“LicenseNumber”

“PostalCode”

“Restrictions”

“State”

“Suffix”

“Surname”

“Weight”

このプロパティは、**Open**メソッドにより初期化されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは成功しました。
OPOS_E_ILLEGAL	最後に通されたカードの*Name*パラメータが無効、もしくは**ParseDecodeData**プロパティにFALSEがセットされています。

参照　**CardTypeList** プロパティ、**ParseDecodeData** プロパティ

RetrieveDeviceAuthenticationDataメソッド　*第1.14.1版にて更新*

形式　LONG RetrieveDeviceAuthenticationData (BSTR* *Challenge*);

パラメータ	説明
Challenge	認証解除を生成するデバイスによって生成されたチャレンジトークンが戻ります。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

説明　アプリケーションは、その後**AuthenticateDevice**メソッド、**DeauthenticatedDevice**メソッドに渡されるレスポンストークンを生成するのに用いられるチャレンジトークンを取得するためにこのメソッドを呼びます。チャレンジトークンは適切なレスポンストークンを生成する秘密情報の能力を持つ外部モジュールに送られます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは成功しました。
OPOS_E_ILLEGAL	デバイス認証をサポートしていません。（**CapDeviceAuthentication** = MSR_DA_NOT_SUPPORTED）

参照　**CapDeviceAuthentication** プロパティ、**AuthenticateDevice** メソッド、 **deauthenticateDevice** メソッド

UpdateKeyメソッド　　*第1.12版で追加*

形式　　LONG UpdateKey (BSTR *Key*, BSTR *KeyName*);

パラメータ	説明
Key	新しい鍵（16進ASCII値）です。
KeyName	鍵を識別するための名前です。

説明　　デバイスに新たな暗号鍵を提供します。暗号鍵は暗号化アルゴリズムでのみ用いられます。暗号化アルゴリズムでは暗号鍵の値はホストから標準的なメッセージとしてターミナルに送られます。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは成功しました。
OPOS_E_ILLEGAL	以下のいずれかの原因が考えられます。 ・選択された**DataEncryptionAlgorithm**プロパティは、この機能をサポートしていません。 ・*KeyName*が受け付けられません。 ・鍵が不正です。（非16進ASCII値、長さが間違っている、またはパリティが不正）

参照　　**CapDataEncryption** プロパティ、**DataEncryptionAlgorithm** プロパティ

WriteTracks メソッド　　*第1.14.1版にて更新*

形式　LONG WriteTracks (VARIANT *Data*, LONG *Timeout*);

パラメータ	説明
Data	書き込むデータを含んだ全トラックのバイナリ配列データを指定します。 トラック1を*最初の配列要素として*、常に４トラック分のバイナリデータを指定します。 書き込み不要トラックはデータ長０のバイナリデータを指定します。 本メソッドは、TracksToWriteプロパティに指定されたトラックに対してデータ書き込みを行います。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
Timeout	書き込みされるまでの最大待ち時間（ms）を示します。OPOS_FOREVER (-1)が指定されている場合は、書き込みが行われるまで本メソッドで待ちます。

説明　**TracksToWrite**プロパティで選択されているMSRのトラックに対しデータのエンコードを開始します。
呼び出し時、タイムアウトで設定された期間までカードが通された際書き込むデータを準備します。タイムアウト期間内にカードが通されなかった場合、エラーが返ります。
続いて、読み取られたカードに**WriteCardType**プロパティによって指定されるフォーマットで書きこまれます。
カードに書き込まれたデータはカードから再度全く同じ形式で読み取られます。サービスは、いかなる場合でもデータをデコード／エンコードしてはいけません。
このメソッドは常に同期で動作するため、次にカードが通された時データの書込みが試みられます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは成功しました。

OPOS_E_ILLEGAL	書き込みデータが**EncodingMaxLength**値を超えているか、**CapWritableTracks**がMSR_TR_NONEに設定されています。
OPOS_E_ FAILURE	指定した時間内にカード操作されましたが指定されたカード／トラックへの書き込みに失敗しました。
OPOS_E_ TIMEOUT	指定した時間内にカード操作されませんでした。（書き込みされませんでした）.

参照　**TracksToWrite** プロパティ、**WriteCardType** プロパティ、**EncodingMaxLength** プロパティ

イベント

DataEvent イベント　　*第1.12版で更新*

形式　　void DataEvent (LONG *Status*);

パラメータ	説明

Status　4つのトラックに関するバイト情報をLONG値で4分割して持っています。下の図はどのように *Status*が分割されているかを示します。

上位ワード		下位ワード	
上位バイト	下位バイト	上位バイト	下位バイト
トラック 4	トラック 3	トラック 2	トラック 1

トラックの値が0なら、指定トラックでのデータ読み込みはありません。これは、ハードウェアデバイスがそのトラックに対する読み取りヘッドを単に持たないだけか、アプリケーションが**TracksToRead**プロパティでそのトラックからの入力を意図的に排除しています。1以上なら、対応する**TrackXData**プロパティまたは暗号化が利用可能であれば**TrackxEncryptedData**プロパティのデータ長を示します。

説明　　デバイスから入力データが入ったことを示すためのイベント通知です。

イベントが通知される前に、カード操作されて入力されたデータは**TrackXData**プロパティおよび(または) **TrackxEncryptedData**プロパティに設定されます。もし、**DecodeData**がTRUEであればそのデータはデコードされます。また、**ParseDecodeData**がTRUEであれば、データは解析され各フィールドに対応するプロパティに設定されます。

参照　　**DecodeData**プロパティ、**ParseDecodeData**プロパティ、**TrackXData**プロパティ、**TrackxEncryptedData**プロパティ、**TracksToRead**プロパティ

ErrorEvent イベント

第1.10版で追加

形式 **void ErrorEvent (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。下記を参照のこと。
ResultCodeExtended	エラーイベントが生じた原因を示す拡張コードです。下記を参照のこと。
ErrorLocus	エラーの発生場所です。下記を参照のこと。
pErrorResponse	エラーイベントに対する処理を指定する領域へのポインタです。下記を参照のこと。

ErrorReportingTypeプロパティの値がMSR_ERT_CARDならば、*ResultCode*には次の値のいずれかが設定されます。

値	意味
OPOS_E_EXTENDED	**ResultCodeExtended** =OPOS_EMSR_START スタート符号エラー (Start sentinel error)。
	ResultCodeExtended = OPOS_EMSR_END エンド符号エラー (End sentinel error)。
	ResultCodeExtended = OPOS_EMSR_PARITY パリティエラー。
	ResultCodeExtended = OPOS_EMSR_LRC LRC エラー。

その他 **ResultCode**の項目を参照してください。

ErrorReportingTypeプロパティの値がMSR_ERT_TRACKならば、*ResultCode*には次の値のいずれかが設定されます。

なお、**ErrorReportingType**プロパティの値がMSR_ERT_TRACKで*ErrorLocus*がOPOS_EL_INPUT、*ResultCode*がOPOS_E_EXTENDEDの場合、*ResultCodeExtended*にはトラックレベルの状態が設定されます。

値	意味
OPOS_E_EXTENDED	**ResultCodeExtended** =下図のように分解されるトラックレベルステータス。

上位ワード		下位ワード	
上位バイト	下位バイト	上位バイト	下位バイト
トラック 4	トラック 3	トラック2	トラック 1

それぞれのトラックステータスは次のいずれかです。

OPOS_SUCCESS	エラーはありません。
OPOS_EMSR_START	スタート符号エラー (Start sentinel error)
OPOS_EMSR_END	エンド符号エラー (End sentinel error)
OPOS_EMSR_PARITY	パリティエラー
OPOS_EMSR_LRC	LRCエラー
OPOS_E_FAILURE	他のエラー

その他	**ResultCode**の項目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_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。すでにキューイングされたデータは利用できます。

*pErrorResponse*パラメータが示す位置にある内容は、*ErrorLocus*に基づいた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_CLEAR	キューイングされている入力データをクリアし、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、**DataEventEnabled**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、**DataEventEnabled**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

磁気ストライプデータの読み込み中にエラーが見つかったとき、イベントが発行されます。

正しいアプリケーションシーケンシングが行われるように、**DataEventEnabled**プロパティがTRUEに設定されるまで、入力エラーイベントは通知されません。

ErrorReportingTypeプロパティがMSR_ERT_CARDならば、エラーの発生したトラックまでは明示せず、トラックデータプロパティは変更されません。

ErrorReportingTypeプロパティがMSR_ERT_TRACKならば、*ResultCode*及び*ResultCodeExtended*パラメータによってトラックレベルステータスが示されます。また、トラックデータプロパティが**DataEvent**と共に変更されますが、エラーが生じたトラックには空の文字列が設定されます。

DataEventのように各トラックのデータ長は通知されません。しかしながら、アプリケーションは**TrackXData**の各プロパティを取得する事によりデータ長を決める事が出来ます。また、これは**ErrorEvent**なので**DataCount**プロパティの値は増加しません。そして、コントロールは**AutoDisable**プロパティの値に関わらずイネーブルのままです。

参照　「ステータス、結果コード、状態モデル」、ErrorReportingTypeプロパティ

StatusUpdateEvent イベント　*第1.12版で更新*

形式　void StatusUpdateEvent (LONG *Status*);

*Status*パラメータには、ステータス変化後の値が設定されます。

値	説明
MSR_SUE_DEVICE_AUTHENTICATED	デバイスは認証状態に入りました。
MSR_SUE_DEVICE_DEAUTHENTICATED	デバイスは認証状態ではありません。
*電源通知の**StatusUpdateEvent**の値*	「第2章 共通プロパティ、メソッド、イベント」の**StatusUpdateEvent**の記述を参照。

説明　ＭＳＲデバイスの変化を通知します。

第 8 章
POSプリンタ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.2	Boolean	R/W	*適用外*
BinaryConversion	1.2	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.0	String	R	Open
Claimed	1.0	Boolean	R	Open
DataCount	1.2	Long	R	*適用外*
DataEventEnabled	1.0	Boolean	R/W	*適用外*
DeviceEnabled	1.0	Boolean	R/W	Open & Claim
FreezeEvents	1.0	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.0	Long	R	Open
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.0	Long	R	--
ResultCodeExtended	1.0	Long	R	Open
State	1.0	Long	R	--
ControlObjectDescription	1.0	String	R	--
ControlObjectVersion	1.0	Long	R	--
ServiceObjectDescription	1.0	String	R	Open
ServiceObjectVersion	1.0	Long	R	Open
DeviceDescription	1.0	String	R	Open
DeviceName	1.0	String	R	Open
専用				
CapCharacterSet	1.1	Long	R	Open
CapConcurrentJrnRec	1.0	Boolean	R	Open

CapConcurrentJrnSlp	1.0	Boolean	R	Open
CapConcurrentPageMode	1.9	Boolean	R	Open
CapConcurrentRecSlp	1.0	Boolean	R	Open
CapCoverSensor	1.0	Boolean	R	Open
CapMapCharacterSet	1.7	Boolean	R	Open
CapTransaction	1.1	Boolean	R	Open
CapJrnPresent	1.0	Boolean	R	Open
CapJrn2Color	1.0	Boolean	R	Open
CapJrnBold	1.0	Boolean	R	Open
CapJrnDhigh	1.0	Boolean	R	Open
CapJrnDwide	1.0	Boolean	R	Open
CapJrnDwideDhigh	1.0	Boolean	R	Open
CapJrnEmptySensor	1.0	Boolean	R	Open
CapJrnItalic	1.0	Boolean	R	Open
CapJrnNearEndSensor	1.0	Boolean	R	Open
CapJrnUnderline	1.0	Boolean	R	Open
CapJrnCartridgeSensor	1.5	Long	R	Open
CapJrnColor	1.5	Long	R	Open
CapRecPresent	1.0	Boolean	R	Open
CapRec2Color	1.0	Boolean	R	Open
CapRecBarCode	1.0	Boolean	R	Open
CapRecBitmap	1.0	Boolean	R	Open
CapRecBold	1.0	Boolean	R	Open
CapRecDhigh	1.0	Boolean	R	Open
CapRecDwide	1.0	Boolean	R	Open
CapRecDwideDhigh	1.0	Boolean	R	Open
CapRecEmptySensor	1.0	Boolean	R	Open
CapRecItalic	1.0	Boolean	R	Open
CapRecLeft90	1.0	Boolean	R	Open
CapRecNearEndSensor	1.0	Boolean	R	Open
CapRecPapercut	1.0	Boolean	R	Open
CapRecRight90	1.0	Boolean	R	Open
CapRecRotate180	1.0	Boolean	R	Open
CapRecStamp	1.0	Boolean	R	Open
CapRecUnderline	1.0	Boolean	R	Open
CapRecCartridgeSensor	1.5	Long	R	Open
CapRecColor	1.5	Long	R	Open
CapRecMarkFeed	1.5	Long	R	Open
CapRecPageMode	1.9	Boolean	R	Open
CapRecRuledLine	1.13	Long	R	Open

CapSlpPresent	1.0	Boolean	R	Open
CapSlpFullslip	1.0	Boolean	R	Open
CapSlp2Color	1.0	Boolean	R	Open
CapSlpBarCode	1.0	Boolean	R	Open
CapSlpBitmap	1.0	Boolean	R	Open
CapSlpBold	1.0	Boolean	R	Open
CapSlpDhigh	1.0	Boolean	R	Open
CapSlpDwide	1.0	Boolean	R	Open
CapSlpDwideDhigh	1.0	Boolean	R	Open
CapSlpEmptySensor	1.0	Boolean	R	Open
CapSlpItalic	1.0	Boolean	R	Open
CapSlpLeft90	1.0	Boolean	R	Open
CapSlpNearEndSensor	1.0	Boolean	R	Open
CapSlpRight90	1.0	Boolean	R	Open
CapSlpRotate180	1.0	Boolean	R	Open
CapSlpUnderline	1.0	Boolean	R	Open
CapSlpBothSidesPrint	1.5	Boolean	R	Open
CapSlpCartridgeSensor	1.5	Long	R	Open
CapSlpColor	1.5	Long	R	Open
CapSlpPageMode	1.9	Boolean	R	Open
CapSlpRuledLine	1.13	Long	R	Open
AsyncMode	1.0	Boolean	R/W	Open
CartridgeNotify	1.5	Long	R/W	Open
CharacterSet	1.0	Long	R/W	Open, Claim, & Enable
CharacterSetList	1.0	String	R	Open
CoverOpen	1.0	Boolean	R	Open, Claim, & Enable
ErrorLevel	1.1	Long	R	Open
ErrorStation	1.0	Long	R	Open
ErrorString	1.1	String	R	Open
FontTypefaceList	1.1	String	R	Open
FlagWhenIdle	1.0	Boolean	R/W	Open
MapCharacterSet	1.7	Boolean	R/W	Open
MapMode	1.0	Long	R/W	Open
PageModeArea	1.9	String	R	Open
PageModeDescriptor	1.9	Long	R	Open
PageModeHorizontalPosition	1.9	Long	R/W	Open
PageModePrintArea	1.9	String	R/W	Open
PageModePrintDirection	1.9	Long	R/W	Open
PageModeStation	1.9	Long	R/W	Open
PageModeVerticalPosition	1.9	Long	R/W	Open

RotateSpecial	1.1	Long	R/W	Open
JrnLineChars	1.0	Long	R/W	Open, Claim, & Enable
JrnLineCharsList	1.0	String	R	Open
JrnLineHeight	1.0	Long	R/W	Open, Claim, & Enable
JrnLineSpacing	1.0	Long	R/W	Open, Claim, & Enable
JrnLineWidth	1.0	Long	R	Open, Claim, & Enable
JrnLetterQuality	1.0	Boolean	R/W	Open, Claim, & Enable
JrnEmpty	1.0	Boolean	R	Open, Claim, & Enable
JrnNearEnd	1.0	Boolean	R	Open, Claim, & Enable
JrnCartridgeState	1.5	Long	R	Open, Claim, & Enable
JrnCurrentCartridge	1.5	Long	R/W	Open, Claim, & Enable
RecLineChars	1.0	Long	R/W	Open, Claim, & Enable
RecLineCharsList	1.0	String	R	Open
RecLineHeight	1.0	Long	R/W	Open, Claim, & Enable
RecLineSpacing	1.0	Long	R/W	Open, Claim, & Enable
RecLineWidth	1.0	Long	R	Open, Claim, & Enable
RecLetterQuality	1.0	Boolean	R/W	Open, Claim, & Enable
RecEmpty	1.0	Boolean	R	Open, Claim, & Enable
RecNearEnd	1.0	Boolean	R	Open, Claim, & Enable
RecSidewaysMaxLines	1.0	Long	R	Open, Claim, & Enable
RecSidewaysMaxChars	1.0	Long	R	Open, Claim, & Enable
RecLinesToPaperCut	1.0	Long	R	Open, Claim, & Enable
RecBarCodeRotationList	1.0	String	R	Open
RecCartridgeState	1.5	Long	R	Open, Claim, & Enable
RecCurrentCartridge	1.5	Long	R/W	Open, Claim, & Enable
RecBitmapRotationList	1.7	String	R	Open
SlpLineChars	1.0	Long	R/W	Open, Claim, & Enable
SlpLineCharsList	1.0	String	R	Open
SlpLineHeight	1.0	Long	R/W	Open, Claim, & Enable
SlpLineSpacing	1.0	Long	R/W	Open, Claim, & Enable
SlpLineWidth	1.0	Long	R	Open, Claim, & Enable
SlpLetterQuality	1.0	Boolean	R/W	Open, Claim, & Enable
SlpEmpty	1.0	Boolean	R	Open, Claim, & Enable
SlpNearEnd	1.0	Boolean	R	Open, Claim, & Enable
SlpSidewaysMaxLines	1.0	Long	R	Open, Claim, & Enable
SlpSidewaysMaxChars	1.0	Long	R	Open, Claim, & Enable
SlpMaxLines	1.0	Long	R	Open, Claim, & Enable
SlpLinesNearEndToEnd	1.0	Long	R	Open, Claim, & Enable
SlpBarCodeRotationList	1.1	String	R	Open
SlpPrintSide	1.5	Long	R	Open, Claim, & Enable

SlpCartridgeState	1.5	Long	R	Open, Claim, & Enable
SlpCurrentCartridge	1.5	Long	R/W	Open, Claim, & Enable
SlpBitmapRotationList	1.7	String	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.0	--
Close	1.0	Open
ClaimDevice	1.0	Open
ReleaseDevice	1.0	Open & Claim
CheckHealth	1.0	Open, Claim, & Enable
ClearInput	1.0	*適用外*
ClearInputProperties	1.10	*適用外*
ClearOutput	1.0	Open & Claim
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.0	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用		
PrintNormal	1.0	Open, Claim, & Enable
PrintTwoNormal	1.0	Open, Claim, & Enable
PrintImmediate	1.0	Open, Claim, & Enable
BeginInsertion	1.0	Open, Claim, & Enable
EndInsertion	1.0	Open, Claim, & Enable
BeginRemoval	1.0	Open, Claim, & Enable
EndRemoval	1.0	Open, Claim, & Enable
CutPaper	1.0	Open, Claim, & Enable
RotatePrint	1.0	Open, Claim, & Enable
PrintBarCode	1.0	Open, Claim, & Enable
PrintBitmap	1.0	Open, Claim, & Enable
TransactionPrint	1.1	Open, Claim, & Enable
ValidateData	1.1	Open, Claim, & Enable
SetBitmap	1.0	Open, Claim, & Enable
SetLogo	1.0	Open, Claim, & Enable
ChangePrintSide	1.5	Open, Claim, & Enable
MarkFeed	1.5	Open, Claim, & Enable
ClearPrintArea	1.9	Open, Claim, & Enable

PageModePrint	1.9	Open, Claim, & Enable
PrintMemoryBitmap	1.10	Open, Claim, & Enable
DrawRuledLine	1.13	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.0	*適用外*
DirectIOEvent	1.0	Open, Claim
ErrorEvent	1.0	Open, Claim, & Enable
OutputCompleteEvent	1.0	Open, Claim, & Enable
StatusUpdateEvent	1.0	Open, Claim, & Enable

概説

POSプリンタコントロールのOLEプログラマティクIDは、“OPOS.POSPrinter”です。
POSプリンタコントロールは、一般的なWindowsグラフィカルプリンタを対象としておらず、パフォーマンスと利用の容易さのために、インターフェイスは直接プリンタをコントロールするように定義しています。通常、アプリケーションは、利用の容易さとエラーからの復帰における正確さのために、1行ごとにメソッドを呼び出して、プリンタに印刷します。

本仕様では、一般的に利用される三つのプリンタについて、定義しています。

Journal　トランザクション処理と動作情報についてのログをとるため、簡単なテキストで印字するのに利用されます。会計検査等の目的のために、店舗にて保管されます。

Receipt　トランザクション処理情報を印刷するために利用されます。たいてい、顧客に渡されます。しばしば、ストアレポートのためにも利用されます。取引間で用紙を切るためのカッターか、用紙を手動で切るためのバーのどちらかが付いています。

Slip　用紙上に情報を印刷するために利用されます。たいてい、顧客に渡されます。

用紙上にバリデーション情報を印刷するためにも利用されます。用紙タイプは、小切手やあるいは、クレジットカードスリップが典型的です。

用紙処理機能を制限することによって、レシートまたはジャーナルに組み込んで、バリデーション印刷が出来るようにしているものもあります。この場合、POSプリンタの用紙を処理するスロットの深さにより印刷行の数が制限される事がほとんどです。それにもかかわらず、POSプリンタコントロールは、このPOSプリンタの機能をスリップとして扱います。

機能　*第1.8版で更新*

POSプリンタは以下の機能を実装しています。

- デフォルトの文字設定では、0x20から0x7FのASCII文字を印刷できます。これは、スペース、数字、大文字、小文字といくつかの特種な文字を含んでいます。（もし、POSプリンタがこれらの全てをサポートしていなければ、小文字から大文字のように、近似した文字に、POSプリンタが変換する場合もあります。）

POSプリンタには、種々な付加的な機能があり、各機能の有無については、特定のまたは、その機能を示すプロパティを参照する必要があります。

また次の機能は、このバージョンのOPOS仕様書では定義しておらず、サービスオブジェクトで**DirectIO**機能を通じてそれらをサポートしている場合があります。

ダウンロード可能な文字設定。

代替文字。

ピクセルレベル印刷は、WidthパラメータをPTR_BM_ASISに設定し、PrintBitmap或いはSetBitmapメソッドが呼ばれた時のみに、ビットマップによってサポートされます。したがって、アプリケーションはプログラムに基づいて用意した必要とされるピクセルセットを印刷することが可能となります。

モデル

第1.13版で更新

POSプリンタは、一部拡張していますが、一般的な出力モデルに従います。

次のメソッドは、常に同期的に実行されます。
BeginInsertion、EndInsertion、BeginRemoval、EndRemoval、CheckHealth、ChangePrintSide
もし、非同期出力中ならば、これらのメソッドは失敗します。

次のメソッドも、常に同期的に実行されます。
PrintImmediate
PrintImmediateメソッドは、そのデータを直ちに印刷しようとします。（つまり、直後の印字動作に組み込まれます。）非同期出力中でも、このメソッドは呼び出せます。 PrintImmediateの主な目的は、非同期出力中に例外的に印字することです。

次のメソッドは、同期的または非同期的に実行されます。それは、AsyncModeプロパティの値に依存します。
PrintNormal、PrintTwoNormal、CutPaper、**DrawRuledLine**、RotatePrint、PrintBarCode、PrintBitmap、TransactionPrint、MarkFeed
AsyncModeがFALSEの時、これらのメソッドは同期的に実行され、完了ステータスをアプリケーションに返します。

AsyncModeがTRUEの時、これらのメソッドは次のように動作します。

- POSプリンタコントロールはその要求をプログラムメモリにバッファリングし、物理デバイスが受信処理可能になるとすぐに発行し、OutputIDプロパティをこの要求のための識別子として設定し、できるかぎり早く制御を戻します。デバイスがその要求を正常に完了した時、POSプリンタコントロールはOutputCompleteEventを通知します。このイベントのパラメータは、完了した要求のOutputIDを含みます。

  非同期印刷メソッドは、用紙切れやPOSプリンタ欠陥のようなハードの問題によるエラーステータスは返しません。これらのエラーは、**ErrorEvent**によってのみ通知されます。エラーステータスは、POSプリンタが排他アクセス権の獲得およびイネーブルをされていなかったり、パラメータが不正だったり、要求がキューに追加できなかった場合にのみ、返されます。最初の二つのケースは、アプリケーションのエラーです。一方、最後のケースは重大なシステムリソースの例外です。

- 非同期の要求が実行されている間に、エラーが発生すると、ErrorEventが通知されます。ErrorStationプロパティには、エラーが発生した時に印刷していたPOSプリンタが設定されます。*第1.11版以降*：ErrorLevelには、発生したエラーのレベルが設定されます。ErrorStringには、発生したエラーの詳細を示す文字列が設定されます。イベントハンドラはエラー要因を排除した後に、たとえば同期的な印刷メソッドを呼び（非同期的なメソッドではない）、終了していない出力を再度試みるかまたは削除することができます。
- POSプリンタコントロールは、非同期出力がFIFOで実行されることを保証しています。
- ClearOutputメソッドを呼ぶことによって、POSプリンタコントロールによって非同期出力を含むバッファリングされた全ての出力は、削除されます。OutputCompleteEventは、削除された出力に対しては通知されません。このメソッドは、（可能ならば）処理中の出力も中止します。
- FlagWhenIdleプロパティを設定することにより、処理中の出力が正常、削除にかかわらず全て終了した時に、StatusUpdateEventを通知させることができます。

第1.11版以降　－　一括処理（トランザクションモード）

トランザクションはユニットとしてステーションにプリントされる印刷オペレーションの順序です。一括処理に含まれる印刷操作は、PrintNormal、CutPaper、RotatePrint、PrintBarCode、PrintBitmap、DrawRuledLine、MarkFeedです。一括処理中、印刷操作は最初に有効性が確認されます。もし正当ならば、これら印刷操作は一括処理バッファに追加されますが、まだ印刷はされません。アプリケーションが必要とするいくつかの印刷操作を行った後、一括処理の印刷のメソッドが呼び出されます。

もし一括処理が同期的に印刷されるなら、戻り値は全ての一括処理の印刷が成功したこと、あるいはエラーが印刷の間に起こったことを表します。もし一括処理が非同期的に印刷されるならば、前記の非同期的な印刷の規則に従います。もしエラーが発生し、そしてエラーイベントのハンドラーがリトライの要求するならば、全ての一括処理の印刷はリトライされます。

POSプリンタのエラー通知モデルは以下の通りです。

POSプリンタの用紙切れとカバーオープン状態、およびカートリッジの状態は、ResultCodeにOPOS_E_EXTENDEDを設定することによって通知され、さらに、ResultCodeExtendedに以下のエラー状態の内の一つが設定されます。
OPOS_EPTR_JRN_EMPTY、OPOS_EPTR_REC_EMPTY、
OPOS_EPTR_SLP_EMPTY、
OPOS_EPTR_COVER_OPEN、
OPOS_EPTR_JRN_CARTRIDGE_REMOVED、
OPOS_EPTR_REC_CARTRIDGE_REMOVED、
OPOS_EPTR_SLP_CARTRIDGE_REMOVED、
OPOS_EPTR_JRN_CARTRIDGE_EMPTY、
OPOS_EPTR_REC_CARTRIDGE_EMPTY、
OPOS_EPTR_SLP_CARTRIDGE_EMPTY、
OPOS_EPTR_JRN_HEAD_CLEANING、
OPOS_EPTR_REC_HEAD_CLEANING、
OPOS_EPTR_SLP_HEAD_CLEANING。

他のPOSプリンタエラーは、ResultCodeにOPOS_E_FAILUREを設定するか、あるいは他の標準のエラーステータスを設定することによって、通知されます。これらの障害はPOSプリンタ欠陥や、ジャムや、その他の重大なエラーの場合が考えられます。

プリンタがイネーブルにされている間、プリンタの状態はモニターされ、変化はアプリケーションに通知されます。ほとんどのプリンタステータスは、**StatusUpdateEvent**とプリンタプロパティの更新によって、報告されます。以降のプロパティとイベントの章で定義されているステータスは、下記の通りです。

StatusUpdateEvent	プロパティ
PTR_SUE_COVER_OPEN	**CoverOpen** = TRUE
PTR_SUE_COVER_OK	**CoverOpen** = FALSE
PTR_SUE_JRN_EMPTY	**JrnEmpty** = TRUE
PTR_SUE_JRN_NEAREMPTY	**JrnNearEnd** = TRUE
PTR_SUE_JRN_PAPEROK	**JrnEmpty** = **JrnNearEnd** = FALSE
PTR_SUE_REC_EMPTY	**RecEmpty** = TRUE
PTR_SUE_REC_NEAREMPTY	**RecNearEnd** = TRUE
PTR_SUE_REC_PAPEROK	**RecEmpty** = **RecNearEnd** = FALSE
PTR_SUE_SLP_EMPTY	**SlpEmpty** = TRUE
PTR_SUE_SLP_NEAREMPTY	**SlpNearEnd** = TRUE

PTR_SUE_SLP_PAPEROK　　**SlpEmpty** = **SlpNearEnd** = FALSE

第1.5版以降

カートリッジステータスを取得するために２つのプロパティを使用します。**RecCurrentCartridge**等のプロパティで指定したカートリッジのステータスを、**RecCartridgeState**等のプロパティから取得できます。カートリッジの**StatusUpdateEvent**が示すステータスは、全てのカートリッジの内で、優先順位最上位のカートリッジステータスです。この場合、少なくともひとつのカートリッジステータスは、**StatusUpdateEvent**の対応するプロパティの値と一致しています。一方、他のカートリッジはより低い優先順位のステータスであるか、あるいは正常です。

PTR_SUE_JRN_CARTRIDGE_EMPTY
JrnCartridgeState = PTR_CART_EMPTY
or PTR_CART_REMOVED

PTR_SUE_JRN_HEAD_CLEANING
JrnCartridgeState = PTR_CART_CLEANING

PTR_SUE_JRN_CARTRIDGE_NEAREMPTY
JrnCartridgeState = PTR_CART_NEAREND

PTR_SUE_JRN_CARTRIDGE_OK
JrnCartridgeState = PTR_CART_OK

PTR_SUE_REC_CARTRIDGE_EMPTY
RecCartridgeState = PTR_CART_EMPTY
or PTR_CART_REMOVED

PTR_SUE_REC_HEAD_CLEANING
RecCartridgeState = PTR_CART_CLEANING

PTR_SUE_REC_CARTRIDGE_NEAREMPTY
RecCartridgeState = PTR_CART_NEAREND

PTR_SUE_REC_CARTRIDGE_OK
RecCartridgeState = PTR_CART_OK

PTR_SUE_SLP_CARTRIDGE_EMPTY
SlpCartridgeState = PTR_CART_EMPTY
or PTR_CART_REMOVED

PTR_SUE_SLP_HEAD_CLEANING
SlpCartridgeState = PTR_CART_CLEANING

PTR_SUE_SLP_CARTRIDGE_NEAREMPTY
SlpCartridgeState = PTR_CART_NEAREND

PTR_SUE_SLP_CARTRIDGE_OK
SlpCartridgeState = PTR_CART_OK

第1.8版以降

PTR_SUE_JRN_COVER_OPEN
CoverOpen = true

PTR_SUE_JRN_COVER_OK
CoverOpen = false if all covers closed;
CoverOpen = true if any other cover is open

PTR_SUE_REC_COVER_OPEN
CoverOpen = true

PTR_SUE_REC_COVER_OK
CoverOpen = false if all covers closed;
CoverOpen = true if any other cover is open

PTR_SUE_SLP_COVER_OPEN
CoverOpen = true

PTR_SUE_SLP_COVER_OK
CoverOpen = false if all covers closed;
CoverOpen = true if any other cover is open

第1.8版 － 説明

プリンタのスリップステーションの状態は、スリップの挿入と除去メソッド（**beginInsertion / endInsertion / beginRemoval / endRemoval**）から独立して報告されなければなりません。いくつかのアプリケーションが基本論理判断をプリンタの状態変化に基づいて行っているため、これは重要です。すなわち、アプリケーションは、スリップ用紙がスリップステーションの挿入口に置かれたことを認識した後にだけスリップの挿入を実行します。

一つの例を示します：合計キーが押された後、アプリケーションは支払いモードに入ります。実行する支払い方法を決定するために、周辺機器とキーボーのモニターが始まります。クレジットカード或いはデビッドカードがMSRで操作された場合に通知される**DataEvent**により、アプリケーションはクレジットカード或いはデビッドカードでの支払い処理を開始します。しかし、スリップ用紙がスリップステーションに置かれた場合に通知される**StatusUpdateEvent**或いは**SlpEmpty**プロパティの変化により、アプリケーションはチェックのMICR読取り処理を開始します。

スリップ用紙がスリップステーションの挿入口に置かれた時、プリンタはPTR_SUE_SLP_PAPEROKの**StatusUpdateEvent**を通知し、**SlpEmpty**と**SlpNearEnd**プロパティをFALSEに設定しなければなりません。アプリケーションは、その時、論理的に失敗しない前提で、**BeginInsertion**と**EndInsertion**メソッドを呼び出します。PTR_SUE_SLP_PAPEROKを受信したからといって、スリップ用紙の印刷が可能であると想定してはいけないことに注意してください。**BeginInsertion**と**EndInsertion**呼び出し成功の後にのみ、スリップ用紙の印刷が可能になります。

スリップ用紙がスリップステーションから取り除かれた時、プリンタはPTR_SUE_SLP_EMPTYの**StatusUpdateEvent**を通知し、**SlpEmpty**プロパティをTRUEに設定しなければなりません。**BeginInsertion**と**EndInsertion**メソッドのシーケンスが呼び出されていない場合でも、スリップステーションの挿入口からスリップ用紙を取り除くことは、前述と同じ状況となります。このメソッドシーケンスが正常に終了した場合、イベントとプロパティの変化は、一般的に**BeginRemoval**と**EndRemoval**メソッドシーケンスの後で起こります。しかし、スリップ用紙の長さを超えて印刷されたり、スリップ用紙が無理に取り除かれた場合にも、前述と同じ状況となります。

例外：BeginInsertionが呼び出されてプリンタの挿入口が開くまで、サービスがスリップ用紙の存在を検出することができない設計のプリンタもあります。この例外のほとんどは、スリップステーションが「バリデーション」型であることを示している**CapSlpFullslip**プロパティがFALSEである場合です。バリデーションステーションは、一般的にレシート／ジャーナルステーションと同じプリンタメカニズムを使用します。これら場合は、デバイスの制約条件を考慮し、スリップ状態のイベントをできるだけすぐに通知しなければなりません。

第1.5版以降　－　印刷カートリッジ

印刷カートリッジの機能は下記の通りです。

CapJrnCartridgeSensor、CapRecCartridgeSensor、CapSlpCartridgeSensorは、プリンタがカートリッジの状態を検出する機能をもっているかどうかを決定するために使用するプロパティです。

カートリッジの状態を参照する前に、以下のプロパティを使用して状態を参照したいカートリッジを選択します。
JrnCurrentCartridge、RecCurrentCartridge、SlpCurrentCartridge

以下のプロパティを使用する事により、現在選択されているカートリッジの状態を参照することができます。JrnCartridgeState,、RecCartridgeState、SlpCartridgeState
カートリッジが複数の状態を持つ場合は、以下の優先順位で値が設定されます。
PTR_CART_UNKNOWN、PTR_CART_REMOVED、PTR_CART_EMPTY、PTR_CART_CLEANING、PTR_CART_NEAREND、PTR_CART_OK

CapJrnColor、CapRecColor、CapSlpColorプロパティは、各ステーションのカラー機能を示します。

モノクロ

- CapJrnColor 、 CapRecColor 、 CapSlpColor プロパティ = PTR_COLOR_PRIMARY.

2色

- CapJrnColor 、 CapRecColor 、 CapSlpColor プロパティは、PTR_COLOR_PRIMARYとPTR_COLOR_CUSTOM1の論理和が設定されます。
- PTR_COLOR_CUSTOM1は、ESC|rCエスケープシーケンスを使用することにより印刷でき、その色は多くの場合赤色です。

マルチカスタムカラー

- CapJrnColor 、 CapRecColor 、 CapSlpColor プロパティは、PTR_COLOR_PRIMARYと下記の値の何れかとの論理和が設定されます。
  PTR_COLOR_CUSTOM1 、 PTR_COLOR_CUSTOM2 、 PTR_COLOR_CUSTOM3 、 PTR_COLOR_CUSTOM4 、 PTR_COLOR_CUSTOM5、PTR_COLOR_CUSTOM6
- カスタムカラーの選択は、ESC|#rCエスケープシーケンスを使用して行うことができます。

フルカラー

- CapJrnColor 、 CapRecColor 、 CapSlpColor プロパティは、PTR_COLOR_FULLと、下記の値の何れかとの論理和が設定されます。
  PTR_COLOR_CYAN 、 PTR_COLOR_MAGENTA 、 PTR_COLOR_YELLOW
- PTR_COLOR_FULLは、実際に搭載されるカートリッジを示すものではありません。フルカラー出力のためにカラーを混ぜる機能を示しています。
- フルカラー印刷は、ESC|#fCを使用することにより実現できます。

カスタムカラーを持つフルカラープリンタ

- CapJrnColor、CapRecColor、CapSlpColorプロパティは、マルチカスタムカラーとフルカラーの論理和が設定されます。

第1.5版以降 － DeviceEnableとカートリッジ状態レポート条件

印刷カートリッジ状態レポート モデルは下記の通りです。

CartridgeNotifyプロパティ:

StatusUpdateEventsイベント、JrnCartridgeState、RecCartridgeState、SlpCartridgeStateプロパティを使用したカートリッジ状態通知機能を有効にするために、アプリケーションがこのプロパティをセットします。このプロパティはデバイスが利用可能になる前にのみ設定することができます。（すなわちDeviceEnabledがTRUEにセットされる前）

この制限により、アプリケーションに影響を与えることなく、カートリッジの状態通知機能の実装が可能となっています。アプリケーションは、通知を受けるか否かのいずれかをプロパティに値を設定する事により簡単に選択する事がきます。このプロパティの値は下記のうちのいずれかです。 :

- PTR_CN_DISABLED
- PTR_CN_ENABLED

CapJrnCartridgeSensor、**CapRecCartridgeSensor**、**CapSlpCartridgeSensor**が0以外の値で、かつ**CartridgeNotify**がPTR_CN_ENABLEDの場合、下記の記述内容が、**DeviceEnabled**に追加されます。

コントロールがDeviceEnabledをFALSEからTRUEに変更すると、カートリッジの状態監視が開始されます。

コントロールがDeviceEnabledをTRUEからFALSEに変更すると、 カートリッジの状態監視を終了します。したがって：
JrnCartridgeState 、 RecCartridgeState 、 SlpCartridgeState は 、 PTR_CART_UNKNOWNに設定されます。

第1.8版以降　－　同期印刷　　第1.10版で更新

第1.7版以前、サーマルプリンタに代表される、ラインプリンタの同期モードでの動きは明確に定義されていませんでした。例えば、アプリケーションが**PrintNormal (PTR_S_RECEIPT**, “UnifiedPOS”)を呼び出した時、紙にテキストが印刷されるまで、メソッドはSUCCESSを返すべきではないと同期モデルに記述されていました。しかし、ラインプリンタの例では、紙送りまたはキャリッジリターンが印刷データに含まれているか、或いは印刷行がいっぱいになるまで、紙に印刷されません。第1.8版以降では、同期モードにおける**PrintNormal**、**PrintTwoNormal**、或いは**PrintImmediate**の各呼び出しでは、データを完全に印刷する（1行に満たない未印字の文字データが残ることはありえません）か、エラーを返すかのどちらかです。例えば、”UnifiedPOS¥n”（改行を伴うテキスト）や”¥x1B|3B”（3番で登録されたビットマップの印刷）を指定した場合は正しく印字されますが、”UnifiedPOS”（改行を伴わないテキスト）を指定した場合はエラーを返します。アプリケーションが、全ての印刷モードで、この内容を実施することを推奨します。

第1.9版以降　－　ページモード機能

ページモード機能は、一括印刷（トランザクションモード）と類似点があります。ページモード領域に対する一連の操作は、1つのトランザクションとして一括処理されます。ページモード機能は、**PrintNormal**、**PrintBitmap**、**PrintBarCode**メソッドと、新たに追加されたメソッドやプロパティを用いて、複雑な構成のレシートを印刷する手段をユーザに提供します。構成したレシートの印刷データを印刷、保存することができます。また、ページモードを終了するまでの間は構成を変更することもできます。

デバイスの共有

POSプリンタは以下のように排他的な利用をするデバイスです。

アプリケーションは、デバイスをイネーブルにする前に、そのデバイスの排他アクセス権を獲得しなければなりません。

アプリケーションは、大半のプリンタ特有のプロパティにアクセスする前に、そのデバイスの排他アクセス権を獲得し、イネーブルにしなければなりません。

アプリケーションは、デバイスを操作するメソッドを呼び出す前に、そのデバイスの排他アクセス権を獲得し、イネーブルにしなければなりません。

使用上の必要条件の詳細は、「一覧」の表を参照してください。

印字データとエスケープシーケンス　*第1.13版で更新*

全てのPOSプリンタのデフォルトの文字設定は、少なくとも0x20から0x7FのASCII文字をサポートすると仮定されます。これは、スペース、数字、大文字、小文字、いくつかの特種な文字を含んでいます。もし、プリンタが小文字をサポートしていなければ、サービスオブジェクトは小文字を大文字に変換する事があります。

あらゆるエスケープシーケンスは、後ろに垂直バー（'|'）が続くエスケープ文字ESC（この値は10進法で27）で始まります。この後ろには、0またはより大きな数、小文字のアルファベットが続きます。エスケープシーケンスは、大文字のアルファベットで終わります。

以下のエスケープシーケンス表中、代替文字 “#”は負でない数で形成される数字になります。エスケープシーケンスの一部がオプションの場合、この部分は大括弧“[...]”によって囲まれます。例えば、用紙カットのOPOSのエスケープシーケンスは “ESC|[#]P” であり、代替文字 “#”はオプションであることを意味します。したがって、“ESC|75P”(75%カット)と“ESC|P”(フルカット)とは共に有効なエスケープシーケンスです。

エスケープシーケンスがESC、垂直バー('|')、アスタリスク('*')で始まる場合、このシーケンスは大文字のアルファベットの区切り文字の後に可変長のデータを含みます。アスタリスクの後には、このデータの長さを指定する10進数の文字列が続きます。仮想的な例を以下に示します。
ESC |*6azQHELLO!
6文字の"HELLO!"でシーケンスが終わります。

エスケープシーケンスがESC、垂直バー('|')、感嘆符('!')で始まる場合、残っているシーケンスの効果が反転されます。本書ではこの機能が有効な場合、以下のように示します。
ESC|[!]bC
'!'が指定された場合は、ボールドを無効にします。

ESC“|”で始まらないエスケープシーケンス、あるいは ESC“|”で始まるがOPOSエスケープシーケンスに当てはまらないエスケープシーケンスの場合、サービスオブジェクトはPOSプリンタにそのまま渡そうとします。ただし、このようなエスケープシーケンスのすべてについて印字可能であるかどうかを識別するわけではないため、予期しない結果が発生する場合があります。

第1.7版以降 アプリケーションがESC|#Eエスケープシーケンスを利用すれば、プリンタへそのまま渡されたデータの総量について、より信頼性の高い取り扱いをすることが可能になります。このエスケープシーケンスを用いた場合、アプリケーションの移植性が低くなります。ただし、**ServiceObjectDescription**、**DeviceDescription**、**DeviceName**プロパティの値に基づいた条件分岐コードとして、埋め込みデータのエスケープシーケンスを用いれば、移植性を維持できます。
備考：本仕様書のこのコマンド定義とPoint Card Reader Writerの類似するコマンド定義のみが、デバイスの互換性を損なうものとして既知です。このコマンドを使用する必要性が発生した場合、その詳細な使用法と共にUnifiedPOSコミッティー(www.nrf-arts.org)に連絡して下さい。それによってUnifiedPOS標準の次版以降のリリースでは、より標準的/一般的なアプリケーションインターフェイスを導入することができます。アプリケーションレベルでデバイス非依存性と相互運用性を保つために、このコマンドシーケンスは“最後の手段”としてのみ使用して下さい。

アプリケーションは**ValidateData**メソッドを呼び出す事によって、エスケープシーケンスやデータが、プリンタで実行されることができるか判断する事ができます。（いくつかのエスケープシーケンスは、同等なプロパティを使用することによって同じように判断できます。）エスケープシーケンスのパラメーター領域違反、あるいはサポート外のパラメーター値が原因で、予測不可能な印字結果となることを避けるために、ValidateDataメソットを呼び出し、エスケープシーケンスの検証を行なうこと推奨します。

以下のエスケープシーケンスが定義されています。もし、エスケープシーケンスがプリンタによってサポートされないオペレーションを指定すれば、そのエスケープシーケンスは無視されます。

（1）指定時のみ、動作するエスケープシーケンス　*第1.10版で更新*

名称	データ	内容
用紙カット	ESC |[#]P	レシート用紙を切ります。代替文字’#’は、要求されたカットのパーセンテージを示すASCII10進数の文字列です。もし、’#’が省略されていれば、フルカットされます。たとえば、C言語の文字列“¥x1B|75P”は、75%の部分カットを要求しています。 指定したカット率をプリンタがサポートしていない場合、サービスの実装はハードウェアに最も適したカット動作を選択します。

フィードと用紙カット	ESC |[#]fP	**RecLinesToPaperCut**行分の紙送りをした後、レシート用紙をカットします。代替文字'#'は、「用紙カット」エスケープシーケンスで定義されています。
フィードと用紙カットとスタンプ印刷	ESC |[#]sP	**RecLinesToPaperCut**行分の紙送りをした後、レシート用紙のカットとスタンプ印刷を行います。代替文字'#'は、「用紙カット」エスケープシーケンスで定義されています。
ビットマップ印刷	ESC |#B	前もって保存されていたビットマップを印刷します。代替文字'#'はビットマップ番号です。
トップロゴ印刷	ESC |tL	前もって保存されていたトップロゴを印刷します。
ボトムロゴ印刷	ESC |bL	前もって保存されていたボトムロゴを印刷します。
スタンプ印刷	ESC |sL	スタンプを押します。
複数行フィード	ESC |[#]lF	複数行のフィードをします。代替文字'#'は、フィードする行数を示すASCII10進数の文字列です。もし、'#'が省略されていれば、1行分のフィードをします。
単位フィード	ESC |[#]uF	**MapMode**で指定された単位でフィードをします。代替文字'#'は、フィード量を示すASCII10進数の文字列です。もし、'#'が省略されていれば、1単位分のフィードをします。
逆フィード	ESC |[#]rF	複数行の逆フィードをします。代替文字'#'は、逆フィードする行数を示すASCII10進数の文字列です。もし、'#'が省略されていれば、1行分の逆フィードをします。
埋め込みデータの送信 （欄外の備考aを参照）	ESC |[*]#E	"#E"に続く文字列は、一切変更されずにデバイスへ渡されます。代替文字'#'は、ASCII10進数の文字列で、このエスケープシーケンスに続く、そのままデバイスへ渡されるべき文字列のバイト数を指定します。
バーコード印刷 （欄外の備考bを参照）	ESC|[*]#R	バーコードを印刷します。代替文字'#'は、ASCII10進数の文字列で、Rに続く文字列（バーコードの特性を定義）の文字数を指定します。詳細は欄外を参照してください。
罫線印刷 （欄外の備考cを参照）	ESC|*#dL	罫線を印刷します。代替文字'#'は、ASCII10進数の文字列で、dLに続く文字列（罫線の特性を定義）の文字数を指定します。詳細は欄外を参照してください。

備考a：このエスケープシーケンスは、第1.7版以降でのみ利用可能です。
備考b：このエスケープシーケンスは、第1.10版以降でのみ利用可能です。
備考c：このエスケープシーケンスは、第1.13版以降でのみ利用可能です。

バーコード印刷 ***第1.13版で更新***

第1.10版以降 アプリケーションはESC|[*]#Rエスケープシーケンスを利用することにより、他の印刷データとともに、バーコードを印刷することができます。代替文字'#'は、Rに続く文字列（バーコードの特性を定義）の文字数を指定します。

Rに続く文字列では、小文字のアルファベットと数値を使用して、バーコードの特性を指定します。数値には、**PrintBarCode**メソッドのために定義された定数値が利用できます。

属性を示す文字は以下の通りです。

s　symbology（バーコードタイプ）
h　height（バーコードの高さ）
w　width（バーコードの幅）
a　alignment（バーコードの位置）
t　text position（HRI文字列の位置）
d　start of data（バーコードデータの開始位置）
e　end of data（バーコードデータの終了位置）

属性は、**上記のリストで示された順番で必ず記述**しなければなりませんすべての属性は省略不可です。パラメーター領域違反、あるいはサポート外のパラメーター値といった、２つの条件のうちどちらか一方があっても、予測不可能な印字結果となります。

UPC-Aを、センタリング、バーコードの下にHRI文字列を印字、200dot高さ、400dot幅、の条件で印刷する場合の例を以下に示します。

ESC|33Rs101h200w400a-2t-13d123456789012e

第1.13版以降　以下も選択できます。

ESC|*33Rs101h200w400a-2t-13d123456789012e

※上記例で使用されている定数の定義をヘッダーファイルから抜粋したものを以下に示します。

```
const LONG PTR_BCS_UPCA        = 101;  //
Digits
const LONG PTR_BC_CENTER       = -2;
const LONG PTR_BC_TEXT_BELOW    = -13;
```

罫線印刷　　***第1.13版で追加***

第1.13版以降 アプリケーションはESC|*#dLエスケープシーケンスを利用することにより、他の印刷データとともに、罫線を印刷することができます。代替文字'#'は、dLに続く文字列（罫線の特性を定義）の文字数を指定します。

dLに続く文字列では、小文字のアルファベットと数値を使用して、罫線の特性を指定します。数値には、**DrawRuledLine**メソッドのために定義された定数値が利用できます。

属性を示す文字は以下の通りです。

p　position（位置）
d　direction（方向）
w　width（太さ）
s　style（線種）
c　color（色）

属性は、**上記のリストで示された順番で必ず記述**しなければなりませんすべての属性は省略不可です。パラメーター領域違反、あるいはサポート外のパラメーター値といった、２つの条件のうちどちらか一方があっても、予測不可能な印字結果となります。

横罫線を二重線、0ドット目から300ドット、太さ1ドット、赤色(Custom1の色)の条件で罫線を印刷する場合の例を以下に示します。

ESC|*14dLp0,300d1w1s2c1

(2) 印字中に、動作するエスケープシーケンス

明確に変更されるまで、その状態を保つ特徴があります。

名称	データ	内容
フォントタイプフェイス指定	ESC |#fT	以後のデータに対する新しいフォントタイプフェイスを選択します。代替文字'#'の値は、 0 = デフォルトフォントタイプ 1 = **FontTypefaceList**の最初のタイプを選択します。 2 = **FontTypefaceList**の2番目のタイプを選択します。 3以下も同様です。 与えられたフォントタイプフェイス番号が、FontTypefaceList プロパティに定義されたフォントタイプフェイス番号と異なる場合、予測不可能な印字結果になります。

（3）印字時に、動作するエスケープシーケンス　　*第1.12版で更新*

各印刷メソッドの終わりや、あるいは”ノーマル”シーケンスによってリセットされる特徴があります。

名称	データ	内容
ボールド	ESC |[!]bC	ボールド体または重ね打ちで印刷します。'!'が指定された場合は、ボールドを無効にします（欄外の備考cを参照）。
アンダーライン	ESC|[!][#]uC	アンダーラインと共に印刷します。代替文字'#'は、アンダーラインの太さ(ドット単位)を示すASCII10進数の文字列です。もし、'#'が省略されていれば、デフォルトの太さのアンダーラインが使用されます。"!"が指定されている場合、アンダーラインモードは解除されます。
イタリック	ESC |[!]iC	イタリック体で印刷します。'!'が指定された場合は、イタリックを無効にします（欄外の備考cを参照）。
カスタムカラー	ESC |[#]rC	カスタムカラーを指定して印刷します。代替文字'#'は、要求する色を示すASCII10進数の文字列です。10進の文字の値は、カートリッジの定数（プロパティで使用するカートリッジの番号）の値と同じです。もし、'#'が省略されていれば、カスタムカラー1（セカンダリーカラー）が使用されます。カスタムカラー1は、多くの場合赤です。与えられたカラーの値がサポート外のカートリッジ番号を指定した場合、予測不可能な印字結果となります。
反転文字	ESC |[!]rvC	明暗を反対にして印刷します。'!'が指定された場合は、反転文字を無効にします（欄外の備考cを参照）。
網掛け文字	ESC |[#]sC	網掛け装飾で印刷します。代替文字'#'は、網掛けのパーセンテージを示すASCII10進数の文字列です。もし、'#'が省略されていれば、プリンタのデフォルトレベルの網掛け装飾が使用されます。
縦横1倍角	ESC |1C	通常の大きさで印刷します。
横倍角	ESC |2C	横倍角文字で印刷します。
縦倍角	ESC |3C	縦倍角文字で印刷します。
縦横倍角	ESC |4C	縦横倍角文字で印刷します。

横倍率[7]	ESC |#hC	文字を横方向に拡大して印刷します。代替文字'#'は、横方向の拡大倍率を示すASCII10進数の文字列です。印字可能領域を超える拡大倍率文字を指定した場合、予測不可能な印字結果になります。
縦倍率	ESC |#vC	文字を縦方向に拡大して印刷します。代替文字'#'は、縦方向の拡大倍率を示すASCII10進数の文字列です。印字可能領域を超える拡大倍率文字を指定した場合、予測不可能な印字結果になります。
色指定 （欄外の備考aを参照）	ESC |[#]fC	'#'カラーで印刷します。代替文字'#'は、RGBを示すASCII10進数で置きかえられます。それらは、赤、緑、青を示す3桁です。例えば、 "255255000"は、黄色になります。 RGBのそれぞれの3桁の数値は0から255までを指定することができます。もし、'#'が省略されていれば、プリンタのデフォルトのカラーで印字されます。 ビットマップ印刷には影響しません。指定した値が、RGBで許容された範囲を超える場合、予測不可能な印字結果になります。
中央揃え	ESC |cA	中心に以下のテキストを整列させます。
右寄せ	ESC |rA	右に以下のテキストを整列させます。
左寄せ （欄外の備考bを参照）	ESC|lA	左に以下のテキストを整列させます。
ノーマル	ESC |N	プリンタの特徴を通常の状態に回復させます。
サブ スクリプト （欄外の備考aを参照）	ESC |[!]tbC	サブ スクリプト（下付き文字）で印字します。'!'が指定された場合は、サブスクリプトを無効にします（欄外の備考cを参照）。
スーパー スクリプト （欄外の備考aを参照）	ESC |[!]tpC	スーパー スクリプト（上付き文字）で印字します。'!'が指定された場合は、スーパー スクリプトを無効にします（欄外の備考cを参照）。

[7] Application Programmer's Guide日本拡張仕様書 第1.0版では、誤記により横倍率と縦倍率の説明が逆転していたものを修正しました。

取り消し線（欄外の備考dを参照）	ESC |[!][#]stC	取り消し線モードで印刷します。代替文字"#"は取り消し線の太さ（ドット単位）を示すASCII10進数の文字列です。もし、’#’が省略されていれば、デフォルトの太さの取り消し線となります。"!"が指定されている場合、取り消し線モードは解除されます。 与えられた太さがプリンタでサポートされた最大値を超えた場合、予測不可能な印字結果になります。

備考a：このエスケープシーケンスは、第1.5版以降でのみ利用可能です。
備考b：このエスケープシーケンスは、第1.10版以降でのみ利用可能です。
備考c：エスケープシーケンスを無効にする機能は、第1.10版以降でのみ利用可能です。

備考 d ：このエスケープシーケンスは、第1.12版以降でのみ利用可能です。

プロパティ

AsyncMode プロパティ　R/W

形式　BOOL AsyncMode;

説明　TRUE：　PrintNormal、PrintTwoNormal、CutPaper、RotatePrint、 PrintBarCode、PrintBitmap、DrawRuledLine、MarkFeed、TransactionPrintの印刷メソッドは、非同期に実行されます。
FALSE：これらのメソッドは同期的に印刷されます。

このプロパティは、Openメソッドにより FALSEに初期化されます。

戻り値　本プロパティ設定時、次の値がResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

CapCharacterSet プロパティ　　*第1.11版で追加、第1.5版で更新*

形式　　LONG CapCharacterSet;

説明　　プリンタの印刷可能な文字設定を示します。

このプロパティには、次に示す値の内の一つが入ります。

値	意味
PTR_CCS_ALPHA	文字設定は、大文字のアルファベットと、数値、空白、マイナス、ピリオドをサポートします。
PTR_CCS_ASCII	文字設定は、0x20と0x7Fの間の全てのASCII文字をサポートします。
PTR_CCS_KANA	文字設定は、コードページ932の一部をサポートします。それは、0xA1と0xDFの間の半角のカタカナと、0x20と0x7Fの間の全てのASCII文字を含んでいますが、漢字は含まれていません。
PTR_CCS_KANJI	文字設定は、コードページ932をサポートします。それは、0xA1と0xDFの間の半角のカタカナ、0x20と0x7Fの間の全てのASCII文字をサポートします。また、JIS第一水準、JIS第二水準レベルで定義される文字のシフトJISコード文字をサポートします。
PTR_CCS_UNICODE	文字設定は、Unicodeをサポートします。

デフォルトの文字設定は、これらの範囲の上位集合を含みます。初めの**CharacterSet**プロパティは、付加的な情報のために調べられます。

このプロパティは、**Open**メソッドにより初期化されます。

CapConcurrentJrnRec プロパティ

形式 BOOL CapConcurrentJrnRec;

説明 TRUE：ジャーナルとレシートの同時印刷が可能です。
PrintTwoNormalメソッドは、PTR_TWO_RECEIPT_JOURNALやPTR_S_JOURNAL_RECEIPTパラメータと共に使用できます。
FALSE：アプリケーションは、一時に2つのプリンタの内の1つだけに印刷し、プリンタ間の移動は最低限にするべきです。非同時印刷(Non-concurrent Printing)は、以下のような理由により必要とされます:

エラーの可能性がより高くなる。すなわち、異なるステーション間の移動により、紙ジャムの機会が増大する。

それぞれのステーションが別々に印字されるほうが、パフォーマンスが向上する。

このプロパティは、**Open**メソッドにより初期化されます。

CapConcurrentJrnSlp プロパティ

形式 BOOL CapConcurrentJrnSlp;

説明 TRUE：ジャーナルとスリップの同時印刷が可能です。
PrintTwoNormalメソッドは、PTR_TWO_SLIP_JOURNALやPTR_S_JOURNAL_SLIPパラメータと共に使用できます。
FALSE：アプリケーションは、ジャーナルに印刷を行う前には**BeginInsertion/EndInsertion**メソッドに続いてスリップへの印刷要求を行い、**BeginRemoval/EndRemoval**メソッドを行うという手順をとらなければなりません。 非同時印刷(Non-concurrent Printing)は、以下のような理由により必要とされます:

スリップフォームがジャーナルステーションの前方に位置しているといった物理的な制約がある。

エラーの可能性がより高くなる。すなわち、異なるステーション間の移動により、紙ジャムの機会が増大する。

それぞれのステーションに別々に印字されるほうが、パフォーマンスが向上する。

このプロパティは、**Open**メソッドにより初期化されます。

CapConcurrentRecSlp プロパティ

形式　BOOL CapConcurrentRecSlp;

説明　TRUE：レシートとスリップの同時印刷が可能です。
PrintTwoNormalメソッドは、PTR_TWO_SLIP_RECEIPTや
PTR_S_RECEIPT_SLIPパラメータと共に使用できます。
FALSE：アプリケーションは、 レシートに印刷を行う前には
BeginInsertion/EndInsertionメソッドに続いてスリップへの印刷要求を行い、**BeginRemoval/EndRemoval**メソッドを行うといった手順をとらなければなりません。 非同時印刷(Non-concurrent Printing)は、以下のような理由により必要とされます:

スリップフォームがレシートステーションの前方に位置しているといった物理的な制約がある。

エラーの可能性がより高くなる。すなわち、異なるステーション間の移動により、紙ジャムの機会が増大する。

それぞれのステーションに別々に印字されるほうが、パフォーマンスが向上する。

このプロパティは、**Open**メソッドにより初期化されます。

CapConcurrentPageMode プロパティ　*第1.9版で追加*

形式　BOOL CapConcurrentPageMode;

説明　TRUE：レシートとスリップの両方のステーションがページモード機能を持っていて、一方のステーションがページモードに遷移している間も、同時に別のステーションがページモードへ遷移することができます。この場合、ステーション毎に異なる構成のページを作成し、保存することができます。
FALSE：いずれのステーションもページ機能を持っていないか、ページモード機能を利用できるステーションが一つしか存在しません。もしくは、一方のステーションがページモードへ遷移している間は、別のステーションがページモードを開始することはできません。

このプロパティは、**Open**メソッドにより初期化されます。

CapCoverSensor プロパティ

形式　　BOOL CapCoverSensor;

説明　　TRUE：POSプリンタは「カバーオープンセンサ」を持ちます。
FALSE：持ちません。

このプロパティは、**Open**メソッドにより初期化されます。

CapJrn2Color プロパティ

形式　　BOOL CapJrn2Color;

説明　　TRUE：ジャーナルは2色印刷が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapJrnBold プロパティ

形式　　BOOL CapJrnBold;

説明　　TRUE：ジャーナルはボールド属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapJrnCartridgeSensorプロパティ　*第1.5版で追加*

形式　LONG CapJrnCartridgeSensor;

説明　このプロパティは、ジャーナル カートリッジ センサーの機能を示します。

CapJrnPresentプロパティがFALSEであれば、このプロパティには、“0”が設定されます。そうでなければ、次に示すいずれかの値の論理和が設定されます。

値	意味
PTR_CART_REMOVED	カートリッジが取り除かれていることを示すための機能があります。
PTR_CART_EMPTY	カートリッジが空であることを示すための機能があります。
PTR_CART_NEAREND	カラーカートリッジがニアエンドであることを示すための機能があります。
PTR_CART_CLEANING	ヘッドのクリーニング実行中を示す機能があります。

このプロパティは、Openメソッドにより初期化されます。

参照　JrnCartridgeStateプロパティ、JrnCurrentCartridgeプロパティ、CartridgeNotifyプロパティ

CapJrnColorプロパティ　　*第1.5版で追加*

形式　　LONG CapJrnColor;

説明　　ジャーナル カラー印刷機能を示します。

CapJrnPresentプロパティがFALSEであれば、このプロパティには、“0”が設定されます。そうでなければ、サポートするカラーを示します。

CapJrnColorプロパティは、次に示すいずれかの値の論理和が設定されます。

値	意味
PTR_COLOR_PRIMARY	プライマリ カートリッジが搭載されています。
PTR_COLOR_CUSTOM1	カスタム カートリッジ1が搭載されています。(セカンダリーカラーは、多くの場合 赤)
PTR_COLOR_CUSTOM2	カスタム カートリッジ2が搭載されています。
PTR_COLOR_CUSTOM3	カスタム カートリッジ3が搭載されています。
PTR_COLOR_CUSTOM4	カスタム カートリッジ4が搭載されています。
PTR_COLOR_CUSTOM5	カスタム カートリッジ5が搭載されています。
PTR_COLOR_CUSTOM6	カスタム カートリッジ6が搭載されています。
PTR_COLOR_CYAN	フルカラー印刷のためにシアンカートリッジがが搭載されています。
PTR_COLOR_MAGENTA	フルカラー印刷のためにマジェンダカートリッジが搭載されています。
PTR_COLOR_YELLOW	フルカラー印刷のためにイエロー カートリッジが搭載されています。
PTR_COLOR_FULL	フルカラー印字が可能です。

このプロパティは、Openメソッドにより初期化されます。

CapJrnDhigh プロパティ

形式　BOOL CapJrnDhigh;

説明　TRUE：ジャーナルは縦倍角属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapJrnDwide プロパティ

形式　BOOL CapJrnDwide;

説明　TRUE：ジャーナルは横倍角属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapJrnDwideDhigh プロパティ

形式　BOOL CapJrnDwideDhigh;

説明　TRUE：ジャーナルは縦横倍角属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapJrnEmptySensor プロパティ

形式　BOOL CapJrnEmptySensor;

説明　TRUE：ジャーナルは「用紙無しセンサ」を持ちます。
FALSE：持ちません。

このプロパティは、Openメソッドにより初期化されます。

CapJrnItalic プロパティ

形式　BOOL CapJrnItalic;

説明　TRUE：ジャーナルはイタリック属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapJrnNearEndSensor プロパティ

形式　BOOL CapJrnNearEndSensor;

説明　TRUE：ジャーナルは「用紙ニアエンドセンサ」を持ちます。
FALSE：持ちません。

このプロパティは、Openメソッドにより初期化されます。

CapJrnPresent プロパティ

形式　BOOL CapJrnPresent;

説明　TRUE：ジャーナルステーションが存在します。
FALSE：存在しません。

このプロパティは、Openメソッドにより初期化されます。

CapJrnUnderline プロパティ

形式　BOOL CapJrnUnderline;

説明　TRUE：ジャーナルのアンダーライン属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapMapCharacterSet プロパティ　*第1.7版で追加*

形式　BOOL CapMapCharacterSet;

説明　印字の際にアプリケーションの文字を選択した文字セットにマップするサービスオブジェクトの機能を示します。
TRUE：サービスオブジェクトは文字を**CharacterSetList**プロパティに定義された文字セットにマッピングすることが可能です。
FALSE：不可能です。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CharacterSet**プロパティ、**MapCharacterSet**プロパティ、**CharacterSetList**プロパティ

CapRec2Color プロパティ

形式　BOOL CapRec2Color;

説明　TRUE：レシートの2色印刷が可能です。
FALSE：不可能です。

このプロパティは、**Open**メソッドにより初期化されます。

CapRecBarCode プロパティ

形式　BOOL CapRecBarCode;

説明　TRUE：レシートのバーコード印刷が可能です。
FALSE：不可能です。

このプロパティは、**Open**メソッドにより初期化されます。

CapRecBitmap プロパティ

形式　BOOL CapRecBitmap;

説明　TRUE：レシートのビットマップ印刷が可能です。
FALSE：不可能です。

このプロパティは、**Open**メソッドにより初期化されます。

CapRecBold プロパティ

形式　　BOOL CapRecBold;

説明　　TRUE：レシートのボールド属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapRecCartridgeSensorプロパティ　*第1.5版で追加*

形式　　LONG CapRecCartridgeSensor;

説明　　このプロパティは、レシート カートリッジ センサーの機能を示します。

CapRecPresentプロパティがFALSEであれば、このプロパティには、“0”が設定されます。そうでなければ、次に示すいずれかの値の論理和が設定されます。

値	意味
PTR_CART_REMOVED	カートリッジが取り除かれていることを示すための機能があります。
PTR_CART_EMPTY	カートリッジが空であることを示すための機能があります。
PTR_CART_NEAREND	カラーカートリッジがニアエンドであることを示すための機能があります。
PTR_CART_CLEANING	ヘッドのクリーニング実行中を示す機能があります。

このプロパティは、Openメソッドにより初期化されます。

参照　　RecCartridgeStateプロパティ、RecCurrentCartridgeプロパティ、CartridgeNotifyプロパティ

CapRecColorプロパティ　　*第1.5版で追加*

形式　LONG CapRecColor;

説明　レシートカラー印刷機能を示します。

CapRecPresentプロパティがFALSEであれば、このプロパティには、“0”が設定されます。そうでなければ、サポートするカラーを示します。

CapRecColorプロパティは、次に示すいずれかの値の論理和が設定されます。

値	意味
PTR_COLOR_PRIMARY	プライマリ カートリッジが搭載されています。
PTR_COLOR_CUSTOM1	カスタム カートリッジ1が搭載されています。(セカンダリーカラーは、多くの場合 赤)
PTR_COLOR_CUSTOM2	カスタム カートリッジ2が搭載されています。
PTR_COLOR_CUSTOM3	カスタム カートリッジ3が搭載されています。
PTR_COLOR_CUSTOM4	カスタム カートリッジ4が搭載されています。
PTR_COLOR_CUSTOM5	カスタム カートリッジ5が搭載されています。
PTR_COLOR_CUSTOM6	カスタム カートリッジ6が搭載されています。
PTR_COLOR_CYAN	フルカラー印刷のためにシアンカートリッジがが搭載されています。
PTR_COLOR_MAGENTA	フルカラー印刷のためにマジェンダカートリッジが搭載されています。
PTR_COLOR_YELLOW	フルカラー印刷のためにイエロー カートリッジが搭載されています。
PTR_COLOR_FULL	フルカラー印字が可能です。

このプロパティは、Openメソッドにより初期化されます。

CapRecDhigh プロパティ

形式　BOOL CapRecDhigh;

説明　TRUE：レシートの縦倍角属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapRecDwide プロパティ

形式　BOOL CapRecDwide;

説明　TRUE：レシートの横倍角属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapRecDwideDhigh プロパティ

形式　BOOL CapRecDwideDhigh;

説明　TRUE：レシートの縦横倍角属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapRecEmptySensor プロパティ

形式　BOOL CapRecEmptySensor;

説明　TRUE：レシートの「用紙無しセンサ」を持ちます。
FALSE：持ちません。

このプロパティは、Openメソッドにより初期化されます。

CapRecItalic プロパティ

形式　BOOL CapRecItalic;

説明　TRUE：レシートのイタリック属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapRecLeft90 プロパティ

形式　BOOL CapRecLeft90;

説明　TRUE：レシートの左90度回転属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapRecMarkFeedプロパティ　　*第1.5版で追加*

形式　　LONG CapRecMarkFeed;

説明　　このプロパティは、マーク付き用紙の制御機能を示します。

CapRecMarkFeedプロパティは、次に示すいずれかの値の論理和が設定されます。
MarkFeedメソッドにこれらの値の1つを指定すると、その機能を実行することができます。

値	意味
PTR_MF_TO_TAKEUP	マーク付き用紙を、取り出せる位置までフィードできます。
PTR_MF_TO_CUTTER	マーク付き用紙を、自動カッターのカット位置の上までフィードできます。
PTR_MF_TO_CURRENT_TOF	現在のマーク付き用紙の先頭へフィードできます。（リバースフィード）
PTR_MF_TO_NEXT_TOF	次のマーク付き用紙の先頭へフィードできます。

CapRecMarkFeedプロパティが“0”であれば、マーク付き用紙の制御機能はサポートされていません。

このプロパティは、Openメソッドにより初期化されます。

参照　　MarkFeedメソッド

CapRecNearEndSensor プロパティ

形式　　BOOL CapRecNearEndSensor;

説明　　TRUE：レシートの「用紙ニアエンドセンサ」を持ちます。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapRecPapercut プロパティ

形式　BOOL CapRecPapercut;

説明　TRUE：レシートの用紙カット機能が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapRecPageMode プロパティ　*第1.9版で追加*

形式　BOOL CapRecPageMode;

説明　TRUE：レシートステーションにページモード機能があります。
FALSE：ありません。

このプロパティは、Openメソッドにより初期化されます。

CapRecPresent プロパティ

形式　BOOL CapRecPresent;

説明　TRUE：レシートステーションが存在します。
FALSE：存在しません。

このプロパティは、Openメソッドにより初期化されます。

CapRecRight90 プロパティ

形式　BOOL CapRecRight90;

説明　TRUE：レシートの右90度回転属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapRecRotate180 プロパティ

形式　BOOL CapRecRotate180;

説明　TRUE：レシートの180度回転属性が可能です。
FALSE：不可能です。

このプロパティは、**Open**メソッドにより初期化されます。

CapRecRuledLineプロパティ　*第1.13版で追加*

形式　Long CapRecRuledLine;

説明　レシートの印刷可能な罫線を示します。

次に示すいずれかの値の論理和が設定されます。

値	意味
PTR_RL_HORIZONTAL(=1)	横罫線を印刷する機能があります。
PTR_RL_VERTICAL(=2)	縦罫線を印刷する機能があります。

罫線印刷機能が無い場合は、このプロパティには“0”が設定されます。

CapRecPresentプロパティがFALSEであれば、このプロパティには、“0”が設定されます。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CapRecPresent**プロパティ、**DrawRuledLine**メソッド

CapRecStamp プロパティ

形式　BOOL CapRecStamp;

説明　TRUE：レシートのスタンプ印刷が可能です。
FALSE：不可能です。

このプロパティは、**Open**メソッドにより初期化されます。

CapRecUnderline プロパティ

形式　BOOL CapRecUnderline;

説明　TRUE：レシートのアンダーライン属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapSlp2Color プロパティ

形式　BOOL CapSlp2Color;

説明　TRUE：スリップの2色印刷が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapSlpBarCode プロパティ

形式　BOOL CapSlpBarCode;

説明　TRUE：スリップのバーコード印刷が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapSlpBitmap プロパティ

形式　BOOL CapSlpBitmap;

説明　TRUE：スリップのビットマップ印刷が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapSlpBold プロパティ

形式　　BOOL CapSlpBold;

説明　　TRUE：スリップのボールド属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapSlpBothSidesPrint プロパティ　*第1.5版で追加*

形式　　BOOL CapSlpBothSidesPrint;

説明　　TRUEであれば、スリップ印刷は、両面印字が可能です。可能でなければFALSEが設定されます。

このプロパティは、Openメソッドにより初期化されます。

CapSlpCartridgeSensorプロパティ　*第1.5版で追加*

形式　LONG CapSlpCartridgeSensor;

説明　このプロパティは、スリップ カートリッジ センサーの機能を示します。

CapSlpPresentプロパティがFALSEであれば、このプロパティには、“0”が設定されます。そうでなければ、次に示すいずれかの値の論理和が設定されます。

値	意味
PTR_CART_REMOVED	カートリッジが取り除かれていることを示すための機能があります。
PTR_CART_EMPTY	カートリッジが空であることを示すための機能があります。
PTR_CART_NEAREND	カラーカートリッジがニアエンドであることを示すための機能があります。
PTR_CART_CLEANING	ヘッドのクリーニング実行中を示す機能があります。

このプロパティは、Openメソッドにより初期化されます。

参照　SlpCartridgeStateプロパティ、SlpCurrentCartridgeプロパティ、CartridgeNotifyプロパティ

CapSlpColorプロパティ　　*第1.5版で追加*

形式　　LONG CapSlpColor;

説明　　スリップカラー印刷機能を示します。

CapSlpPresentプロパティがFALSEであれば、このプロパティには、“0”が設定されます。そうでなければ、サポートするカラーを示します。

CapSlpColorプロパティは、次に示すいずれかの値の論理和が設定されます。

値	意味
PTR_COLOR_PRIMARY	プライマリ カートリッジが搭載されています。
PTR_COLOR_CUSTOM1	カスタム カートリッジ1が搭載されています。(セカンダリーカラーは、多くの場合 赤)
PTR_COLOR_CUSTOM2	カスタム カートリッジ2が搭載されています。
PTR_COLOR_CUSTOM3	カスタム カートリッジ3が搭載されています。
PTR_COLOR_CUSTOM4	カスタム カートリッジ4が搭載されています。
PTR_COLOR_CUSTOM5	カスタム カートリッジ5が搭載されています。
PTR_COLOR_CUSTOM6	カスタム カートリッジ6が搭載されています。
PTR_COLOR_CYAN	フルカラー印刷のためにシアンカートリッジがが搭載されています。
PTR_COLOR_MAGENTA	フルカラー印刷のためにマジェンダカートリッジが搭載されています。
PTR_COLOR_YELLOW	フルカラー印刷のためにイエロー カートリッジが搭載されています。
PTR_COLOR_FULL	フルカラー印字が可能です。

このプロパティは、Openメソッドにより初期化されます。

CapSlpDhigh プロパティ

形式　BOOL CapSlpDhigh;

説明　TRUE：スリップの縦倍角属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapSlpDwide プロパティ

形式　BOOL CapSlpDwide;

説明　TRUE：スリップの横倍角属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapSlpDwideDhigh プロパティ

形式　BOOL CapSlpDwideDhigh;

説明　TRUE：スリップの縦横倍角属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapSlpEmptySensor プロパティ

形式　BOOL CapSlpEmptySensor;

説明　TRUE：スリップが「用紙無しセンサ」を持ちます。
FALSE：持ちません。

このプロパティは、Openメソッドにより初期化されます。

CapSlpFullslip プロパティ

形式　BOOL CapSlpFullslip;

説明　TRUE：スリップはフルスリップ型です。用紙一杯に印刷できます。
FALSE：スリップは「バリデーション」型です。この場合、印字行数は、通常制限されます。また、バリデーションスリップが使用されている間は、レシートやジャーナルへのアクセスができません。

このプロパティは、Openメソッドにより初期化されます。

CapSlpItalic プロパティ

形式　BOOL CapSlpItalic;

説明　TRUE：スリップのイタリック属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapSlpLeft90 プロパティ

形式　BOOL CapSlpLeft90;

説明　TRUE：スリップの左90度属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapSlpNearEndSensor プロパティ

形式　BOOL CapSlpNearEndSensor;

説明　TRUE：スリップの「用紙ニアエンドセンサ」が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapSlpPageMode プロパティ　*第1.9版で追加*

形式　BOOL CapSlpPageMode;

説明　TRUE：スリップステーションにページモード機能があります。
FALSE：ありません。

このプロパティは、Openメソッドにより初期化されます。

CapSlpPresent プロパティ

形式　BOOL CapSlpPresent;

説明　TRUE：スリップステーションが存在します。
FALSE：存在しません。

このプロパティは、Openメソッドにより初期化されます。

CapSlpRight90 プロパティ

形式　BOOL CapSlpRight90;

説明　TRUE：スリップの右90度回転属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapSlpRotate180 プロパティ

形式　BOOL CapSlpRotate180;

説明　TRUE：スリップの180度回転属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapSlpRuledLineプロパティ　*第1.13版で追加*

形式　Long CapSlpRuledLine;

説明　レシートの印刷可能な罫線を示します。

次に示すいずれかの値の論理和が設定されます。

値	意味
PTR_RL_HORIZONTAL(=1)	横罫線を印刷する機能があります。
PTR_RL_VERTICAL(=2)	縦罫線を印刷する機能があります。

罫線印刷機能が無い場合は、このプロパティには“0”が設定されます。

CapSlpPresentプロパティがFALSEであれば、このプロパティには、“0”が設定されます。

このプロパティは、Openメソッドにより初期化されます。

参照　CapSlpPresentプロパティ、DrawRuledLineメソッド

CapSlpUnderline プロパティ

形式　BOOL CapSlpUnderline;

説明　TRUE：スリップのアンダーライン属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapTransaction プロパティ　　*第1.11版で追加*

形式　　BOOL CapTransaction;

説明　　TRUE：一括処理が各ステーションで有効です。
FALSE：無効です。

このプロパティは、Openメソッドにより初期化されます。

CartridgeNotifyプロパティ　R/W　　*第1.5版で追加*

形式　　LONG CartridgeNotify;

説明　　このプロパティは、カートリッジの状態通知機能の有効/無効を示します。 このプロパティは、アプリケーションにより設定されます。

CartridgeNotifyの値は下記の通りです：

値	意味
PTR_CN_DISABLED	コントロールは、アプリケーションに対してカートリッジ状態通知を提供しません。カートリッジ状態通知に関する**StatusUpdateEvents**は通知されず、また**JrnCartridgeState**、**RecCartridgeState**、**SlpCartridgeState**プロパティには何も設定されません。
PTR_CN_ENABLED	**DeviceEnabled**がTRUEに設定されると、コントロールはカートリッジ状態通知に関する**StatusUpdateEvents**の通知や、**JrnCartridgeState**、**RecCartridgeState**、**SlpCartridgeState**プロパティの更新が始まります。発生するイベントやプロパティに設定される値は、 **CapJrnCartridgeSensor**プロパティ、 **CapRecCartridgeSensor**プロパティ、**CapSlpCartridgeSensor**プロパティの値に依存します。

CartridgeNotify プロパティは、デバイスがディセーブルの間、すなわち**DeviceEnabled**プロパティがFALSEの間のみ設定が可能です。

このプロパティは**Open**メソッドによりOPOS_CN_DISABLEDに初期化されます。この設定により、OPOSの従来バージョンとの互換性を保ちます。

戻り値　　本プロパティ設定時、次の値のいすれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	このプロパティの設定に成功しました。
OPOS_E_ILLEGAL	次のいずれかが発生しました。 • デバイスは既にイネーブルです。

- **CartridgeNotify**プロパティは、OPOS_CN_ENABLEDですが、**CapJrnCartridgeSensor**プロパティ、**CapRecCartridgeSensor**プロパティ、**CapSlpCartridgeSensor**プロパティは0です。

その他　　**ResultCode**の項目を参照してください。

参照　　CapJrnCartridgeSensor プロパティ、CapRecCartridgeSensor プロパティ、CapSlpCartridgeSensor プロパティ、JrnCartridgeState プロパティ、RecCartridgeState プロパティ、SlpCartridgeStateプロパティ

CharacterSet プロパティ　R/W　*第1.10版で更新*

形式　LONG CharacterSet;

説明　文字の印字に用いるキャラクタを設定します。

このプロパティは、**Open**メソッド後最初にデバイスをイネーブルにした時に初期化されます。

このプロパティには、次に示す値の内の一つが設定されます。

値	意味
Range 101 - 199	デバイス特有のキャラクタを設定します。コードページにもASCII、Windows ANSIにも当てはまりません。
Range 400 - 990	コードページを設定します。標準の内の一つと適合します。
PTR_CS_UNICODE	Unicodeを設定します。この定数の値は997です。
PTR_CS_ASCII	ASCIIキャラクタを設定します。0x20と0x7Fの間のASCIIキャラクタをサポートします。この定数値は998です。
PTR_CS_ANSI	PTR_CS_WINDOWSと同じ値を持っています。
PTR_CS_WINDOWS	Windows ANSIキャラクタを設定します。この定数値は999です。これは正確には、Windowsのコードページ1252と等しい。
Range 1000 and higher	Windowsのコードページ；標準値の一つと適合します。

このプロパティに関する追加情報については、付録Cの「キャラクターセットのマッピング」を参照してください。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	不正な値が使用されました。
その他	**ResultCode**の項目を参照してください。

参照　CharacterSetListプロパティ

CharacterSetList プロパティ

形式　BSTR CharacterSetList;

説明　キャラクタ設定番号の文字列。

このプロパティは、Openメソッドにより初期化されます。ASCII数字からなる番号設定の文字列はカンマによって区切られます。たとえば文字列が"101,850,999"で設定されれば、デバイス特有のキャラクタとコードページ850とWindows ANSI キャラクタがサポートされます。

参照　CharacterSetプロパティ

CoverOpen プロパティ

形式　BOOL CoverOpen;

説明　TRUE：POSプリンタのカバーが開いています。
FALSE：閉まっています。

CapCoverSensorプロパティがFALSEならば、POSプリンタはカバーオープンセンサを持ちません。そして、このプロパティは常にFALSEを返します。

このプロパティは、デバイスをイネーブルにしている間に初期化され、現状を保持します。

ErrorLevel プロパティ　　*第1.11版で追加*

形式　　LONG ErrorLevel;

説明　　エラー状態の重大さを示します。

このプロパティには、次に示す値の内の一つが設定されます。

値	意味
PTR_EL_NONE	エラー状態ではありません。
PTR_EL_RECOVERABLE	復帰可能エラーが発生しました。（例：用紙無しエラー。）
PTR_EL_FATAL	復帰不可能エラーが発生しました。（例：内部的なプリンタ欠陥。）

このプロパティは、**ErrorEvent**が通知される前に、コントロールにより設定されます。エラーが解除されれば、このプロパティはPTR_EL_NONEに変わります。

ErrorStation プロパティ

形式　　LONG ErrorStation;

説明　　エラーを検出した時に、印刷しているステーションを保持します。

このプロパティには、次に示す値の内の一つが設定されます。

PTR_S_JOUNAL,　PTR_S_RECEIPT,　PTR_S_SLIP,
PTR_S_JOUNAL_RECEIPT,　PTR_S_JOUNAL_SLIP,
PTR_S_RECEIPT_SLIP,
PTR_TWO_RECEIPT_JOURNAL,　PTR_TWO_SLIP_JPURNAL,
PTR_TWO_SLIP_RECEIPT

このプロパティは**ErrorEvent**が通知される前に設定されます。

ErrorString プロパティ　　*第1.11版で追加*

形式　BSTR ErrorString;

説明　現在のエラーについてのベンダー固有の記述を保持します。

このプロパティは、**ErrorEvent**が通知される前に、コントロールにより設定されます。この記述を利用しないならば、プロパティには空文字列が設定されます。エラーが解除されれば、このプロパティは空文字列に変わります。

FlagWhenIdle プロパティ　R/W

形式　BOOL FlagWhenIdle;

説明　TRUE：POSプリンタコントロールがアイドル状態であれば、**StatusUpdateEvent**を通知します。
FALSE：このイベントは通知されません。

このステータスイベントが通知されると、**FlagWhenIdle**は自動的にFALSEにリセットされます。

このプロパティを使用してステータスイベントを利用することにより、アプリケーションは全ての非同期出力が終了した時を知ることができます。出力が正常終了した時、または**ClearOutput**メソッドもしくは**ErrorEvent**を受け取ったイベントハンドラにより出力が削除された時に、イベントが通知されます。

もし、**FlagWhenIdle**プロパティをTRUEに設定する時に、**State**プロパティが既にOPOS_S_IDLEであれば、**StatusUpdateEvent**は直ちに通知されます。従って、非同期出力の終了とこのフラグの設定のすれ違いを気にせずに、アプリケーションはこのイベントを使用できます。

このプロパティは、**Open**メソッドによりFALSEに初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

FontTypefaceList プロパティ　　*第1.11版で追加*

形式　　BSTR FontTypefaceList;

説明　　プリンタによりサポートされるフォントやフォントタイプを指定する文字列です。

このプロパティは、Openメソッドにより初期化されます。フォントまたはフォントタイプからなる文字列はカンマによって区切られます。アプリケーションは、フォントタイプ指定のエスケープシーケンス(ESC|#fT)を使うことにより、プリンタに対するフォントまたはフォントタイプを選択します。文字"#"はリストの中のフォントまたはフォントタイプの番号(1、2など)に置き換わります。

日本では、このプロパティは「明朝」フォントと「ゴシック」フォントをしばしば含みます。その他のフォントやフォントタイプは、他の国でサポートされるかもしれません。

空文字列は、デフォルトのフォントタイプだけがサポートされていることを示します。

参照　　「**印字データとエスケープシーケンス**」

JrnCartridgeStateプロパティ　　*第1.5版で追加*

形式　　LONG JrnCartridgeState;

説明　　現在選択されているジャーナルのカートリッジ（インク、リボン、トナー）の状態を示します。

値は次の通りです。

値	意味
PTR_CART_UNKNOWN	下記のいずれかの理由により、カートリッジ状態を決定できません。 • **CapJrnCartridgeSensor**＝0 デバイスは、カートリッジ状態通知機能をサポートしていません。 • **CartridgeNotify** = PTR_CN_DISABLED カートリッジ状態通知機能は、ディセーブルです。 • **DeviceEnabled** = FALSE デバイスがイネーブルになるまで、カートリッジ状態監視は行われません。
PTR_CART_REMOVED	**JrnCurrentCartridge**プロパティで選択されているカートリッジは取り外されています。
PTR_CART_EMPTY	**JrnCurrentCartridge**プロパティで選択されているカートリッジは空です。
PTR_CART_CLEANING	**JrnCurrentCartridge**プロパティで選択されているカートリッジはクリーニングを行っています。
PTR_CART_NEAREND	**JrnCurrentCartridge**プロパティで選択されているカートリッジはニアエンドです。
PTR_CART_OK	**JrnCurrentCartridge**プロパティで選択されているカートリッジは正常です。

カートリッジの状態が上記の複数に合致する場合は、上記の定数の順で優先的に値が設定されます。このプロパティは、デバイスをイネーブルにしている間に初期化され、現状を保持します。

参照　JrnCurrentCartridgeプロパティ、CapJrnCartridgeSensor プロパティ、CartridgeNotify プロパティ

JrnCurrentCartridgeプロパティ R/W　*第1.8版で更新*

形式　LONG JrnCurrentCartridge;

説明　このプロパティは、現在選択されているジャーナルカートリッジを指定します。

このプロパティは、Openメソッド後最初にデバイスをイネーブルにした時に、初期化されます。CapJrnPresentプロパティがFALSEならば、このプロパティはゼロに初期化されます。そうでなければ、この値は、 CapJrnColorプロパティによって指定されたカラーカートリッジの１つでなければなりません。（ PTR_COLOR_FULLは設定できません。）

JrnCurrentCartridgeプロパティの設定は、 JrnCartridgeStateプロパティも更新するかもしれません。

戻り値　本プロパティ設定時、次の値のいずれかがResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	このプロパティの設定に成功しました。
OPOS_E_ILLEGAL	不正な値が指定されました。
その他	ResultCodeの項目を参照してください。

参照　CapJrnPresentプロパティ、JrnCartridgeStateプロパティ

JrnEmpty プロパティ

形式　BOOL JrnEmpty;

説明　TRUE：ジャーナルは用紙切れです。
FALSE：ジャーナルの用紙はあります。

CapJrnEmptySensorがFALSEならば、このプロパティの値は常にFALSEです。

このプロパティは、デバイスをイネーブルにしている間に初期化され、現状を保持します。

参照　JrnNearEndプロパティ

JrnLetterQuality プロパティ　R/W

形式　BOOL JrnLetterQuality;

説明　TRUE：ジャーナルは高品質モードで印字されます。
FALSE：ジャーナルは高速モードで印字されます。

双方向印字能力をもつPOSプリンタにおいて、カラム位置がより正確になるようにするためには、高品質モード（単一方向印字）に設定します。

もしMapModeがPTR_MM_DOTSであれば、本プロパティへの値の設定は、JrnLineWidth,JrnLineHeight,それにJrnLineSpacingの値も更新する可能性があります。（詳細については「MapModeプロパティ」の脚注を参照してください。）

このプロパティは、Openメソッド後最初にデバイスをイネーブルにした時に、FALSEに初期化されます。

戻り値　本プロパティ設定時、次の値がResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

JrnLineChars プロパティ　R/W

形式　　LONG JrnLineChars;

説明　　ジャーナルの一行に印字される半角文字の文字数（行文字幅）です。

この値が、サポートされている行文字幅に変われば、文字幅はその指定された値に設定されます。正確な幅がサポートできない場合、値がプリンタがサポートする最大値よりも小さいならば、サポートされている行文字幅の中で、指定された値より大きく、かつ最も近い値に設定されます。（例えば、36をセットしてプリンタがライン毎に30あるいは40のいずれかの文字数を印刷することができるならば、サービスオブジェクトは文字のサイズ「40」を選んで、それぞれのラインの上に最高36文字をプリントするべきです。）

もし、値がプリンタがサポートする最大値よりも大きく、文字幅がサポートできなければ、エラーが返されます。（例えば、42をセットしてプリンタがライン毎に30あるいは40のいずれかの文字数を印刷することができるならば、サービスオブジェクトはリクエストをサポートすることができません。）

JrnLineCharsを設定することにより、**JrnLineWidth**、**JrnLineHeight**、**JrnLineSpacing**も更新されるかもしれません。なぜなら、文字のピッチやフォントが変わるかもしれないからです。

JrnLineCharsの値は、**Open**メソッド後最初にデバイスをイネーブルにした時に、POSプリンタのデフォルトの行文字幅に初期化されます。

戻り値　　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	不正な行文字幅が指定されました。

参照　　**JrnLineCharsListプロパティ**

JrnLineCharsList プロパティ

形式　BSTR JrnLineCharsList;

説明　ジャーナルによりサポートされる行文字幅を含む文字列です。

このプロパティは、**Open**メソッドにより初期化されます。

文字列は、カンマによって分けられるASCII数字の設定から成ります。
たとえば、文字列が"32,36,40"ならば、ジャーナルは、32文字、36文字、40文字の行幅をサポートします。

参照　JrnLineCharsプロパティ

JrnLineHeight プロパティ　R/W

形式　LONG JrnLineHeight;

説明　ジャーナル印刷の行の高さです。**MapMode**で定義された単位で表記します。

現在の文字の幅と一緒にサポートできる高さに変更されれば、その行の高さがこの値に設定されます。もし、正確な高さがサポートされていなければ、その高さは最も近似のサポートされている値に設定されます。

JrnLineCharsが変わると、**JrnLineHeight**は、選択された幅に対するデフォルトの行高さに更新されます。

JrnLineHeightの値は、**Open**メソッド後最初にデバイスをイネーブルにした時に、POSプリンタのデフォルトの行高さに初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

JrnLineSpacing プロパティ　R/W

形式　LONG JrnLineSpacing;

説明　通常文字の印刷行の高さを示します。すなわち、印字行の高さと行間スペースの高さの両方を加えたものです。本プロパティは**MapMode**で定義された単位で表記します。

プリンタによってサポートされる行間に変更されれば、その行間はこの値に設定されます。もし、その行間がサポートされなければ、その行間は最も近似的にサポートされる値に設定されます。

JrnLineCharsまたは**JrnLineHeight**が変わると、**JrnLineSpacing**は、選択された幅あるいは高さに対するデフォルトの行間に更新されます。

JrnLineSpacingの値は、**Open**メソッド後最初にデバイスをイネーブルにした時に、POSプリンタのデフォルトの行間に初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

JrnLineWidth プロパティ

形式　LONG JrnLineWidth;

説明　**JrnLineChars**の一行の幅です。**MapMode**で定義された単位で表記します。

JrnLineCharsを設定すると、**JrnLineWidth**も更新されます。

JrnLineWidthの値は、デバイスをイネーブルにした時に、POSプリンタのデフォルトの行幅に初期化されます。

JrnNearEnd プロパティ

形式　BOOL JrnNearEnd;

説明　TRUE：ジャーナル用紙がニアエンドです。
FALSE：ニアエンドではありません。

もし、CapJrnNearEndSensorがFALSEならば、このプロパティの値は常にFALSEです。

このプロパティは、デバイスをイネーブルにした時に初期化され、イネーブルしている間、カレント値が保持されます。

参照　JrnEmptyプロパティ

MapCharacterSet プロパティ　R/W　*第1.7版で追加*

形式　BOOL MapCharacterSet;

説明　TRUE：印字の際にサービスオブジェクトはアプリケーションから渡された文字をCharacterSetプロパティで選択された文字セットにマップします。
FALSE：マッピングはサポートされません。この場合、アプリケーションはCharacterSetプロパティで選択された文字セットへのマップを保証する必要があります。

CapMapCharacterSetプロパティがFALSEならば、このプロパティの値は常にFALSEです。

このプロパティは、Openメソッドにより初期化されます。

参照　CharacterSetプロパティ、CapMapCharacterSetプロパティ

MapMode プロパティ　R/W

形式　　LONG MapMode;

説明　　プリンタのマッピングモードを示します。マッピングモードは他のプロパティで使用されるラインの高さや行間を示すような尺度の単位を定義します。

以下のマップモードをサポートしています。

値	意味
PTR_MM_DOTS	POSプリンタのドット幅。この幅は、各POSプリンタによって異なるかもしれません。[8]
PTR_MM_TWIPS	1インチの1/1440。
PTR_MM_ENGLISH	0.001インチ。
PTR_MM_METRIC	0.01ミリメータ。

MapModeを設定すると、**JrnLineHeight**、**JrnLineSpacing**、**JrnLineWidth**、**RecLineHeight**、**RecLineSpacing**、**RecLineWidth**、**SlpLineHeight**、**SlpLineSpacing**、**SlpLineWidth**も変化するかもしれません。

MapModeの値は、**Open**メソッド後最初にデバイスをイネーブルにした時に、PTR_MM_DOTSに初期化されます。POSプリンタ以外のデバイスでは、PTR_MM_DOTSに初期化されるとは限りません。

---

[8] OPOS POSプリンタの観点からすると”dot”は正確には定義付けできません。例えばあるプリンタは”half-dot”を高精細グラフィックス印刷の際に使われると定義し、またあるプリンタはテキスト印刷の際に使われるとしています。 OPOS POSプリンタのServiceObjectは以下の手段のうち１つを採用するものとします:

(a) 一貫して”dot”をプリンタの最小物理サイズ、すなわちハーフドットとします。

(b) ServiceObjectが**XxxLetterQuality**の設定に応じビットマップ印刷密度を変化させるのならば、ビットマップ密度に合うようにドットサイズは変化します。（すなわち、FALSEの場合はプリンタの物理ドットサイズ、TRUEの場合はハーフドットサイズです。） もしプリンタのテキスト印字方式がハーフドットサイズを基準としているような場合には、印字方式に対しての正確な値の適用が不可能な場合があるので、この選択を行うべきではありません。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	不正なマッピングモードが指定されました。

PageModeArea プロパティ　*第1.9版で追加*

形式　BSTR PageModeArea;

説明　**PageModeStation**プロパティに指定したステーションのページモード領域を、**MapMode**プロパティで指定した単位で保持します。ページモード領域は、ページモード印刷領域とは異なり、プリンタのハードウェア的な能力で決定されます。このプロパティは、ページモード領域の幅と高さを表すカンマ区切りの2つのASCII数字で構成されます。

例えば、文字列が"450,800"であれば、ページモード領域は幅450単位、高さ800単位です。そのステーションのページモード領域は、左上隅(0,0)と右下隅(449,799)で囲まれる長方形で示されます。

このプロパティにアクセスする前に、**PageModeStation**プロパティに有効なステーションを指定して下さい。有効なステーションが指定されていない場合、空文字列を返します。

参照　**MapMode**プロパティ、**PageModeStation**プロパティ

PageModeDescriptor プロパティ　*第1.9版で追加*

形式　LONG PageModeDescriptor;

説明　PageModeStationプロパティに指定したステーションで利用可能なページモード機能を、下記の値の論理和で示します。

値	意味
PTR_PM_BITMAP	PageModeStationプロパティに指定したステーションで、ビットマップ印刷ができます。
PTR_PM_BARCODE	PageModeStationプロパティに指定したステーションで、バーコード印刷ができます。
PTR_PM_BM_ROTATE	PageModeStationプロパティに指定したステーションで、ビットマップの回転印刷ができます。
PTR_PM_BC_ROTATE	PageModeStationプロパティに指定したステーションで、バーコードの回転印刷ができます。
PTR_PM_OPAQUE	文字と画像とその背景は、不透明体として処理されます。ページモード領域上にすでに配置された印刷データの上に、別の印刷データを重ねて配置すると、先に配置した印刷データは後で配置した印刷データに隠されます。

このプロパティにアクセスする前に、PageModeStationプロパティに有効なステーションを指定して下さい。有効なステーションが指定されていない場合、(0)を返します。

参照　PageModeStationプロパティ

PageModeHorizontalPosition R/W プロパティ　*第1.9版で追加*

形式　LONG PageModeHorizontalPosition;

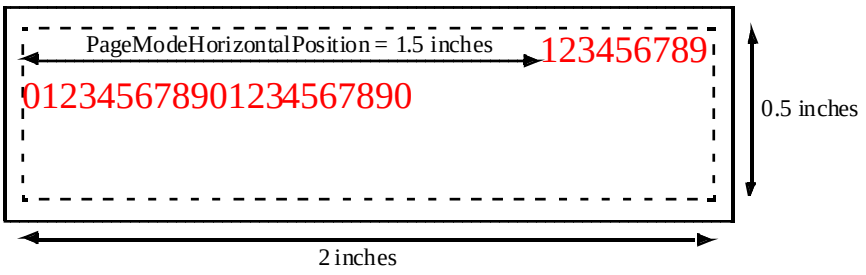
説明　**PageModeStation**に指定したステーションで、ページモード印刷領域内の印刷開始位置を水平方向に補正するためのオフセット値です。**MapMode**プロパティに指定した単位で表記します。水平方向とは、**PageModePrintDirection**プロパティで設定した印刷方向と同じ方向を指します。値を小刻みに指定することができないデバイスの場合、それに最も近い位置から印刷を開始します。

このプロパティは、現在位置ではなくて、最後に指定した水平方向のオフセット値を返します。このプロパティにアクセスする前に、**PageModeStation**プロパティに有効なステーションを指定して下さい。有効なステーションが指定されていない場合、(0)を返します。

PageModeHorizontalPositionプロパティの使い方を、以下のコード記述例で示します。

```
myptr.setMapMode(PTR_MM_ENGLISH);
myptr.setPageModeStation(PTR_S_RECEIPT);
myptr.pageModePrint(PTR_PM_PAGE_MODE);
// Set print area to 2 inches by 0.5 inches
myptr.setPageModePrintArea("0,0,2000,500");
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.setPageModeHorizontalPosition(1500);
myptr.printNormal(PTR_S_RECEIPT, "123456789012345678901234567890¥n");
```

上記のコードを記述した場合、下記のレシート印刷結果となります。



参照　**MapMode**プロパティ、**PageModePrintDirection**プロパティ、**PageModeStation**プロパティ

PageModePrintArea R/W プロパティ　*第1.9版で追加*

形式　BSTR PageModePrintArea;

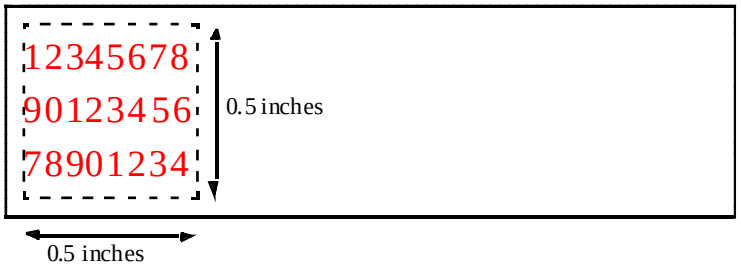
説明　PageModeStationプロパティに指定したステーションのページモード印刷領域を、MapModeプロパティで指定した単位で保持します。ページモード印刷領域のサイズは、ページモード領域より大きく取ることはできません。

このプロパティは、カンマ区切りの4つのASCII数字で構成されます。始点の水平方向座標、始点の垂直方向座標、水平方向の幅、垂直方向の高さ、の順に列挙します。例えば文字列が、"50,100,200,400"である場合、そのステーションのページモード印刷領域は、左上隅(50,100)と右下隅(249,499)で囲まれる長方形で示されます。このプロパティは、"0,0,0,0"に初期化されます。

ページモード印刷領域の右端を超える文字は、次の行に印刷されます。ページモード印刷領域の下端を超える文字と画像は、印刷されません。具体例を示します。

```
myptr.setMapMode(PTR_MM_ENGLISH);
myptr.setPageModeStation(PTR_S_RECEIPT);
myptr.pageModePrint(PTR_PM_PAGE_MODE);
// Set print area to half inch square block
myptr.setPageModePrintArea("0,0,500,500");
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789¥n");
```

上記のコードを記述した場合、下記のレシート印刷結果となります。



このプロパティにアクセスする前に、PageModeStationプロパティに有効なステーションを指定して下さい。有効なステーションが指定されていない場合、空文字列を返します。

参照　MapModeプロパティ、PageModeStationプロパティ

PageModePrintDirection R/W プロパティ　*第1.9版で追加*

形式　LONG PageModePrintDirection;

説明　ページモード印刷領域の印刷方向を保持します。印刷方向は以下です。

値	意味
PTR_PD_LEFT_TO_RIGHT	ページモード印刷領域の左上隅を始点に、左から右方向へ印刷します。通常の印刷方向です。
PTR_PD_BOTTOM_TO_TOP	ページモード印刷領域の左下隅を始点に、下から上方向へ印刷します。左90度の回転印刷です。
PTR_PD_RIGHT_TO_LEFT	ページモード印刷領域の右下隅を始点に、右から左方向へ印刷します。180度の回転印刷です。
PTR_PD_TOP_TO_BOTTOM	ページモード印刷領域の右上隅を始点に、上から下方向へ印刷します。右90度の回転印刷です。

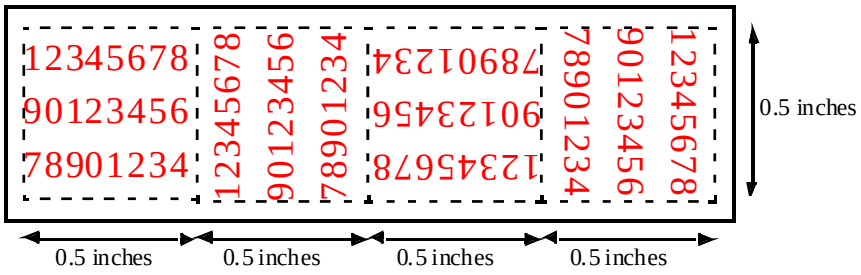
このプロパティは、**Open**メソッド実行後、最初にイネーブルされた時点で、PTR_PD_LEFT_TO_RIGHTに初期化されます。

このプロパティを変更すると、**PageModeHorizontalPosition**プロパティと**PageModeVerticalPosition**プロパティで示される印刷開始点の補正方向も変化します。このプロパティの変更は、現在編集中のページモード印刷領域に対してのみ効果があります。ページモード印刷領域を切り替えることで、文字の回転方向を組み合わせたレシートや単票を印刷することができます。具体例を示します。

```
myptr.setMapMode(PTR_MM_ENGLISH);
myptr.setPageModeStation(PTR_S_RECEIPT);
myptr.pageModePrint(PTR_PM_PAGE_MODE);
// Set print area to half inch square block
myptr.setPageModePrintArea("0,0,500,500");
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789O¥n");
myptr.setPageModePrintArea("500,0,500,500");
myptr.setPageModePrintDirection(PTR_PD_BOTTOM_TO_TOP);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789O¥n");
myptr.setPageModePrintArea("1000,0,500,500");
myptr.setPageModePrintDirection(PTR_PD_RIGHT_TO_LEFT);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789O¥n");
myptr.setPageModePrintArea("1500,0,500,500");
```

```
myptr.setPageModePrintDirection(PTR_PD_TOP_TO_BOTTOM);
myptr.printNormal(PTR_S_RECEIPT,“12345678901234567890123456789O¥n”);
```

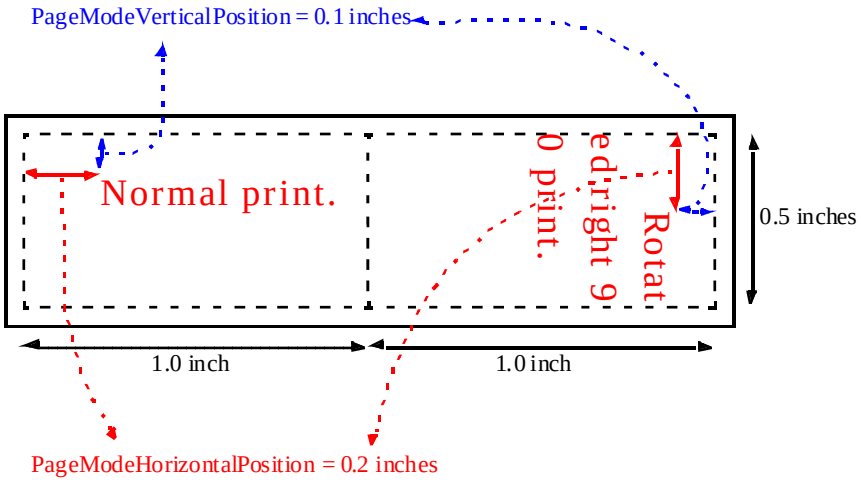
上記のコードを記述した場合、下記のレシート印刷結果となります。



同様に文字を回転させることもできます。

```
myptr.setMapMode(PTR_MM_ENGLISH);
myptr.setPageModeStation(PTR_S_RECEIPT);
myptr.pageModePrint(PTR_PM_PAGE_MODE);
myptr.pageModeVerticalPosition(100);
myptr.pageModeHorizontalPosition(200);
myptr.setPageModePrintArea(“0,0,1000,500”);
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.printNormal(PTR_S_RECEIPT, “Normal print.¥n”);
myptr.setPageModePrintArea(“1000,0,1000,500”);
myptr.setPageModePrintDirection(PTR_PD_TOP_TP_BOTTOM);
myptr.printNormal(PTR_S_RECEIPT, “Rotated right 90 print.¥n”);
myptr.setPageModePrint(PTR_PM_NORMAL);
```

上記のコードを記述した場合、下記のレシート印刷結果となります。



このプロパティにアクセスする前に、**PageModeStation**プロパティに有効なステーションを指定して下さい。有効なステーションが指定されていない場合、(0)を返します。

参照 **PageModeHorizontalPosition**プロパティ、**PageModeStation**プロパティ、**PageModeVerticalPosition**プロパティ

PageModeStation R/W プロパティ *第1.9版で追加*

形式 LONG PageModeStation;

説明 ページモード機能の対象とするステーションを指定します。このとき、ステーション毎に個別の値を保持しているプロパティは、その値が切り替わります。**PageModePrint**メソッドの対象となるステーションも切り替わります。一度に指定できるステーションは、PTR_S_RECEIPTかPTR_S_SLIPのいずれか一方のみです。ページモード機能を持つステーションが存在しなければ、0に初期化されます。2つ以上のステーションがページモード機能を持つ場合、このプロパティで切り替えてください。

このプロパティは、**Open**メソッドにより0に初期化されます。ページモード機能のプロパティやメソッドにアクセスする前に、必ずこのプロパティに有効なステーションを指定して下さい。

参照 **PageModePrint**メソッド

PageModeVerticalPosition R/W プロパティ *第1.9版で追加*

形式 LONG PageModeVerticalPosition;

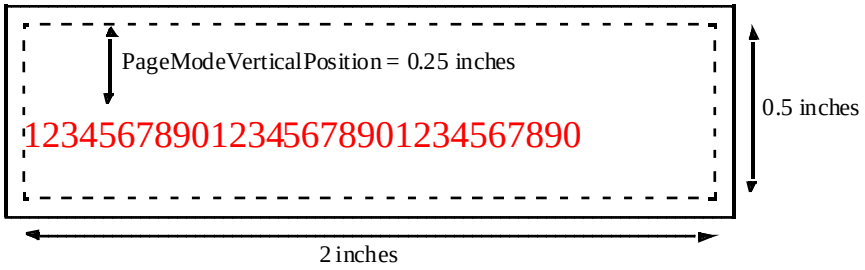
説明 **PageModeStation**に指定したステーションで、ページモード印刷領域内の印刷開始位置を垂直方向に補正するためのオフセット値です。**MapMode**プロパティに指定した単位で表記します。垂直方向とは、**PageModePrintDirection**プロパティで設定した印刷方向に対して垂直な方向です。値を小刻みに指定することができないデバイスの場合、それに最も近い位置から印刷を開始します。このプロパティは、現在位置ではなくて、最後に指定した垂直方向のオフセット値を返します。

PageModeVerticalPositionプロパティの使い方を、以下のコード記述例で示します。

```
myptr.setMapMode(PTR_MM_ENGLISH);
myptr.setPageModeStation(PTR_S_RECEIPT);
```

```
myptr.pageModePrint(PTR_PM_PAGE_MODE);
// Set print area to 2 inches by 0.5 inches
myptr.setPageModePrintArea("0,0,2000,500");
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.setPageModeVerticalPosition(250);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789O¥n");
```

上記のコードを記述した場合、下記のレシート印刷結果となります。



このプロパティにアクセスする前に、**PageModeStation**プロパティに有効なステーションを指定して下さい。有効なステーションが指定されていない場合、(0)を返します。

参照　　**MapMode**プロパティ、**PageModePrintDirection**プロパティ、**PageModeStation**プロパティ

RecBarCodeRotationList プロパティ　　*第1.11版で追加、第1.7版で更新*

形式　　BSTR RecBarCodeRotationList;

説明　　この文字列は、レシートのバーコードが回転できる方向を示します。

このプロパティは、 **Open**メソッドにより初期化されます。文字列は、カンマで区切られた回転方向を示す文字列群で構成されます。空文字列は、バーコード印刷がサポートされていないことを示します。回転方向を示す文字列は、

値	意味
0	バーコードは通常の方向に印刷できます。
R90	バーコードは右90度に回転印刷できます。
L90	バーコードは左90度に回転印刷できます。
180	バーコードは180度(倒立)に回転印刷できます。

例えば、文字列が”0,180”ならば、プリンタは通常のバーコード印刷と倒立したバーコード印刷が可能です。

参照　　RotateSpecialプロパティ; PrintBarCodeメソッド、RotatePrintメソッド

RecBitmapRotationList プロパティ　*第1.7版で追加*

形式　BSTR RecBitmapRotationList;

説明　この文字列は、レシートのビットマップが回転できる方向を示します。

このプロパティは、 Openメソッドにより初期化されます。文字列は、カンマで区切られた回転方向を示す文字列群で構成されます。空文字列は、ビットマップ印刷がサポートされていないことを示します。回転方向を示す文字列は、

値	意味
0	ビットマップは通常の方向に印刷できます。
R90	ビットマップは右90度に回転印刷できます。
L90	ビットマップは左90度に回転印刷できます。
180	ビットマップは180度(倒立)に回転印刷できます。

例えば、文字列が”0,180”ならば、プリンタは通常のビットマップ印刷と倒立したビットマップ印刷が可能です。

参照　PrintBitmapメソッド、RotatePrintメソッド

RecCartridgeStateプロパティ　　*第1.5版で追加*

形式　　LONG RecCartridgeState;

説明　　現在選択されているレシートのカートリッジ（インク、リボン、トナー）の状態を示します。

値は次の通りです。

値	意味
PTR_CART_UNKNOWN	下記のいずれかの理由により、カートリッジ状態を決定できません。 • **CapRecCartridgeSensor**＝0 デバイスは、カートリッジ状態通知機能をサポートしていません。 • **CartridgeNotify** = PTR_CN_DISABLED カートリッジ状態通知機能は、ディセーブルです。 • **DeviceEnabled** = FALSE デバイスがイネーブルになるまで、カートリッジ状態監視は行われません。
PTR_CART_REMOVED	**RecCurrentCartridge**プロパティで選択されているカートリッジは取り外されています。
PTR_CART_EMPTY	**RecCurrentCartridge**プロパティで選択されているカートリッジは空です。
PTR_CART_CLEANING	**RecCurrentCartridge**プロパティで選択されているカートリッジはクリーニングを行っています。
PTR_CART_NEAREND	**RecCurrentCartridge**プロパティで選択されているカートリッジはニアエンドです。
PTR_CART_OK	**RecCurrentCartridge**プロパティで選択されているカートリッジは正常です。

カートリッジの状態が上記の複数に合致する場合は、上記の定数の順で優先的に値が設定されます。

このプロパティは、デバイスをイネーブルにしている間に初期化され、現状を保持します。

参照　RecCurrentCartridgeプロパティ、CapRecCartridgeSensor プロパティ、CartridgeNotify プロパティ

RecCurrentCartridgeプロパティ R/W *第1.8版で更新*

形式　LONG RecCurrentCartridge;

説明　このプロパティは、現在選択されているレシートカートリッジを指定します。

このプロパティは、Openメソッド後最初にデバイスをイネーブルにした時に、初期化されます。CapRecPresentプロパティがFALSEならば、このプロパティはゼロに初期化されます。

そうでなければ、この値は、 CapRecColorプロパティによって指定されたカラーカートリッジの１つでなければなりません。
（ PTR_COLOR_FULLは設定できません。）

RecCurrentCartridgeプロパティの設定は、 RecCartridgeStateプロパティも更新するかもしれません。

戻り値　本プロパティ設定時、次の値のいずれかがResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	このプロパティの設定に成功しました。
OPOS_E_ILLEGAL	不正な値が指定されました。
その他	ResultCodeの項目を参照してください。

参照　CapRecPresentプロパティ、RecCartridgeStateプロパティ

RecEmpty プロパティ

形式　　BOOL RecEmpty;

説明　　TRUE：レシート用紙は紙切れです。
FALSE：レシート用紙はあります。

CapRecEmptySensorプロパティがFALSEならば、このプロパティの値は常にFALSEです。

このプロパティは、デバイスをイネーブルにしている間に初期化され、現状を保持します。

参照　　RecNearEndプロパティ

RecLetterQuality プロパティ　R/W

形式　BOOL RecLetterQuality;

説明　TRUE：レシートは高品質モードで印字されます。
FALSE：レシートは高速モードで印字されます。

双方向印字能力をもつPOSプリンタにおいて、カラム位置がより正確になるようにするには、高品質モード（単一方向印字）に設定します。

ビットマップは、高品質モードでは高密度のグラフィックで印刷され、高速モードでは低密度のグラフィックで印刷されます。

もしMapModeがPTR_MM_DOTSであれば、本プロパティへの値の設定は、RecLineWidth,RecLineHeight,それにRecLineSpacingの値も更新する可能性があります。（詳細については「MapModeプロパティ」の脚注を参照してください。）

このプロパティは、Openメソッド後最初にデバイスをイネーブルにした時に、FALSEに初期化されます。

戻り値　本プロパティ設定時、次の値がResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

参照　MapModeプロパティ、RecLineHeightプロパティ、RecLineSpacingプロパティ、RecLineWidthプロパティ

RecLineChars プロパティ　R/W

形式　LONG RecLineChars;

説明　レシートの一行に印字される半角文字の文字数（行文字幅）です。

この値が、サポートされている行文字幅に変われば、文字幅はその指定された値に設定されます。正確な幅がサポートできない場合、値がプリンタがサポートする最大値よりも小さいならば、サポートされている行文字幅の中で、指定された値より大きく、かつ最も近い値に設定されます。（例えば、36をセットしてプリンタがライン毎に30あるいは40のいずれかの文字数を印刷することができるならば、サービスオブジェクトは文字のサイズ「40」を選んで、それぞれのラインの上に最高36文字をプリントするべきです。）

もし、値がプリンタがサポートする最大値よりも大きく、文字幅がサポートできなければ、エラーが返されます。（例えば、42をセットしてプリンタがライン毎に30あるいは40のいずれかの文字数を印刷することができるならば、サービスオブジェクトはリクエストをサポートすることができません。）

RecLineCharsを設定することにより、**RecLineWidth**、**RecLineHeight**、**RecLineSpacing**も更新されるかもしれません。なぜなら、文字のピッチやフォントが変わるかもしれないからです。

RecLineCharsの値は、**Open**メソッド後最初にデバイスをイネーブルにした時に、POSプリンタのデフォルトの行文字幅に初期化されます。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	不正な行文字幅が指定されました。

参照　**RecLineCharsList**プロパティ

RecLineCharsList プロパティ

形式　BSTR RecLineCharsList;

説明　レシートによりサポートされる行文字幅を含む文字列です。

このプロパティは、**Open**メソッドにより初期化されます。

文字列は、カンマによって分けられるASCII数字の設定から成ります。
たとえば、文字列が"32,36,40"ならば、レシートは、32文字、36文字、40文字の行幅をサポートします。

参照　RecLineCharsプロパティ

RecLineHeight プロパティ　R/W

形式　LONG RecLineHeight;

説明　レシート印刷の行の高さです。**MapMode**で定義された単位で表記します。

現在の文字の幅と一緒にサポートできる高さに変更されれば、その行の高さがこの値に設定されます。もし、正確な高さがサポートされていなければ、その高さは最も近似のサポートされている値に設定されます。

RecLineCharsが変わると、**RecLineHeight**は、選択された幅に対するデフォルトの行高さに更新されます。

RecLineHeightの値は、**Open**メソッド後最初にデバイスをイネーブルにした時に、POSプリンタのデフォルトの行高さに初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

参照　RecLineCharsプロパティ

RecLineSpacing プロパティ　R/W

形式　LONG RecLineSpacing;

説明　通常文字の印刷行の高さを示します。すなわち、印字行の高さと行間スペースの高さの両方を加えたものです。プリンタの種類や現在の行間の値によっては、縦倍角文字はこの値を超えるかも知れません。本プロパティはMapModeで定義された単位で表記します。

POSプリンタによってサポートされる行間に変更されれば、その行間はこの値に設定されます。もし、その行間がサポートされなければ、その行間は最も近似的にサポートされる値に設定されます。

RecLineCharsまたはRecLineHeightが変わると、RecLineSpacingは、選択された幅あるいは高さに対するデフォルトの行間に更新されます。

RecLineSpacingの値は、Openメソッド後最初にデバイスをイネーブルにした時に、POSプリンタのデフォルトの行間に初期化されます。

戻り値　本プロパティ設定時、次の値がResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

RecLinesToPaperCut プロパティ

形式　LONG RecLinesToPaperCut;

説明　レシート用紙が切られる前に、進まなければならない行数を保持します。

CapRecPapercutがTRUEならば、これは用紙カット機構に着くまでの行数です。そうでなければ、これは手動の切断バーまでの行数です。

プロパティRecLineCharsやRecLineHeight、RecLineSpacingを変えることによって、このプロパティが変わるかもしれません。

このプロパティは、Openメソッド後最初にデバイスをイネーブルにした時に初期化されます。

RecLineWidth プロパティ

形式　　LONG RecLineWidth;

説明　　**RecLineChars**の一行の幅です。**MapMode**で定義された単位で表記します。

RecLineCharsを設定すると、**RecLineWidth**も更新されます。

RecLineWidthの値は、**Open**メソッド後最初にデバイスをイネーブルにした時に、POSプリンタのデフォルトの行幅に初期化されます。

RecNearEnd プロパティ

形式　　BOOL RecNearEnd;

説明　　TRUE：レシート用紙がニアエンドです。
FALSE：ニアエンドではありません。

もし、**CapRecNearEndSensor**がFALSEならば、このプロパティの値は常にFALSEです。

このプロパティは、デバイスをイネーブルにした時に初期化され、イネーブルしている間、カレント値が保持されます。

参照　　**RecEmpty**プロパティ

RecSidewaysMaxChars プロパティ

形式　LONG RecSidewaysMaxChars;

説明　サイドウェイモード（左90度または右90度回転印刷）において、各行に印刷される半角文字の最大文字数です。

CapRecLeft90とCapRecRight90の両方がFALSEならば、RecSidewaysMaxCharsは0です。

RecLineHeightとRecLineSpacing、RecLineCharsプロパティを変えることにより、このプロパティは変わるかもしれません。

このプロパティは、Openメソッド後最初にデバイスをイネーブルにした時に初期化されます。

参照　RecSidewaysMaxLinesプロパティ

RecSidewaysMaxLines プロパティ

形式　LONG RecSidewaysMaxLines;

説明　サイドウェイモード（左90度または右90度回転印刷）において、印刷される最大行数です。

CapRecLeft90とCapRecRight90の両方がFALSEならば、RecSidewaysMaxLinesは0です。

RecLineHeightとRecLineSpacing、RecLineCharsプロパティを変えることにより、このプロパティは変わるかもしれません。

このプロパティは、Openメソッド後最初にデバイスをイネーブルにした時に初期化されます。

参照　RecSidewaysMaxCharsプロパティ

RotateSpecial プロパティ　R/W　*第1.11版で追加*

形式　LONG RotateSpecial;

説明　バーコードの回転方向を示します。

このプロパティは、 **Open**メソッドによりPTR_RP_NORMALに初期化されます。

このプロパティには、次に示す値の内の一つが設定されます。

値	意味
PTR_RP_NORMAL	次のバーコードを通常の方向で印刷します。
PTR_RP_RIGHT90	右90度に回転印刷します。
PTR_RP_LEFT90	左90度に回転印刷します。
PTR_RP_ROTATE180	180度に回転印刷します。つまり、倒立して印刷します。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	不正なプロパティ値が指定されました。

参照　**PrintBarCode**メソッド

SlpBarCodeRotationList プロパティ　*第1.11版で追加、第1.7版で更新*

形式　BSTR SlpBarCodeRotationList;

説明　この文字列は、スリップのバーコードが回転できる方向を示します。

このプロパティは、 Openメソッドにより初期化されます。文字列は、カンマで区切られた回転方向を示す文字列群で構成されます。空文字列は、バーコード印刷がサポートされていないことを示します。回転方向を示す文字列は、

値	意味
0	バーコードは通常の方向に印刷できます。
R90	バーコードは右90度に回転印刷できます。
L90	バーコードは左90度に回転印刷できます。
180	バーコードは180度(倒立)に回転印刷できます。

例えば、文字列が”0,180”ならば、プリンタは通常のバーコード印刷と倒立したバーコード印刷が可能です。

参照　RotateSpecialプロパティ; PrintBarCodeメソッド、RotatePrintメソッド

SlpBitmapRotationList プロパティ　*第1.7版で追加*

形式　BSTR SlpBitmapRotationList;

説明　この文字列は、スリップのビットマップが回転できる方向を示します。

このプロパティは、 Openメソッドにより初期化されます。文字列は、カンマで区切られた回転方向を示す文字列群で構成されます。空文字列は、ビットマップ印刷がサポートされていないことを示します。回転方向を示す文字列は、

値	意味
0	ビットマップは通常の方向に印刷できます。
R90	ビットマップは右90度に回転印刷できます。
L90	ビットマップは左90度に回転印刷できます。
180	ビットマップは180度(倒立)に回転印刷できます。

例えば、文字列が”0,180”ならば、プリンタは通常のビットマップ印刷と倒立したビットマップ印刷が可能です。

参照　PrintBitmapメソッド、RotatePrintメソッド

SlpCartridgeStateプロパティ　　*第1.5版で追加*

形式　　LONG SlpCartridgeState;

説明　　現在選択されているスリップのカートリッジ（インク、リボン、トナー）の状態を示します。

値は次の通りです。

値	意味
PTR_CART_UNKNOWN	下記のいずれかの理由により、カートリッジ状態を決定できません。 • **CapSlpCartridgeSensor**＝0 デバイスは、カートリッジ状態通知機能をサポートしていません。 • **CartridgeNotify** = PTR_CN_DISABLED カートリッジ状態通知機能は、ディセーブルです。 • **DeviceEnabled** = FALSE デバイスがイネーブルになるまで、カートリッジ状態監視は行われません。
PTR_CART_REMOVED	**SlpCurrentCartridge**プロパティで選択されているカートリッジは取り外されています。
PTR_CART_EMPTY	**SlpCurrentCartridge**プロパティで選択されているカートリッジは空です。
PTR_CART_CLEANING	**SlpCurrentCartridge**プロパティで選択されているカートリッジはクリーニングを行っています。
PTR_CART_NEAREND	**SlpCurrentCartridge**プロパティで選択されているカートリッジはニアエンドです。
PTR_CART_OK	**SlpCurrentCartridge**プロパティで選択されているカートリッジは正常です。

カートリッジの状態が上記の複数に合致する場合は、上記の定数の順で優先的に値が設定されます。

このプロパティは、デバイスをイネーブルにしている間に初期化され、現状を保持します。

参照　SlpCurrentCartridgeプロパティ、CapSlpCartridgeSensor プロパティ、CartridgeNotify プロパティ

SlpCurrentCartridgeプロパティ R/W　*第1.8版で更新*

形式　LONG SlpCurrentCartridge;

説明　このプロパティは、現在選択されているスリップカートリッジを指定します。

このプロパティは、Openメソッド後最初にデバイスをイネーブルにした時に、初期化されます。CapSlpPresentプロパティがFALSEならば、このプロパティはゼロに初期化されます。

そうでなければ、この値は、 CapSlpColorプロパティによって指定されたカラーカートリッジの１つでなければなりません。
（ PTR_COLOR_FULLは設定できません。）

SlpCurrentCartridgeプロパティの設定は、 SlpCartridgeStateプロパティも更新するかもしれません。

戻り値　本プロパティ設定時、次の値のいずれかがResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	このプロパティの設定に成功しました。
OPOS_E_ILLEGAL	不正な値が指定されました。
その他	ResultCodeの項目を参照してください。

参照　CapSlpPresentプロパティ、SlpCartridgeStateプロパティ

SlpEmpty プロパティ

形式　BOOL SlpEmpty;

説明　TRUE：スリップ用紙が設定されていません。
FALSE：スリップ用紙はあります。

CapSlpEmptySensorプロパティがFALSEならば、このプロパティの値は常にFALSEです。

このプロパティは、デバイスをイネーブルにしている間に初期化され、現状を保持します。

備考　スリップ用紙無しセンサは、印刷する前にスリップ用紙が挿入されているかどうかを判定するために用いられ、さらに、スリップ用紙が定位置に留まっていることを監視できます。このセンサは通常、スリップのプリンタヘッド位置より一行以上上に設置されます。

逆に、スリップ用紙ニアエンドセンサ（設置されていれば）は、スリップ用紙の終了が近づいたことを判定するために用いられます。このセンサは通常、スリップのプリンタヘッド位置より一行以上下に設置されます。

参照　SlpNearEndプロパティ

SlpLetterQuality プロパティ　R/W

形式　　BOOL SlpLetterQuality;

説明　　TRUE：スリップは高品質モードで印字されます。
FALSE：スリップは高速モードで印字されます。

双方向印字能力をもつPOSプリンタにおいて、カラム位置がより正確になるようにするには、高品質モード（単一方向印字）に設定します。

ビットマップは、高品質モードでは高密度のグラフィックで印刷され、高速モードでは低密度のグラフィックで印刷されます。

もしMapModeがPTR_MM_DOTSであれば、 本プロパティへの設定は、SlpLineWidth,SlpLineHeight,それにSlpLineSpacingも更新する可能性があります。（詳細については「MapModeプロパティ」の脚注を参照してください。）

このプロパティは、Openメソッド後最初にデバイスをイネーブルにした時に、FALSEに初期化されます。

戻り値　　本プロパティ設定時、次の値のいずれかがResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

SlpLineChars プロパティ　R/W

形式　LONG SlpLineChars;

説明　スリップの一行に印字される半角文字の文字数（行文字幅）です。

この値が、サポートされている行文字幅に変われば、文字幅はその指定された値に設定されます。正確な幅がサポートできない場合、値がプリンタがサポートする最大値よりも小さいならば、サポートされている行文字幅の中で、指定された値より大きく、かつ最も近い値に設定されます。（サービスオブジェクトは、スリップ用紙が差し込まれたスリップテーブルの側面に最も近いカラム位置で、リクエストされた文字列を印刷するべきです。例えば、オペレーターがテーブル側面に対して右端にスリップを差し込んで、**SlpLineChars** が36にセットされ、そしてプリンタがライン毎に60文字を印刷するならば、サービスオブジェクトは左端マージンにおいて24のスペースを加え、25から60までのカラムで文字列を印刷するべきです。）

もし、値がプリンタがサポートする最大値よりも大きく、文字幅がサポートできなければ、エラーが返されます。（例えば、65をセットしてプリンタがライン毎に60の文字数を印刷することができる場合、サービスオブジェクトはリクエストをサポートすることができません。）

SlpLineCharsを設定することにより、**SlpLineWidth**、**SlpLineHeight**、**SlpLineSpacing**も更新されるかもしれません。なぜなら、文字のピッチやフォントが変わるかもしれないからです。

SlpLineCharsの値は、**Open**メソッド後最初にデバイスをイネーブルにした時に、プリンタのデフォルトの行文字幅に初期化されます。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	不正な行文字幅が指定されました。

参照　**SlpLineCharsList**プロパティ

SlpLineCharsList プロパティ

形式　BSTR SlpLineCharsList;

説明　スリップによりサポートされる行文字幅を含む文字列です。

このプロパティは、**Open**メソッドにより初期化されます。

文字列は、カンマによって分けられるASCII数字の設定から成ります。
たとえば、文字列が”32,36,40”ならば、スリップは、32文字、36文字、40文字の行幅をサポートします。

参照　SlpLineCharsプロパティ

SlpLineHeight プロパティ　R/W

形式　LONG SlpLineHeight;

説明　スリップ印刷の行の高さです。**MapMode**で定義された単位で表記します。

現在の文字の幅と一緒にサポートできる高さに変更されれば、その行の高さがこの値に設定されます。もし、正確な高さがサポートされていなければ、その高さは最も近似のサポートされている値に設定されます。

SlpLineCharsが変わると、**SlpLineHeight**は、選択された幅に対するデフォルトの行高さに更新されます。

SlpLineHeightの値は、**Open**メソッド後最初にデバイスをイネーブルにした時に、プリンタのデフォルトの行高さに初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

参照　SlpLineCharsプロパティ

SlpLinesNearEndToEnd プロパティ

形式　LONG SlpLinesNearEndToEnd;

説明　用紙ニアエンドセンサがTRUEになった後に、POSプリンタがスリップ用紙の終わりに着くまでに印字できる行数です。

このプロパティは、できる限り多くの行数が印刷されるように、スリップ用紙の利用を最適化するために利用されます。

SlpLineHeightあるいは**SlpLineSpacing**、**SlpLineChars**プロパティが変わることによって、このプロパティが変わるかもしれません。

このプロパティは、**Open**メソッド後最初にデバイスをイネーブルにした時に初期化されます。

参照　**SlpEmpty**プロパティ、**SlpNearEnd**プロパティ

SlpLineSpacing プロパティ　R/W

形式　LONG SlpLineSpacing;

説明　通常文字の印刷行の高さを示します。すなわち、印字行の高さと行間スペースの高さの両方を加えたものです。プリンタの種類や現在の行間の値によっては、縦倍角文字はこの値を超えるかも知れません。本プロパティは**MapMode**で定義された単位で表記します。

POSプリンタによってサポートされる行間に変更されれば、その行間はこの値に設定されます。もし、その行間がサポートされなければ、その行間は最も近似的にサポートされる値に設定されます。

SlpLineCharsまたは**SlpLineHeight**が変わると、**SlpLineSpacing**は、選択された幅あるいは高さに対するデフォルトの行間に更新されます。

SlpLineSpacingの値は、**Open**メソッド後最初にデバイスをイネーブルにした時に、POSプリンタのデフォルトの行間に初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

SlpLineWidth プロパティ

形式　　LONG SlpLineWidth;

説明　　SlpLineCharsの一行の幅です。MapModeで定義された単位で表記します。

SlpLineCharsを設定すると、SlpLineWidthも更新されます。

SlpLineWidthの値は、Openメソッド後最初にデバイスをイネーブルにした時に、POSプリンタのデフォルトの行幅に初期化されます。

SlpMaxLines プロパティ

形式　　LONG SlpMaxLines;

説明　　スリップ用紙に印刷できる最大行数です。

CapSlpFullslipプロパティがTRUEの時、この値は0になり、制限のない最大スリップ長であることを指示します。

CapSlpFullslipプロパティがFALSEの時、この値は0以外になります。

SlpLineHeightあるいはSlpLineSpacing、SlpLineCharsプロパティが変わることにより、このプロパティが変わるかもしれません。

SlpMaxLinesの値は、Openメソッド後最初にデバイスをイネーブルにした時に、初期化されます。

SlpNearEnd プロパティ

形式　BOOL SlpNearEnd;

説明　TRUE：スリップ用紙がニアエンドです。
FALSE：ニアエンドではありません。

用紙ニアエンドセンサは、時々”trailing edge”センサとも呼ばれ、スリップ用紙の下端を参照します。

もし、**CapSlpNearEndSensor**がFALSEならば、このプロパティの値は常にFALSEです。

このプロパティは、デバイスをイネーブルにした時に初期化され、イネーブルしている間、カレント値に保持されます。

備考　スリップ用紙無しセンサは、印刷する前にスリップ用紙が挿入されているかどうかを判定するために用いられ、さらに、スリップ用紙が定位置に留まっていることを監視できます。このセンサは通常、スリップのプリンタヘッド位置より一行以上上に設置されます。

逆に、スリップ用紙ニアエンドセンサ（設置されていれば）は、スリップ用紙の終了が近づいたことを判定するために用いられます。このセンサは通常、スリップのプリンタヘッド位置より一行以上下に設置されます。

参照　SlpEmptyプロパティ、SlpLinesNearEndToEndプロパティ

SlpSidewaysMaxChars プロパティ

形式　　LONG SlpSidewaysMaxChars;

説明　　サイドウェイモード（左90度または右90度回転印刷）において、各行に印刷される半角文字の最大文字数です。

CapSlpLeft90とCapSlpRight90の両方がFALSEならば、SlpSidewaysMaxCharsは0です。

SlpLineHeightとSlpLineSpacing、SlpLineCharsプロパティを変えることにより、このプロパティは変わるかもしれません。

このプロパティは、Openメソッド後最初にデバイスをイネーブルにした時に初期化されます。

参照　　SlpSidewaysMaxLinesプロパティ

SlpSidewaysMaxLines プロパティ

形式　LONG SlpSidewaysMaxLines;

説明　サイドウェイモード（左90度または右90度回転印刷）において、印刷される最大行数です。

CapSlpLeft90とCapSlpRight90の両方がFALSEならば、SlpSidewaysMaxLinesは0です。

SlpLineHeightとSlpLineSpacing、SlpLineCharsプロパティを変えることにより、このプロパティは変わるかもしれません。

このプロパティは、Openメソッド後最初にデバイスをイネーブルにした時に初期化されます。

参照　SlpSidewaysMaxCharsプロパティ

SlpPrintSideプロパティ　*第1.5版で追加、第1.7版で更新*

形式　LONG SlpPrintSide;

説明　現在のスリップ印字面を保持します。

スリップ用紙が挿入されていなければ、このプロパティは、PTR_PS_UNKNOWNになります。

プリンタが両面印刷機能を持っている場合、CapSlpBothSidePrintプロパティにはTUREが設定されます。スリップが挿入された時、PTR_PS_SIDE1かPTR_PS_SIDE2のいずれかが格納されます。このプロパティの値は、ChangePrintSideメソッドを実行された場合に、変更されます。

プリンタが両面印刷機能を持っていなければ、CapSlpBothSidePrintプロパティにはFASEが設定され、スリップが挿入された時には、PTR_PS_SIDE1が設定されます。

プリンタが両面印刷機能を持っている場合、BeginInsertion/EndInsertionメソッドが実行された後にSlpPrintSideプロパティの値は、PTR_PS_SIDE1が設定されます。しかし、MICRのBeginInsertion/EndInsertionメソッドが実行された後のSlpPrintSideプロパティの値は、PTR_PS_SIDE1とは限りません。この場合、 SlpPrintSideプロパティは、バリデーション印刷面を示します。

値	意味

PTR_PS_UNKNOWN スリップ用紙が挿入されていません。

PTR_PS_SIDE1　デフォルト印刷面です。（スリップを挿入した後、プリンタはただちにこの面に印刷できます。）

PTR_PS_SIDE2　印刷できるもう片方の面です。（デフォルト面の裏側です。）

このプロパティは、デバイスをイネーブルにしている間に初期化され、現状を保持します。

参照　**CapSlpBothSidesPrint**プロパティ、**ChangePrintSide**メソッド

メソッド

BeginInsertion メソッド

形式　LONG BeginInsertion (LONG *Timeout*);

*Timeout*パラメータは、メソッドを失敗するまでに与えられるミリ秒数です。

０の場合、このメソッドは取り込みモードを行い、即時に適切なステータスを戻します。

OPOS_FOREVER(-1)の場合、このメソッドは取り込みモードを行い、フォームが挿入されるかエラーが発生するまで待ちます。

説明　スリップ用紙の挿入処理を開始する時にこのメソッドを呼び出します。

このメソッドが呼ばれると、スリップは、スリップ用紙を扱う挿入口を開けたり、スリップ用紙挿入モードにすることにより、スリップ用紙を受け取る準備をします。このメソッドは、スリップ用紙の挿入をコントロールする**EndInsertion**と組になっています。

もし、POSプリンタデバイスが挿入モードに設定できなければ、アプリケーションにエラーが返されます。そうでなければ、POSプリンタコントロールは以下のどちらかになるまで、スリップ用紙の挿入を監視し続けます。

スリップ用紙の挿入に成功した場合：

この場合、POSプリンタコントロールは、OPOS_SUCCESSを返します。

*Timeout*ミリ秒が経過するまでにスリップ用紙が挿入されないか、エラーがPOSプリンタデバイスによって報告された場合：

この場合、POSプリンタコントロールはOPOS_E_TIMEOUTか、あるいは他のエラーを返します。しかし、POSプリンタデバイスはスリップ用紙挿入モードを維持します。これによって、アプリケーションはユーザと対話して処理を行うことができ、**BeginInsertion**メソッドを再発行できます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。

OPOS_E_BUSY	出力中なので実行できません。
OPOS_E_ILLEGAL	POSプリンタはスリップを持っていません。（**CapSlpPresent**プロパティ参照）あるいは、無効な*Timeout*パラメータが指定されています。
OPOS_E_TIMEOUT	スリップ用紙が適切に挿入されずに、指定された時間が経過しました。
その他	**ResultCode**の項目を参照してください。

参照　**EndInsertion**メソッド、**BeginRemoval**メソッド、**EndRemoval**メソッド

BeginRemoval メソッド

形式　LONG BeginRemoval (LONG *Timeout*);

*Timeout*パラメータは、メソッドを失敗するまでに与えられるミリ秒数です。

０の場合、このメソッドは排出モードを行い、即時に適切なステータスを戻します。

OPOS_FOREVER(-1)の場合、このメソッドは排出モードを行い、フォームが排出されるかエラーが発生するまで待ちます。

説明　スリップ用紙の排出処理を開始する時にこのメソッドを呼び出します。

このメソッドが呼ばれると、スリップは、スリップ用紙を扱う挿入口を開けたり、スリップ用紙排出モードにすることにより、スリップ用紙を排出する準備をします。このメソッドは、スリップ用紙の挿入をコントロールするEndRemovalと組になっています。

もし、POSプリンタデバイスが排出あるいは除去モードに設定できなければ、アプリケーションにエラーが返されます。そうでなければ、POSプリンタコントロールは以下のどちらかになるまで、スリップ用紙の排出を監視し続けます。

スリップ用紙の排出に成功した場合：

この場合、POSプリンタコントロールは、OPOS_SUCCESSを返します。

*Timeout*ミリ秒が経過するまでにスリップ用紙が排出されないか、エラーがPOSプリンタデバイスによって報告された場合：

この場合、POSプリンタコントロールはOPOS_E_TIMEOUTか、あるいは他のエラーを返します。しかし、POSプリンタデバイスはスリップ用紙排出モードを維持します。これによって、アプリケーションはユーザと対話して処理を行うことができ、BeginRemovalメソッドを再発行できます。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_BUSY	出力中なので実行できません。

OPOS_E_ILLEGAL	POSプリンタはスリップを持っていません。（**CapSlpPresent**プロパティ参照）あるいは、無効な*Timeout*パラメータが指定されています。
OPOS_E_TIMEOUT	スリップ用紙が適切に排出されずに、指定された時間が経過しました。
その他	**ResultCode**の項目を参照してください。
参照	**BeginInsertion**メソッド、**EndInsertion**メソッド、**EndRemoval**メソッド

ChangePrintSideメソッド　　*第1.9版で更新*

形式　　LONG ChangePrintSide (LONG *Side*);

*Side*パラメータは、印刷面を示します。 *Side*値がPTR_PS_SIDE1であれば、用紙のデフォルトの印刷面を選択することを意味します。

（デフォルトの印刷面とは、電源を投入し用紙を挿入したあと直ちに印字することができる面です。**BeginInsertion/EndInsertion**メソッドが実行された後は、 PTR_PS_SIDE1が選択されます。）

*Side*値がPTR_PS_SIDE2であれば、電源投入後のデフォルトの面とは反対の面を選択することを意味します。（ PTR_PS_SIDE1の反対側の面）

*Side*値がPTR_PS_OPPOSITEであれば、現在選択されている面とは反対の面を切り替えて選択することを意味します。（ 例えば、**SlpPrintSide**プロパティがPTR_PS_SIDE1であった場合、PTR_PS_SIDE2に変更されます。）

説明　　印刷面を選択するために使用します。

用紙の両面に印刷する機能を実現します。用紙の向きを変える機種と用紙の両面に印字ヘッドが搭載されている機種があります。

用紙が挿入されていなければ、エラーを返します。

*Side*が**SlpPrintSide**プロパティと違うか、*Side*がPTR_PS_OPPOSITEであれば、用紙面が変更され、用紙の先頭へフィードします。*Side*が**SlpPrintSide**プロパティと同じであれば何も起こらず、 OPOS_SUCCESSが返ります。

メソッド実行中は、**SlpPrintSide**プロパティが変更されるかもしれません。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは成功しました。
OPOS_E_BUSY	出力中のため実行できません。
OPOS_E_ILLEGAL	スリップ印刷ステーションは、プリンタには存在しません。(**CapSlpPresent**プロパティを参照。) スリップ印刷ステーションは、両面印刷に対応していません。（**CapSlpBothSidesPrint**プロパティを参照。)

OPOS_E_EXTENDED　**ResultCodeExtended** =
OPOS_EPTR_COVER_OPEN:
プリンタカバーが開いています。

ResultCodeExtended =
OPOS_EPTR_SLIP_EMPTY:
スリップ用紙がありません。

ResultCodeExtended =
OPOS_EPTR_SLP_CARTRIDGE_EMPTY:
スリップカートリッジが空です。

ResultCodeExtended =
OPOS_EPTR_SLP_CARTRIDGE_REMOVED
:
スリップカートリッジが取り外されています。

ResultCodeExtended =
OPOS_EPTR_SLP_HEAD_CLEANING:
スリップヘッドがクリーニング中です。

その他　**ResultCode**の項目を参照してください。

参照　**SlpPrintSide**プロパティ、**CapSlpBothSidesPrint**プロパティ、**CapSlpPresent**プロパティ

ClearPrintAreaメソッド　　*第1.9版で追加*

形式　LONG ClearPrintArea ();

説明　**PageModePrintArea**プロパティで定義したページモード印刷領域上の印刷データを消去します。

ページモード領域全体を消去したい場合は、**PageModeArea**プロパティで示される領域と同等の領域を**PageModePrintArea**プロパティに指定し、PTR_PM_CANCELを引数として**ClearPrintArea**メソッドを呼び出します。

このメソッドを呼び出す前に、**PageModeStation**プロパティに有効なステーションを指定して下さい。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは成功しました。
その他	**ResultCode**の項目を参照してください。

参照　**PageModeArea**プロパティ、**PageModePrintArea**プロパティ、**PageModeStation**プロパティ; **PageModePrint**メソッド

CutPaper メソッド　*第1.9版で更新*

形式　LONG CutPaper (LONG *Percentage*);

*Percentage*パラメータは、カットすべき用紙のパーセンテージを示します。定数PTR_CP_FULLCUTあるいは値100によって、用紙はフルカットされます。他の値では、部分カットされます。

説明　レシート用紙を切る時にこのメソッドを呼び出します。

このメソッドは、**AsyncMode**がFALSEならば、同期的に実行され、**AsyncMode**がTRUEならば、非同期的に実行されます。

用紙カットのできる多くのPOSプリンタが、フルカットと部分カットの両方を実行できます。いくつかのPOSプリンタは、ミシン目カットやほとんどフルカットのような、段階的な部分カットを提供します。カットの正確なタイプは、POSプリンタの能力によって変わりますが、以下の一般的なガイドが使われます。

値	意味
100	フルカット。
90	とても簡単に切れるように、紙のほんの少しの部分を残して切ります。
70	完全に切ることが多少難しく、不意に触ることによって切れない程度に、紙にミシン目を入れます。
50	紙に部分的にミシン目を入れます。

PrintNormalや**PrintImmediate**メソッド呼び出しの際に、用紙カットのエスケープシーケンスを用いても用紙カットを行うことができます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_BUSY	出力中なので実行できません。（**AsyncMode**がFALSEの時のみ返されます。）

OPOS_E_ILLEGAL	不正なパーセンテージが指定されたり、レシートが存在しなかったり （**CapRecPresent**プロパティ参照）、レシートが用紙カット機構を持たない場合です。（**CapRecPapercut** プロパティ参照）または、レシートステーションがページモードを終了していません。
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN: プリンタカバーが開いています。（**AsyncMode**がFALSEの時のみ返されます。）
	ResultCodeExtended = OPOS_EPTR_REC_EMPTY: レシート用紙がありません。（**AsyncMode**がFALSEの時のみ返されます。）
その他	**ResultCode**の項目を参照してください。

参照 「**印字データとエスケープシーケンス**」

DrawRuledLine メソッド *第1.13版で追加*

形式 LONG DrawRuledLine(LONG *Station*, BSTR *PositionList*, LONG *LineDirection*,
LONG *LineWidth*, LONG *LineStyle*, LONG *LineColor*);

パラメータ	説明
Station	使用するステーション。通常はPTR_S_RECEIPTまたはPTR_S_SLIPです。
PositionList	罫線の位置を指定します。
LineDirection	罫線の方向を指定します。PTR_RL_HORIZONTAL(横罫線)、またはPTR_RL_VERTICAL(縦罫線)です。
LineWidth	罫線の太さを指定します。DOT単位で指定します。

LineStyle	罫線の種類を指定します。罫線の種類は下記のとおりです。 PTR_LS_SINGLE_SOLID_LINE(=1) ―――――― PTR_LS_DOUBLE_SOLID_LINE(=2) ═══════ PTR_LS_BROKEN_LINE(=3) - - - - - - - - - - PTR_LS_CHAIN_LINE(=4) ―・―・―・―・―
LineColor	罫線の色を指定します。エスケープシーケンスのカスタムカラー(ESC|#rC)の"#"で指定する値と同じです。

説明　指定されたステーションに対して、罫線を描画します。

このメソッドは、**AsyncMode**がFALSEならば、同期的に実行され、**AsyncMode**がTRUEならば、非同期的に実行されます。

*PositionList*パラメータの文字列は、*LineDirection*の指定が横罫線と縦罫線の場合で異なります。

横罫線の場合：

横罫線の描画開始位置と長さを示すコンマで区切られたASCII数字で始まり、ASCIIセミコロン(";")の区切りによって複数の横罫線を指定できます。描画開始位置と長さは**MapMode**プロパティの単位で指定可能です。

下記は、*PositionList*パラメータの例です。

- "0,500"
  左から0ドットを描画開始位置とし500ドットの長さの罫線を描画します。
- "0,200;300,100"
  左から0ドットを描画開始位置とし200ドットの長さの横罫線を描画します。また、左から300ドットを描画開始位置とし100ドットの長さの横罫線を描画します。

縦罫線の場合：

縦罫線の描画位置を示すコンマで区切られたASCII数字で指定できます。描画位置は**MapMode**プロパティの単位で指定可能です。

下記は、*PositionList*パラメータの例です。

- “0,100,400,500”
  左から0ドット目、100ドット目、400ドット目, 500ドット目に縦罫線を描画します。

縦罫線は、その後の印字動作を行った場合に描画されます。

縦罫線の描画を終了する場合は、*PositionList*に空文字列(“”)を指定します。

描画位置の基点は回転印字モードによって変わります。

- Normal:左上
- Right90:右上
- Left90:左下
- Rotate180:右下

*LineWidth*パラメータで罫線の太さを指定できます。デバイスでサポートされていない太さが指定された場合、デバイスでサポートしている適切な値が適用されます。

*LineStyle*パラメータで罫線の種類を指定できます。デバイスでサポートされていない種類が指定された場合、デバイスでサポートしている適切な種類が適用されます。

*LineColor*パラメータで罫線の色を指定できます。デバイスでサポートされていない種類が指定された場合、デバイスでサポートしている適切な色が適用されます。

このメソッドは、一括印字モード、もしくは回転印字モードでサービス(デバイス)が印字データをバッファリングしている時に使用できます。それ以外のモードで使用した場合、エラーが返されます。(デバイスによっては90度回転で罫線を印刷できないものがあります。この場合はエラーが返されます。)

ClearOutputメソッド、もしくは印刷モードを変更した場合、縦罫線の指定は終了します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_BUSY	出力中なので実行できません。 （**AsyncMode**がFALSEの時のみ返されます。）
OPOS_E_ILLEGAL	以下のエラーの内の一つが起こりました。 ・指定されたパラメータが不正です。 ・印刷モードが不正です。
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN: プリンタカバーが開いています。 （**AsyncMode**がFALSEの時のみ返されます。）
	ResultCodeExtended = OPOS_EPTR_REC_EMPTY: 指定されたレシートは、用紙が挿入されていません。 （**AsyncMode**がFALSEの時のみ返されます。）
	ResultCodeExtended = OPOS_EPTR_REC_CARTRIDGE_EMPTY: レシートのカートリッジは空です。 （**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_CARTRIDGE_REMOVED:
レシートのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_HEAD_CLEANING:
レシートのヘッドは、クリーニング中です。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_EMPTY:
指定されたスリップには、用紙が挿入されていません。
（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_CARTRIDGE_EMPTY:
スリップのカートリッジは空です。
（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_CARTRIDGE_REMOVED:
スリップのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_HEAD_CLEANING:
スリップのヘッドは、クリーニング中です。（**AsyncMode**がFALSEの時のみ返されます。）

その他　　**ResultCode**の項目を参照してください。

参照　　**MapMode**プロパティ

EndInsertion メソッド

形式　LONG EndInsertion ();

説明　用紙の挿入処理を終了する時にこのメソッドを呼び出します。

このメソッドが呼ばれると、POSプリンタは用紙挿入モードを抜けます。もしスリップデバイスが用紙挿入口を持っていれば、それはこのメソッドによりクローズされます。もし、用紙がデバイス内に見つかれば、OPOS_SUCCESSがアプリケーションに返されます。もし、用紙が無ければ、OPOS_EPTR_SLP_EMPTYが返されます。

このメソッドは用紙の挿入をコントロールするBeginInsertionメソッドと組になっています。もし、用紙がスリップ内にある時間を決めるためにPOSプリンタセンサを使いたいならば、アプリケーションはBeginInsertionが成功した後すぐに、このメソッドを呼んだ方がよいでしょう。代わりに、アプリケーションはこのメソッドを呼ぶ前に、ユーザに注意を促して、キーの押下を待ってもよいでしょう。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_BUSY	出力中のため実行できません。
OPOS_E_ILLEGAL	プリンタはスリップ用紙の挿入モードではありません。
OPOS_E_EXTENDED	ResultCodeExtended = OPOS_EPTR_COVER_OPEN: プリンタカバーが開いている一方で、デバイスが挿入モードを抜けました。 ResultCodeExtended = OPOS_EPTR_SLP_EMPTY: 用紙が挿入されずに、デバイスが挿入モードを抜けました。
その他	ResultCodeの項目を参照してください。

参照　BeginInsertionメソッド、BeginRemovalメソッド、EndRemovalメソッド

EndRemoval メソッド

形式 LONG EndRemoval ();

説明 用紙の排出処理を終了する時にこのメソッドを呼び出します。

このメソッドが呼ばれると、POSプリンタは用紙の排出あるいは除去モードを抜けます。もしスリップデバイス内に用紙が見つからなければ、OPOS_SUCCESSがアプリケーションに返されます。もし、用紙があれば、OPOS_EPTR_SLP_FORMが返されます。

このメソッドは用紙の排出をコントロールするBeginRemovalメソッドと組になっています。もし、用紙が排出し終えた時間を決めるためにPOSプリンタセンサを使いたいならば、アプリケーションはBeginRemovalが成功した後すぐに、このメソッドを呼んだ方がよいでしょう。代わりに、アプリケーションはこのメソッドを呼ぶ前に、ユーザに注意を促して、キーの押下を待ってもよいでしょう。

戻り値 次の値のいずれかが戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_BUSY	出力中のため実行できません。
OPOS_E_ILLEGAL	POSプリンタがスリップの排出モードではありません。
OPOS_E_EXTENDED	ResultCodeExtended = OPOS_EPTR_SLP_FORM: 用紙がまだ存在する一方で、デバイスが排出モードを抜けました。
その他	ResultCodeの項目を参照してください。

参照 BeginInsertionメソッド、EndInsertionメソッド、BeginRemovalメソッド

MarkFeedメソッド　　*第1.5版で追加*

形式　LONG MarkFeed (LONG *Type*);

*Type*パラメータは、マーク付き用紙のハンドリングの種類を示します。

*Type*パラメータの値は次の通りです。

Value	Meaning
PTR_MF_TO_TAKEUP	マーク付き用紙を、取り出せる位置までフィードします。
PTR_MF_TO_CUTTER	マーク付き用紙を、自動カッターのカット位置の上までフィードします。
PTR_MF_TO_CURRENT_TOF	現在のマーク付き用紙の先頭を印字位置までフィードします。（リバースフィード）
PTR_MF_TO_NEXT_TOF	次のマーク付き用紙の先頭を印字位置までへフィードします。

説明　このメソッドは、レシート用紙用のマーク付き用紙を利用するために使用されます。

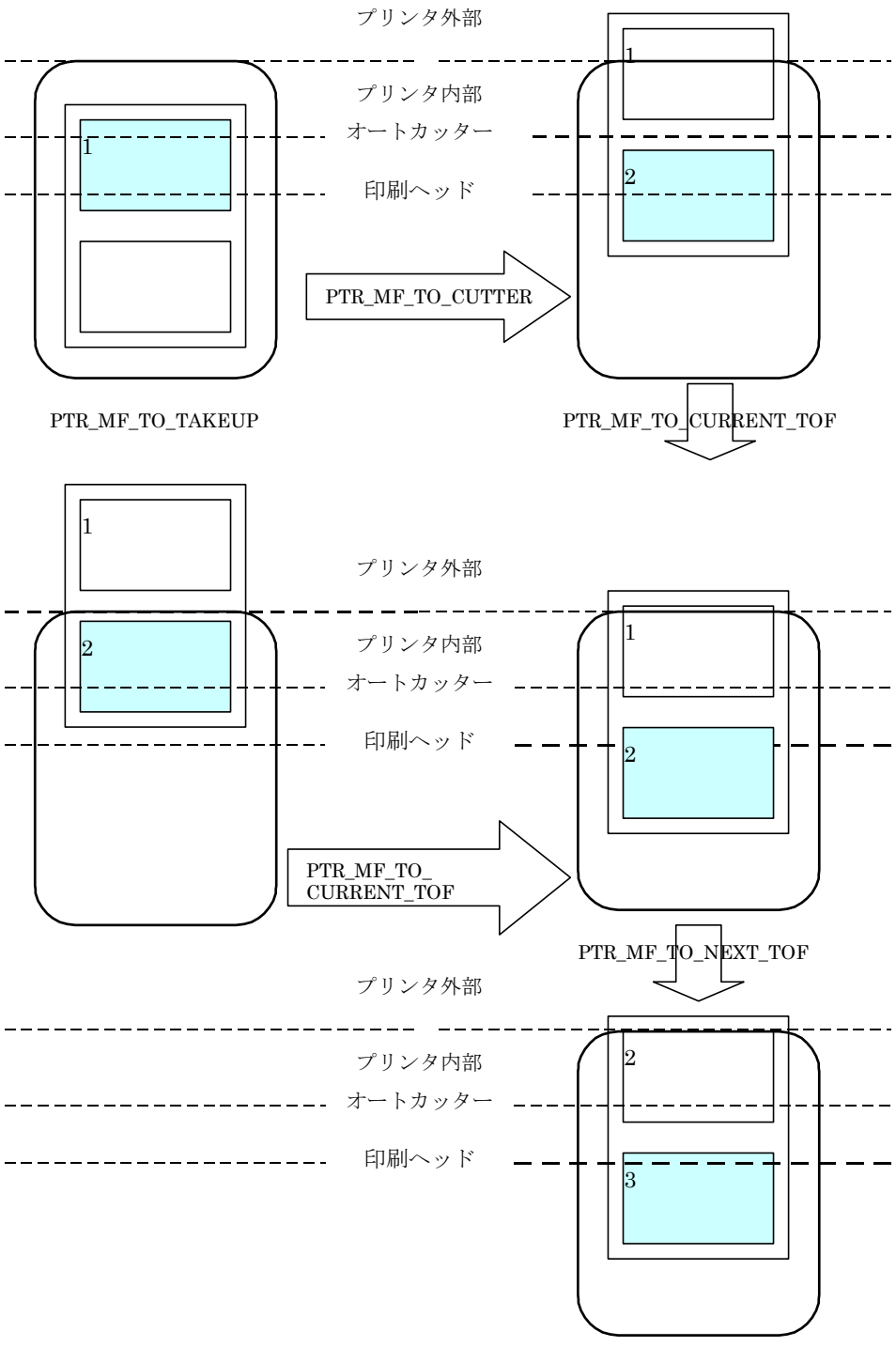
このメソッドは、 **AsyncMode**プロパティがFALSEであれば非同期で、 **AsyncMode**プロパティがTRUEであれば、同期で実行されます。

*Type*がPTR_MF_TO_TAKEUPであれば、プリンタは、オペレータが手動で切り離せるように現在のマーク付き用紙をフィードします。

*Type*がPTR_MF_TO_CUTTERであれば、プリンタは、マーク付き用紙をオートカッターによって切り離すことができる位置までフィードします。続いて**CutPaper**メソッドを使用する事により、マーク付き用紙を切り離す事ができます。

*Type*がPTR_MF_TO_CURRENT_TOFであれば、現在のマーク付き用紙の先頭を印字位置までフィードします。（現在のマーク付き用紙の先頭が印字位置よりも先にある場合は、リバースフィードします。）

*Type*がPTR_MF_TO_NEXT_TOFであれば、次のマーク付き用紙の先頭を印字位置までへフィードします。



戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	レシート印刷ステーションは、マーク付き用紙のハンドリング機能をサポートしていません。（**CapRecMarkFeed**プロパティを参照）
OPOS_E_BUSY	出力中のため実行できません。(**AsyncMode**がFALSEの時のみ返されます。)
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN: プリンタカバーが開いています。（**AsyncMode**がFALSEの時のみ返されます。） **ResultCodeExtended** = OPOS_EPTR_REC_EMPTY: 指定されたレシートは、用紙がありません。（**AsyncMode**がFALSEの時のみ返されます。） **ResultCodeExtended** = OPOS_EPTR_REC_CARTRIDGE_EMPTY: レシートのカートリッジは空です。（**AsyncMode**がFALSEの時のみ返されます。） **ResultCodeExtended** = OPOS_EPTR_REC_CARTRIDGE_REMOVED: レシートのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。） **ResultCodeExtended** = OPOS_EPTR_REC_HEAD_CLEANING: レシートのヘッドは、クリーニング中です。（**AsyncMode**がFALSEの時のみ返されます。）
その他	**ResultCode**の項目を参照してください。

参照　**CapRecMarkFeed**プロパティ

PageModePrintメソッド　　*第1.11版で更新*

形式　　LONG PageModePrint (LONG *Control*);

*Control*パラメータの値は次の通りです。

値	意味
PTR_PM_PAGE_MODE	ページモードを開始します。
PTR_PM_PRINT_SAVE	ページモード印刷領域の印刷データを印刷し、その印刷データを保存します。繰り返し印刷する場合に用います。
PTR_PM_NORMAL	ページモード印刷領域の印刷データを印刷し、その印刷データを消去し、ページモードを終了します。
PTR_PM_CANCEL	ページモード印刷領域の印刷データを消去し、何も印刷せずにページモードを終了します。

説明　　**PageModeStation**プロパティに指定したステーションを対象として、ページモードを開始/終了します。

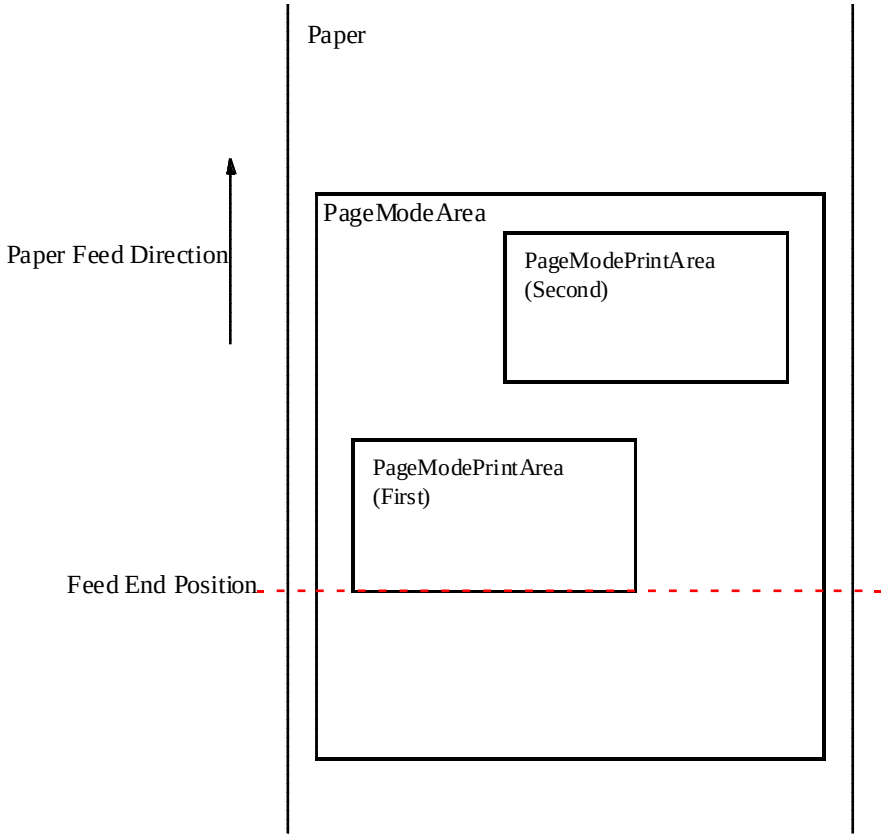
*Control*にPTR_PM_PAGE_MODEを指定した場合、ページモードを開始します。その後、PTR_PM_PRINT_SAVE、PTR_PM_NORMAL、PTR_PM_CANCELを指定して**PageModePrint**メソッドが呼び出されるまでの間、**PrintNormal**、**PrintBarCode**、**PrintBitmap**、**PrintMemoryBitmap**メソッドで印刷データをバッファリングすることができます（プリンタ内にバッファリングするか、サービスオブジェクト内にバッファリングするかは、プリンタの能力に依存します）。（この間に呼び出されたメソッドは、印刷データをバッファリングするだけで、印刷を開始することはありません。また、**AsyncMode**プロパティの設定は、ページモード機能に影響を与えません。それぞれの操作に対して、**OutputID**が割り当てられることも、**OutputCompleteEvent**が通知されることもありません。）

*Control*にPTR_PM_PRINT_SAVEを指定した場合、ページモードが継続します。**PrintNormal**、**PrintBarCode**、**PrintBitmap**、**PrintMemoryBitmap**メソッドでバッファリングした印刷データがあれば、それを保存して、印刷します。ページモード領域内に印刷データを追加して、同じレイアウトで印刷したい場合に用います。

*Control*にPTR_PM_NORMALを指定した場合、ページモードを終了し、通常の状態へ遷移します。**PrintNormal**、**PrintBarCode**、**PrintBitmap**、**PrintMemoryBitmap**メソッドでバッファリングした印刷データがあれば、それを印刷します。バッファリングした印刷データは保存されません。

*Control*にPTR_PM_CANCELを指定した場合、ページモードを終了し通常の状態へ遷移します。**PrintNormal**、**PrintBarCode**、**PrintBitmap**、**PrintMemoryBitmap**メソッドでバッファリングした印刷データがあっても、それらは印刷されず、保存もされません。

PageModePrintメソッドが呼び出された時、**PageModePrintArea**で定義されたページモード印刷領域上のすべての印刷データが印刷され、ページモード印刷領域の下端まで紙送りされます。複数のページモード印刷領域が定義されていた場合は、それぞれのページモード印刷領域上のすべての印刷データが印刷され、最も下方に配置されているページモード印刷領域の下端まで紙送りされます（下図を参照して下さい）。



ページモード領域全体が、1つのトランザクションとして一括処理されます。**AsyncMode**プロパティがTRUEであれば非同期的に、FALSEであれば同期的に処理されます。

ClearOutputメソッドが呼び出されると、ページモードを終了し通常の状態へ遷移します。このとき、バッファリングした印刷データも消去されます。

ページモード機能は、トランザクション印刷の中で利用することはできますが、回転印刷の中で利用することはできません。

このメソッドを呼び出す前に、**PageModeStation**プロパティに有効なステーションを指定して下さい。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	**PageModeStation**プロパティで指定されたステーションが存在しない（**CapRecPresent**、**CapSlpPresent**プロパティを参照）か、**CapXxxPageMode**プロパティがFALSEです。または、**PageModeStation**プロパティで指定されたステーションがページモードに遷移していない状態で、*Control*パラメータにPTR_PM_NORMAL、PTR_PM_PRINT_SAVE、またはPTR_PM_CANCELが指定されました。
OPOS_E_BUSY	出力中のため実行できません。(**AsyncMode**がFALSEで、*Control*パラメータにPTR_PM_NORMAL、PTR_PM_PRINT_SAVE、またはPTR_PM_CANCELが指定された時のみ返されます。)
その他	**ResultCode**の項目を参照してください。

参照　**CapXxxPageMode**プロパティ、**PageModePrintArea**プロパティ、**PageModeStation**プロパティ

PrintBarCode メソッド　*第1.15.1版で更新*

形式　LONG PrintBarCode (LONG *Station*, BSTR *Data*, LONG *Symbology*, LONG *Height*, LONG *Width*, LONG *Alignment*, LONG *TextPosition*);

パラメータ	説明
Station	使用するステーション。通常はPTR_S_RECEIPTまたはPTR_S_SLIPです。
Data	バーコードの文字列。 データのフォーマットは**BinaryConversion**プロパティの値に依存します。詳細は**BinaryConversion**プロパティを参照してください。
Symbology	使用されるバーコードタイプ。 以下の値を参照。
Height	バーコードの高さ。**MapMode**で定義された単位で表記します。
Width	バーコードの幅。**MapMode**で定義された単位で表記します。
Alignment	バーコードの位置。下記の値を参照。
TextPosition	文字列の位置。以下の値を参照。

*Alignment*パラメータの値は、

値	意味
PTR_BC_LEFT	左寄せ。
PTR_BC_CENTER	センタリング。
PTR_BC_RIGHT	右寄せ。
その他	バーコード印刷を開始する左端からの距離。**MapMode**で定義された単位で表記します。

*TextPosition*パラメータの値は、

値	意味

PTR_BC_TEXT_NONE　テキストは印字しない。バーコードのみ
印刷します。

PTR_BC_TEXT_ABOVE　バーコードの上にテキストを印字します。

PTR_BC_TEXT_BELOW　バーコードの下にテキストを印字します。

本リリースでの*Symbology*パラメータの値は:

値	ラベルタイプ
1次元バーコードのシンボル	
PTR_BCS_UPCA	UPC-A
PTR_BCS_UPCA_S	UPC-A 補足バーコード付き
PTR_BCS_UPCE	UPC-E
PTR_BCS_UPCE_S	UPC-E 補足バーコード付き
PTR_BCS_UPCD1	UPC-D1
PTR_BCS_UPCD2	UPC-D2
PTR_BCS_UPCD3	UPC-D3
PTR_BCS_UPCD4	UPC-D4
PTR_BCS_UPCD5	UPC-D5
PTR_BCS_EAN8	EAN 8 (= JAN 8)
PTR_BCS_JAN8	JAN 8 (= EAN 8)
PTR_BCS_EAN8_S	EAN 8 補足バーコード付き
PTR_BCS_EAN13	EAN 13 (= JAN 13)
PTR_BCS_JAN13	JAN 13 (= EAN 13)
PTR_BCS_EAN13_S	EAN 13 補足バーコード付き
PTR_BCS_EAN128	EAN-128 EAN-128はGS1-128に置き換えられました。PTR_BCS_EAN128は廃止予定で、PTR_BCS_GS1128に置き換えます。（同じ値を持っています。）
PTR_BCS_TF	Standard (or discrete) 2 of 5

PTR_BCS_ITF　Interleaved 2 of 5

PTR_BCS_Codabar　Codabar

PTR_BCS_Code39　Code 39

PTR_BCS_Code93　Code 93

PTR_BCS_Code128　Code 128

PTR_BCS_OCRA　OCR “A”

PTR_BCS_OCRB　OCR “B”

第1.8版で追加

PTR_BCS_Code128_Parsed　Code128(Code128の特別な考慮の項目を参照)

PTR_BCS_RSS14　Reduced Space Symbology　廃止予定（第1.12版時）：PTR_BCS_GS1DATABARに置き換えます。（同じ値を持っています。）

PTR_BCS_RSS_EXPANDED　Reduced Space Symbology – Expanded　廃止予定（第1.12版時）：PTR_BCS_GS1DATABAR_Eに置き換えます。（同じ値を持っています。）

第1.12版で追加

PTR_BCS_GS1DATABAR　GS1 DataBar Omnidirectional

PTR_BCS_GS1DATABAR_S　GS1 DataBar Stacked Omnidirectional

PTR_BCS_GS1DATABAR_E　GS1 DataBar Expanded

PTR_BCS_GS1DATABAR_E_S　GS1 DataBar Expanded Stacked

第1.15.1版で追加

PTR_BCS_GS1128　GS1-128
GS1-128はEAN-128を置き換えました。
PTR_BCS_EAN128は廃止予定で、PTR_BCS_GS1128に置き換えます。（同じ値を持っています。）

２次元コードのシンボル

PTR_BCS_PDF417　PDF 417

PTR_BCS_MAXICODE	MAXICODE

第1.13版で追加

PTR_BCS_DATAMATRIX	Data Matrix
PTR_BCS_QRCODE	QR Code
PTR_BCS_UQRCODE	Micro QR Code
PTR_BCS_AZTEC	Aztec
PTR_BCS_UPDF417	Micro PDF 417

第1.15.1版で追加

PTR_BCS_GS1DATAMATRIX	GS1 DataMatrix
PTR_BCS_DOTCODE	DotCode
PTR_BCS_HANXIN	Han Xin Code

特別なケース

PTR_BCS_OTHER	このタイプより大きいか、または等しい ならば、サービスオブジェクトはOPOS　で未定義のシンボルキャラクタを通知しました。

第1.14版で追加

スキャナ（バーコードリーダー）デバイスの章が1.14版で更新されています。これは、すべてのPOSトランザクションにおいて一般的な内容とは限りません。従って、新しいシンボルがPOSプリンターで印刷できることは、通常、一般的な要求とはならないので、新しいシンボルは上記POSプリンターで対応すべきバーコードシンボルのリストには含まれません。しかしながら、新たに追加されたスキャナシンボルの印刷が可能ならば、それは、PTR_BCS_OTHERとして扱われる特殊な場合です。

将来、新しいPOSスキャナーコード対応要望が多くなった場合は、上記リストが更新される可能性があります。

Code 128のための特別な考慮

Code 128のバーコードタイプは、３つのコードセットで構成され、コードセットの変更、ファンクションコード、シフトコードを示すいくつかの特殊文字も含んでいます。各コードセットの文字は、以下の通りです。

コードセット	文字セット
Code A	0x00-0x5f, FNC1, FNC2, FNC3, FNC4, SHIFT, CODE B, CODE C
Code B	0x20-0x7f, FNC1, FNC2, FNC3, FNC4, SHIFT, CODE A, CODE C
Code C	0x00-0x63 for decimal values 00-99, FNC1, CODE A, CODE B

第1.7版以前

アプリケーションによって提供されるデータ形式は、第1.7版以前の版では指定されていませんでした。したがって、デフォルトのコードセットとデータの内容は、ベンダーによって異なっていました。第1.7版以前のサービスにCode 128のデータを送信するアプリケーションは、そのサービスの要求に従う必要があります。

第1.8版以降

現行のアプリケーションの移行のために、以前の版で使用されていたデータ形式をサービスがサポートすることで、symbology のPTR_BCS_Code128を維持します。（新しいサービスは、このsymbologyをPTR_BCS_Code128_Parsedとして扱うべきです。）新しいsymbologyのPTR_BCS_Code128_Parsedは、統一された構文構造でデータ形式が標準化されます。データは、選択されたコードセットに対応する値をサービスがマップするASCII文字により構成されます。Code AとCode Bでは、1対1にマッピングされます。Code Cでは、各一対の数字が0x00から0x63の範囲のひとつのCode Cの文字に変換されます。（Code Cのデータが奇数桁の数字である場合、変換の前にサービスによって先頭にゼロの数字が付加されます。）識別子である左中括弧“{”の制御文字が特殊文字を示すために使用されます。下記のテーブルリストが特殊文字をエンコードするための文字の組合せです。

Special Characters	**ASCII Representation**
SHIFT	{S
CODE A	{A
CODE B	{B
CODE C	{C
FNC1	{1
FNC2	{2
FNC3	{3
FNC4	{4
{	{{

デフォルトのコードセットはベンダーによって異なるので、データの始めに最初のコード設定が必要です。

説明　指定したステーションにバーコードを印刷する時にこのメソッドを呼び出します。

このメソッドは、**AsyncMode**がFALSEならば、同期的に実行され、**AsyncMode**がTRUEならば、非同期的に実行されます。

RotateSpecialプロパティがバーコードを回転することを示していれば、バーコードは回転印刷されます。*Height*と*Width*、*TextPosition*パラメータは回転前にバーコードに適用されます。例えば、PTR_BC_TEXT_BELOWが指定されると、バーコードは左に回転され、それから、テキストがバーコードの右側に印刷されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味

OPOS_SUCCESS　メソッドは正常終了しました。

OPOS_E_ILLEGAL　以下のエラーの内の一つが起こりました。
・*Station*が存在しません。
・*Station*がバーコード印刷をサポートしていません。
・*Height*あるいは*Width*が、0あるいは大きすぎます。
・サポートされていない*Symbology*です。
・*Symbology*でサポートされていない文字があります。
・*Alignment*が、不正な値であるか、大きすぎます。
・*TextPosition*が、不正な値です。
・**RotateSpecial**の回転がサポートされていません。

OPOS_E_BUSY　出力中なので実行できません。（**AsyncMode**がFALSEの時のみ返されます。）

OPOS_E_EXTENDED　**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN:
プリンタカバーが開いています。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_EMPTY:
指定されたレシートは、用紙がありません。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_CARTRIDGE_EMPTY:
レシートのカートリッジは空です。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_CARTRIDGE_REMOVED:
レシートのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_HEAD_CLEANING:
レシートのヘッドは、クリーニング中です。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_EMPTY:
指定されたスリップには、用紙が挿入されていません。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_CARTRIDGE_EMPTY:
スリップのカートリッジは空です。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_CARTRIDGE_REMOVED:
スリップのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_HEAD_CLEANING:
スリップのヘッドは、クリーニング中です。（**AsyncMode**がFALSEの時のみ返されます。）

その他　**ResultCode**の項目を参照してください。

参照　**MapMode**プロパティ、**RotateSpecial**プロパティ

PrintBitmap メソッド　　*第1.7版で更新*

形式　LONG PrintBitmap (LONG *Station*, BSTR *FileName*, LONG *Width*, LONG *Alignment*);

パラメータ	説明
Station	使用するステーション。通常はPTR_S_RECEIPTまたはPTR_S_SLIPです。
FileName	ビットマップファイル名。bmp、gif、jpegといった、様々なファイル形式をサポートしている場合があります。少なくとも、白黒2値で、非圧縮のWindowsビットマップはサポートされています（黒のピクセルは印字され、白のピクセルは印字されません）。
Width	ビットマップの印刷幅。下記の値を参照。
Alignment	ビットマップの印刷位置。下記の値を参照。

*Width*パラメータの値は、

値	意味
PTR_BM_ASIS	POSプリンタのドット当たり1ビットマップピクセルで、ビットマップを印刷します。
その他	ビットマップ幅。**MapMode**で定義された単位で表記します。

*Alignment*パラメータの値は、

値	意味
PTR_BM_LEFT	左寄せ。
PTR_BM_CENTER	センタリング。
PTR_BM_RIGHT	右寄せ。
その他	ビットマップ印刷を開始する左端からの距離。**MapMode**で定義された単位で表記します。

説明　指定したステーションにビットマップを印刷する時にこのメソッドを呼び出します。（例えば**PrintNormal**を用いて）送信済みかつ未印字の文字データがあった場合、文字データを印刷し自動的に改行された後、次の行にビットマップが印刷されます。**PrintBitmap**より後に送信された文字データは、ビットマップの次の行から印字されます。

このメソッドは、**AsyncMode**がFALSEならば、同期的に実行され、**AsyncMode**がTRUEならば、非同期的に実行されます。

*Width*パラメータはビットマップの変換をコントロールします。もし、*Width*がPTR_BM_ASISならば、変換は実行されません。ビットマップはPOSプリンタドット当たり1ビットマップピクセルで印刷されます。このオプションの長所は、以下のことです。

- 高いパフォーマンスのビットマップ印刷ができます。
- ビットマップは、水平方向と垂直方向のドット単位がプリンタ固有のアスペクト比に調整され、良好な印刷が得られます。

もし*Width*が0でなければ、ビットマップは、その幅が指定された幅でそのアスペクト比が変化しないように、拡大されたり圧縮されたりすることにより、変換されます。このオプションの長所は以下のことです。

- いろいろなプリンタに合うように、ビットマップの大きさを作ります。
- ビットマップのアスペクト比の比率を維持します。

欠点は、

- 変換しないデータよりもパフォーマンスが落ちます。
- オリジナルのビットマップでは良好ないくつかの行とイメージが、いくらかギザギザとなるかもしれません。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_BUSY	出力中なので実行できません。（**AsyncMode**がFALSEの時のみ返されます。）
OPOS_E_ILLEGAL	以下のエラーの内の一つが起こりました。 ・*Station*が存在しません。 ・*Station*がビットマップ印刷をサポートしていません。 ・*Width*が大きすぎます。 ・*Alignment*が、不正な値であるか、大きすぎます。

OPOS_E_NOEXIST　*FileName*が見つかりませんでした。

OPOS_E_EXTENDED

ResultCodeExtended =
OPOS_EPTR_TOOBIG:
ビットマップが、変換無しで印刷するにはその幅が広すぎるか、あるいは変換するには大きすぎます。

ResultCodeExtended =
OPOS_EPTR_COVER_OPEN:
プリンタカバーが開いています。
（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended =
OPOS_EPTR_BADFORMAT:
指定されたファイルが、ビットマップファイルでないか、あるいはサポートされていないフォーマットです。

ResultCodeExtended =
OPOS_EPTR_REC_EMPTY:
指定されたレシートは、用紙が挿入されていません。
（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_CARTRIDGE_EMPTY:
レシートのカートリッジは空です。
（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended =
OPOS_EPTR_REC_CARTRIDGE_REMOVED:
レシートのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended =
OPOS_EPTR_REC_HEAD_CLEANING:
レシートのヘッドは、クリーニング中です。
（**AsyncMode**がFALSEの時のみ返されます。）

		ResultCodeExtended = OPOS_EPTR_SLP_EMPTY: 指定されたスリップには、用紙が挿入されていません。 （**AsyncMode**がFALSEの時のみ返されます。）
		ResultCodeExtended = OPOS_EPTR_SLP_CARTRIDGE_EMPTY: スリップのカートリッジは空です。 （**AsyncMode**がFALSEの時のみ返されます。）
		ResultCodeExtended = OPOS_EPTR_SLP_CARTRIDGE_REMOVED: スリップのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。）
		ResultCodeExtended = OPOS_EPTR_SLP_HEAD_CLEANING: スリップのヘッドは、クリーニング中です。 （**AsyncMode**がFALSEの時のみ返されます。）
	その他	**ResultCode**の項目を参照してください。

参照　　**MapMode**プロパティ

PrintImmediate メソッド　*第1.9版で更新*

形式　LONG PrintImmediate (LONG *Station*, BSTR *Data*);

パラメータ	説明
Station	使用するステーション。通常はPTR_S_JOURNAL、PTR_S_RECEIPTまたはPTR_S_SLIPです。
Data	印刷される文字。プリント可能な文字とエスケープシーケンス、キャリッジリターン(13 decimal)、ニューライン/ラインフィード(10 decimal)から成ります。 詳細は**BinaryConversion**プロパティを参照してください。

説明　指定したステーションに*Data*を直ちに印刷する時にこのメソッドを呼び出します。

このメソッドは、直ちにそのデータを印刷しようとします。つまり、次のプリンタオペレーションになるのです。このメソッドは、非同期出力が終わっていない時に呼ばれます。**PrintImmediate**は、エラーイベントハンドラ内のような、非同期出力が未解決の時の例外状況で利用することを意図しています。

Data 内の特別なキャラクタの値は、

値	意味
Newline/Line Feed (10)	バッファ内のデータを印字後、次の行へ進みます。(キャリッジリターンは、その行を印刷するためには必要ありません。)

Carriage Return(13)	もし、キャリッジリターンがラインフィードのすぐ前にあるか、ラインバッファが空ならば、キャリッジリターンは無視されます。 そうでなければ、ラインバッファを印字し、プリンタは次の行には進みません。 プリンタが、キャリッジリターンを直接サポートしているものもあります。 また、プリンタが、常に次の行にフィードするものもあります。その場合には、サービスオブジェクトはラインバッファを印字し、逆フィードがサポートされているならば、逆フィードします。 もし、プリンタがこれらのどちらもサポートしないならば、キャリッジリターンはラインフィードのように振る舞います。 **ValidateData**メソッドは、ラインフィード無しのキャリッジリターンが可能か否かと、そのために逆フィードがサポートされる必要があるか否かを決定するために利用されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	指定したプリンタが存在しません。（**CapJrnPresent**、**CapRecPresent**、**CapSlpPresent**プロパティ参照）または、ページモードを終了するまでは、ただちに印刷することのできないデバイスです。
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN: プリンタカバーが開いています。 **ResultCodeExtended** = OPOS_EPTR_JRN_EMPTY: 指定されたジャーナルは、用紙がありません。 **ResultCodeExtended** = OPOS_EPTR_JRN_CARTRIDGE_EMPTY: ジャーナルのカートリッジは空です。

ResultCodeExtended = OPOS_EPTR_ JRN _CARTRIDGE_REMOVED:
ジャーナルのカートリッジは取り外されています。

ResultCodeExtended = OPOS_EPTR_JRN_HEAD_CLEANING:
ジャーナルのヘッドは、クリーニング中です。

ResultCodeExtended = OPOS_EPTR_REC_EMPTY:
指定されたレシートは、用紙がありません。

ResultCodeExtended = OPOS_EPTR_REC_ CARTRIDGE_EMPTY:
レシートのカートリッジは空です。

ResultCodeExtended = OPOS_EPTR_REC_CARTRIDGE_REMOVED:
レシートのカートリッジは取り外されています。

ResultCodeExtended = OPOS_EPTR_REC_HEAD_CLEANING:
レシートのヘッドは、クリーニング中です。

ResultCodeExtended = OPOS_EPTR_SLP_EMPTY:
指定されたスリップには、用紙が挿入されていません。

ResultCodeExtended = OPOS_EPTR_SLP_ CARTRIDGE_EMPTY:
スリップのカートリッジは空です。

ResultCodeExtended = OPOS_EPTR_SLP_CARTRIDGE_REMOVED:
スリップのカートリッジは取り外されています。

ResultCodeExtended = OPOS_EPTR_SLP_HEAD_CLEANING:
スリップのヘッドは、クリーニング中です。

その他　　ResultCodeの項目を参照してください。

参照　　PrintNormalメソッド、PrintTwoNormalメソッド

PrintMemoryBitmap メソッド　*第1.12版で更新*

形式　LONG PrintMemoryBitmap (LONG *Station*, BSTR *Data*, LONG *Type*, LONG *Width*, LONG *Alignment*);

パラメータ	説明
Station	使用するステーション。通常はPTR_S_RECEIPTまたはPTR_S_SLIPです。
Data	ビットマップデータを保持しているバイト配列。このデータフォーマットはBinaryConversion propertyの値によって決まります。BinaryConversion propertyを参照してください。
Type	ビットマップの形式。bmp、gif、jpegといった、様々な形式をサポートしている場合があります。少なくとも、白黒2値で、非圧縮のWindowsビットマップはサポートされています（黒のピクセルは印字され、白のピクセルは印字されません）。下記の値を参照。
Width	ビットマップの印刷幅。下記の値を参照。
Alignment	ビットマップの印刷位置。下記の値を参照。

*Type*パラメータの値は、

値	意味
PTR_BMT_BMP	*Data*パラメータで示されたデータのフォーマットは、BMPです。
PTR_BMT_JPEG	*Data*パラメータで示されたデータのフォーマットは、JPEGです。
PTR_BMT_GIF	*Data*パラメータで示されたデータのフォーマットは、GIFです。

*Width*パラメータの値は、

値	意味
PTR_BM_ASIS	POSプリンタのドット当たり1ビットマップピクセルで、ビットマップを印刷します。
その他	ビットマップ幅。**MapMode**で定義された単位で表記します。

*Alignment*パラメータの値は、

値	意味
PTR_BM_LEFT	左寄せ。
PTR_BM_CENTER	センタリング。
PTR_BM_RIGHT	右寄せ。
その他	ビットマップ印刷を開始する左端からの距離。**MapMode**で定義された単位で表記します。

説明

指定されたステーションに対して、メモリ上に保持されたビットマップを印刷する時にこのメソッドを呼び出します。（例えば**PrintNormal**を用いて）送信済みかつ未印字の文字データがあった場合、文字データを印刷し自動的に改行された後、次の行にビットマップが印刷されます。**PrintMemoryBitmap**より後に送信された文字データは、ビットマップの次の行から印字されます。

このメソッドは、**AsyncMode**がFALSEならば、同期的に実行され、**AsyncMode**がTRUEならば、非同期的に実行されます。

*Width*パラメータはビットマップの変換をコントロールします。もし、*Width*がPTR_BM_ASISならば、変換は実行されません。ビットマップはPOSプリンタドット当たり1ビットマップピクセルで印刷されます。このオプションの長所は、以下のことです。

- 高いパフォーマンスのビットマップ印刷ができます。
- ビットマップは、水平方向と垂直方向のドット単位がプリンタ固有のアスペクト比に調整され、良好な印刷が得られます。

もし*Width*が0でなければ、ビットマップは、その幅が指定された幅でそのアスペクト比が変化しないように、拡大されたり圧縮されたりすることにより、変換されます。このオプションの長所は以下のことです。

- いろいろなプリンタに合うように、ビットマップの大きさを作ります。
- ビットマップのアスペクト比の比率を維持します。

欠点は、

- 変換しないデータよりもパフォーマンスが落ちます。
- オリジナルのビットマップでは良好ないくつかの行とイメージが、いくらかギザギザとなるかもしれません。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_BUSY	出力中なので実行できません。（**AsyncMode**がFALSEの時のみ返されます。）
OPOS_E_ILLEGAL	以下のエラーの内の一つが起こりました。 ・*Station*が存在しません。 ・*Station*がビットマップ印刷をサポートしていません。 ・*Width*が大きすぎます。 ・*Alignment*が、不正な値であるか、大きすぎます。
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_EPTR_TOOBIG: ビットマップが、変換無しで印刷するにはその幅が広すぎるか、あるいは変換するには大きすぎます。 **ResultCodeExtended** = OPOS_EPTR_COVER_OPEN: プリンタカバーが開いています。（**AsyncMode**がFALSEの時のみ返されます。） **ResultCodeExtended** = OPOS_EPTR_BADFORMAT: 指定されたデータが、ビットマップでないか、あるいはサポートされていないフォーマットです。 **ResultCodeExtended** = OPOS_EPTR_REC_EMPTY: 指定されたレシートは、用紙が挿入されていません。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_CARTRIDGE_EMPTY:
レシートのカートリッジは空です。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_CARTRIDGE_REMOVED:
レシートのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_HEAD_CLEANING:
レシートのヘッドは、クリーニング中です。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_EMPTY:
指定されたスリップには、用紙が挿入されていません。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_CARTRIDGE_EMPTY:
スリップのカートリッジは空です。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_CARTRIDGE_REMOVED:
スリップのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_HEAD_CLEANING:
スリップのヘッドは、クリーニング中です。（**AsyncMode**がFALSEの時のみ返されます。）

その他　　**ResultCode**の項目を参照してください。

参照　　**MapMode**プロパティ

PrintNormal メソッド

形式 LONG PrintNormal (LONG *Station*, BSTR *Data*);

パラメータ	説明
Station	使用するステーション。通常は PTR_S_JOURNAL、PTR_S_RECEIPTまたは PTR_S_SLIPです。
Data	印刷される文字。プリント可能な文字とエスケープシーケンス、キャリッジリターン(13 decimal)、ニューライン/ラインフィード(10 decimal)から成ります。 詳細はBinaryConversionプロパティを参照してください。

説明 *Station*に*Data*を印刷する時にこのメソッドを呼び出します。

このメソッドは、AsyncModeがFALSEならば、同期的に実行され、AsyncModeがTRUEならば、非同期的に実行されます。

データ内の特別なキャラクタの値は、

値	意味
Newline/Line Feed (10)	バッファ内のデータを印字後、次の行へ進みます。(キャリッジリターンは、その行を印刷するためには必要ありません。)

Carriage Return(13)　もし、キャリッジリターンがラインフィードのすぐ前にあるか、ラインバッファが空ならば、キャリッジリターンは無視されます。
そうでなければ、ラインバッファを印字し、プリンタは次の行には進みません。
プリンタが、キャリッジリターンを直接サポートしているものもあります。
また、プリンタが、常に次の行にフィードするものもあります。その場合には、サービスオブジェクトはラインバッファを印字し、逆フィードがサポートされているならば、逆フィードします。
もし、プリンタがこれらのどちらもサポートしないならば、キャリッジリターンはラインフィードのように振る舞います。
ValidateDataメソッドは、ラインフィード無しのキャリッジリターンが可能か否かと、そのために逆フィードがサポートされる必要があるか否かを決定するために利用されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	指定したプリンタが存在しません。（**CapJrnPresent**、**CapRecPresent**、**CapSlpPresent**プロパティ参照）
OPOS_E_BUSY	出力中なので実行できません。（**AsyncMode**がFALSEの時のみ返されます。）
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN: プリンタカバーが開いています。（**AsyncMode**がFALSEの時のみ返されます。） **ResultCodeExtended** = OPOS_EPTR_JRN_EMPTY: 指定されたジャーナルは、用紙がありません。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_JRN_CARTRIDGE_EMPTY:
ジャーナルのカートリッジは空です。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_ JRN _CARTRIDGE_REMOVED:
ジャーナルのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_ JRN _HEAD_CLEANING:
ジャーナルのヘッドは、クリーニング中です。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_EMPTY:
指定されたレシートは、用紙がありません。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_CARTRIDGE_EMPTY:
レシートのカートリッジは空です。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_CARTRIDGE_REMOVED:
レシートのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_ REC _HEAD_CLEANING:
レシートのヘッドは、クリーニング中です。（**AsyncMode**がFALSEの時のみ返されます。）

		ResultCodeExtended = OPOS_EPTR_SLP_EMPTY: 指定されたスリップには、用紙が挿入されていません。 （**AsyncMode**がFALSEの時のみ返されます。）
		ResultCodeExtended = OPOS_EPTR_SLP_CARTRIDGE_EMPTY: スリップのカートリッジは空です。 （**AsyncMode**がFALSEの時のみ返されます。）
		ResultCodeExtended = OPOS_EPTR_SLP_CARTRIDGE_REMOVED: スリップのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。）
		ResultCodeExtended = OPOS_EPTR_SLP_HEAD_CLEANING: スリップのヘッドは、クリーニング中です。 （**AsyncMode**がFALSEの時のみ返されます。）
	その他	**ResultCode**の項目を参照してください。

参照　　**PrintImmediate**メソッド、**PrintTwoNormal**メソッド

PrintTwoNormal メソッド　　*第1.9版で更新*

形式　LONG PrintTwoNormal (LONG *Stations*, BSTR *Data1*, BSTR *Data2*);

パラメータ	説明
Stations	使用するステーション。 ***第1.3版以降*** : 以下の1つを選択します。

Stations パラメータ	第1ステーシ	第2ステーシ
PTR_TWO_RECEIPT_JOURNAL	レシート	ジャーナル
PTR_TWO_SLIP_JOURNAL	スリップ	ジャーナル
PTR_TWO_SLIP_RECEIPT	スリップ	レシート

第1.0版　– 第1.2版 :
以下の1つを選択します。
PTR_S_JOURNAL_RECEIPT、
PTR_S_JOURNAL_SLIPまたは
PTR_S_RECEIPT_SLIPです。

Data1	第1のステーションに印刷される文字。印刷可能な文字と、“印字データとエスケープシーケンス”の“印字時に、動作するエスケープシーケンス”テーブルに記載されているエスケープシーケンスから成ります。プリンタは同時に両方のステーションに印刷しようとするので、文字列は１行以内に収まらなければなりません。 詳細はBinaryConversionプロパティを参照してください。
Data2	第2のステーションに印刷される文字。（制約は*Data1*と同様です。） もし、この文字列が空ならば、*Data1*と同じ文字列が印字されます。いくつかのプリンタでは、このフォーマットを使うことで、印刷能力を上げるかもしれません。

「第２章 共通プロパティ、メソッド、イベント」の**BinaryConversion**プロパティを参照。

説明

2つの文字列を2つのステーションに同時に印字する時にこのメソッドを呼び出します。このメソッドをサポートすると、印刷能力が上がるかもしれません。

このメソッドは、**AsyncMode**がFALSEならば、同期的に実行され、**AsyncMode**がTRUEならば、非同期に実行されます。

第1.0版 – 第1.2版

第1.3版以前にリリースされたドキュメントでは、「第1」と「第2」のステーションの意味について十分に明確ではなかったので、インプリメンテーションが次の2つのように異なっていました。

・定数の中のオーダーに基づいてステーションを割り当てます。
　例えば、PTR_S_JOURNAL_RECEIPT は、ジャーナルに*Data1*をレ
　シートに*Data2*を印刷します。

・物理デバイスの特徴あるいは内部の印刷オーダーに基づいてステーションを割り当てます。

この非整合性のため、もしコントロールオブジェクトとサービスオブジェクトのバージョンが第1.3版あるいはそれ以降を示すならば、アプリケーションは新しい定数を使うべきです。

第1.3版以降

第1.3版あるいはそれ以降のサービスオブジェクトは、両方の定数のセットをサポートするべきです。ベンダーは古い定数の動作を定義して、そして文書化するべきです。

定数でのステーションの順は、ステーション上の物理的な印刷の順番を暗示しません。物理的な順はプリンタに依存し、そして双方向印刷や、多数の印刷ヘッドなどに基づいて異なっているかもしれません。

戻り値

次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。

OPOS_E_ILLEGAL　指定されたS*tations*が同時印刷をサポートしていません。（**CapConcurrentJrnRec**、**CapConcurrentJrnSlp**、**CapConcurrentRecSlp**プロパティ参照）または、*Stations*パラメータで指定したステーションのいずれかが、ページモードを終了していません。

OPOS_E_BUSY　出力中なので実行できません。（**AsyncMode**がFALSEの時のみ返されます。）

OPOS_E_EXTENDED　**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN:
プリンタカバーが開いています。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_JRN_EMPTY:
指定されたジャーナルは、用紙がありません。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_JRN_CARTRIDGE_EMPTY:
ジャーナルのカートリッジは空です。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_ JRN_CARTRIDGE_REMOVED:
ジャーナルのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_ JRN_HEAD_CLEANING:
ジャーナルのヘッドは、クリーニング中です。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_EMPTY:
指定されたレシートは、用紙がありません。（**AsyncMode**がFALSEの時のみ返されます。）

	ResultCodeExtended = OPOS_EPTR_REC_CARTRIDGE_EMPTY: レシートのカートリッジは空です。（**AsyncMode**がFALSEの時のみ返されます。） **ResultCodeExtended** = OPOS_EPTR_REC_CARTRIDGE_REMOVED: レシートのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。） **ResultCodeExtended** = OPOS_EPTR_ REC_HEAD_CLEANING: レシートのヘッドは、クリーニング中です。（**AsyncMode**がFALSEの時のみ返されます。） **ResultCodeExtended** = OPOS_EPTR_SLP_EMPTY: 指定されたスリップには、用紙が挿入されていません。（**AsyncMode**がFALSEの時のみ返されます。） **ResultCodeExtended** = OPOS_EPTR_SLP_CARTRIDGE_EMPTY: スリップのカートリッジは空です。（**AsyncMode**がFALSEの時のみ返されます。） **ResultCodeExtended** = OPOS_EPTR_SLP_CARTRIDGE_REMOVED: スリップのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。） **ResultCodeExtended** = OPOS_EPTR_SLP_HEAD_CLEANING: スリップのヘッドは、クリーニング中です。（**AsyncMode**がFALSEの時のみ返されます。）
その他	**ResultCode**の項目を参照してください。

参照　　PrintNormalメソッド

RotatePrint メソッド　　*第1.13版で更新*

形式　　LONG　RotatePrint (LONG *Station*, LONG　*Rotation*);

パラメータ	説明
Station	使用するステーション。通常は PTR_S_RECEIPTまたはPTR_S_SLIPです。
Rotation	回転方向。下記の値を参照。

値	意味
PTR_RP_RIGHT90	右に90度回転（時計回りに）印刷の開始。
PTR_RP_LEFT90	左に90度回転（時計と反対回りに）印刷の開始。
PTR_RP_ROTATE180	180度回転印刷、つまり、倒立印字の開始。
PTR_RP_BARCODE	回転バーコード印刷の開始。この値は上記の回転印刷開始の値の１つとの論理和。
PTR_RP_BITMAP	回転ビットマップ印刷の開始。この値は上記の回転印刷開始の値の１つとの論理和。
PTR_RP_NORMAL	回転印刷の終了。

説明　　回転印刷モードの開始と終了。

このメソッドは、**AsyncMode**がFALSEならば、同期的に実行され、**AsyncMode**がTRUEならば、非同期的に実行されます。

*Rotation*にPTR_RP_ROTATE180が含まれている場合には、倒立印刷モードが開始されます。
RotatePrintが*Rotation*パラメータをPTR_RP_NORMALに設定して呼ばれるまで、**PrintNormal**と**PrintImmediate**の呼び出しによるデータは倒立して印刷されます。行はPOSプリンタコントロールに送られた順に印刷され、各行の先頭はプリンタの右マージンに合わされます。
*Rotation*にPTR_RP_BARCODEやPTR_RP_BITMAPが含まれていない場合は、**PrintNormal**と**PrintImmediate**の印刷メソッドのみが、倒立印刷モード中で使用されます。

*Rotation*にPTR_RP_RIGHT90またはPTR_RP_LEFT90が含まれている場合には、横書きモードが開始されます。**RotatePrint**が*Rotation*パラメータをPTR_RP_NORMALに設定して呼ばれるまで、**PrintNormal**の呼び出しによるデータはバッファリングされます（POSプリンタ能力によって、プリンタまたはサービスオブジェクトにバッファリングされます）。（この場合、上記のメソッドのデータはバッファリングされるだけで、印刷は開始しません。また、**AsyncMode**プロパティの値は、そのオペレーションに影響しません。つまり、その要求に**OutputID**は割り当てられませんし、**OutputCompleteEvent**も通知しません。）

各印刷行は、90度に回転されます。もし、行が全て同じ長さでなければ、各行の先頭位置が合わされます。*Rotation*にPTR_RP_BARCODEやPTR_RP_BITMAPが含まれていない場合は、**PrintNormal**のみが、横書きモード中で使用されます。

*Rotation*にPTR_RP_NORMALが含まれている場合には、回転印刷モードは終了します。もし、横向き回転印刷モード、が有効な時に、あるデータが**PrintNormal**によってバッファリングされていたならば、そのバッファリングされたデータは印刷されます。全体の回転した行のブロックは、一つのメッセージとして扱われます。

*Rotation*にPTR_RP_BARCODE、あるいはPTR_RP_BITMAPが含まれている場合には、（**PrintBarCode**、または'ESC|#R'エスケープシーケンスで印刷される）バーコード、あるいは（**PrintBitmap**、または'ESC|#B'エスケープシーケンスで印刷される）ビットマップはすべて、**RotatePrint**で回転印刷できます。回転方向は、それぞれ**RecBarCodeRotationList**、**SlpBarCodeRotationList**、**RecBitmapRotationList**、**SlpBitmapRotationList**プロパティで制限されます。

*Rotation*にPTR_RP_BARCODEが含まれている場合には、**RotateSpecial**の内容は無視されます。

回転モードを変えることは、プリンタの行高さや行間、行幅、他の長さも変えるかもしれません。

ClearOutputメソッドを呼ぶことにより、回転印刷モードは中止されます。いかなるバッファリングされた横向き回転印刷行も削除されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。

OPOS_E_ILLEGAL　指定したプリンタが存在しません（**CapJrnPresent**、**CapRecPresent**、**CapSlpPresent**プロパティ参照）、または、*Station*が指定した回転をサポートしていません。（プリンタの回転能力プロパティ参照）

OPOS_E_BUSY　出力中なので実行できません。（**AsyncMode**がFALSEの時のみ返されます。）

OPOS_E_EXTENDED　**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN:
プリンタカバーが開いています。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_EMPTY:
指定されたレシートは、用紙がありません。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_CARTRIDGE_EMPTY:
レシートのカートリッジは空です。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_CARTRIDGE_REMOVED:
レシートのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_ REC_HEAD_CLEANING:
レシートのヘッドは、クリーニング中です。（**AsyncMode**がFALSEの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_EMPTY:
指定されたスリップには、用紙が挿入されていません。（**AsyncMode**がFALSEの時のみ返されます。）

	ResultCodeExtended = OPOS_EPTR_SLP_CARTRIDGE_EMPTY: スリップのカートリッジは空です。（**AsyncMode**がFALSEの時のみ返されます。）
	ResultCodeExtended = OPOS_EPTR_SLP_CARTRIDGE_REMOVED: スリップのカートリッジは取り外されています。（**AsyncMode**がFALSEの時のみ返されます。）
	ResultCodeExtended = OPOS_EPTR_SLP_HEAD_CLEANING: スリップのヘッドは、クリーニング中です。（**AsyncMode**がFALSEの時のみ返されます。）
その他	**ResultCode**の項目を参照してください。

参照　「**印字データとエスケープシーケンス**」、RotateSpecialプロパティ

SetBitmap メソッド　　*第1.7版で更新*

形式　LONG SetBitmap (LONG *BitmapNumber*, LONG *Station*, BSTR *FileName*,
LONG *Width*, LONG *Alignment*);

パラメータ	説明
BitmapNumber	このビットマップに割り当てられる番号。 **第1.6版以前**：有効な値は、1と2です。 **第1.7版以降**：有効な値は、1から20です。
Station	使用するステーション。通常はPTR_S_RECEIPTまたはPTR_S_SLIPです。
FileName	ビットマップファイル名。bmp、gif、jpegといった、様々なファイル形式をサポートしている場合があります。少なくとも、白黒2値で、非圧縮のWindowsビットマップはサポートされています（黒のピクセルは印字され、白のピクセルは印字されません）。 もし、空文字列が設定されれば、ビットマップは設定解除されます。
Width	ビットマップの印刷幅。値は**PrintBitmap**参照。
Alignment	ビットマップの印刷位置。値は**PrintBitmap**参照。

説明　まもなく印刷するビットマップに関する情報を保存する時に呼び出します。

ビットマップは、印刷データ内にビットマップ印刷のエスケープシーケンスをもつ**PrintNormal**あるいは**PrintImmediate**を呼ぶことによって印刷されます。ビットマップ印刷のエスケープシーケンスは、初めと終わりの処理ヘッダーを印刷するための文字列を通常含んでいます。

ビットマップ印刷のエスケープシーケンスより前に送信された未印字の文字データがあった場合、文字データを印字し自動的に改行した後、ビットマップが印刷されます。ビットマップ印刷のエスケープシーケンスより後に送信された文字データは、ビットマップの次の行から印刷されます。

サービスオブジェクトはよりよいパフォーマンスを与えるために、まもなく利用するビットマップをキャッシュするように選びます。ビットマップファイルとパラメータはこのメソッドにより、正確さを増します。

高頻度で使用するビットマップには小さな*BitmapNumber*（1に近い値）を割り当て、時々使用されるビットマップには大きな*BitmapNumber*を割り当ててください。サービスオブジェクトは、ビットマップを保存する場所を決めるためにこの数値を用います。可能な場合にはデバイスにダウンロードしますが、サービスオブジェクト内のメモリに保存することもあります。単純に、*FileName*と印刷に必要な関連プロパティを記憶するだけの場合もあります。

アプリケーションは、このメソッドが呼ばれる前に、文字幅、行高さ、行間のようなプリンタの長さが、*Station*に対して設定されることを保証しなければなりません。サービスオブジェクトは、現在の値に基づいて、後で印刷するための準備として、ビットマップを変換するかもしれません。

第1.0版 – 第1.11版
2つだけのビットマップがセットされるかもしれません。そしてそれぞれのビットマップ番号が、ただ一度に１つのステーションのために使われるだけであるかもしれません。

第1.2版以降：
アプリケーションは有効な２つのステーションに対して１あるいは２の番号でビットマップを設定します。もし要求があれば、同じファイル名のビットマップはそれぞれのステーションに対して同一番号で設定されます。従って、同一のビットマップ印字エスケープシーケンスをそれぞれのステーションに対して使用することもできます。
（第1.11版の仕様において、ビットマップは２つだけが設定可能で、あるビットマップ番号は１つのステーションに対してのみ使用できるとされています。）

第1.7版以降：
アプリケーションは有効な２つのステーションに対して１から２０の番号でビットマップを設定します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。

OPOS_E_ILLEGAL	以下のエラーの内の一つが起こりました。 ・*BitmapNumber*が不正な値です。 ・プリンタが存在しません。 ・*Station*がビットマップ印刷をサポートしていません。 ・*Width*が大きすぎます。 ・*Alignment*が、不正な値であるか、大きすぎます。
OPOS_E_NOEXIST	*FileName*が見つかりませんでした。
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_EPTR_TOOBIG: ビットマップが、変換無しで印刷するにはその幅が広すぎるか、あるいは変換するには大きすぎます。 **ResultCodeExtended** = OPOS_EPTR_BADFORMAT: 指定されたファイルが、ビットマップファイルでないか、あるいはサポートされていないフォーマットです。
その他	**ResultCode**の項目を参照してください。

参照　「**印字データとエスケープシーケンス**」、PrintBitmapメソッド

SetLogo メソッド　　*第1.10版で更新*

形式　LONG SetLogo (LONG *Location*, BSTR *Data*);

パラメータ	説明
Location	設定するロゴ。PTR_L_TOPあるいはPTR_L_BOTTOMです。
Data	ロゴを作る文字。プリント可能な文字とエスケープシーケンス（ロゴ印刷のエスケープシーケンスは除く）、キャリッジリターン(13 decimal)、ニューライン/ラインフィード(10 decimal)から成ります。

詳細はBinaryConversionプロパティを参照してください。

説明　データ文字列を上端あるいは下端のロゴとして保存する時に呼び出します。

プリントデータ内に、上端ロゴあるいは下端ロゴのエスケープシーケンスを含んだPrintNormal、PrintTwoNormalあるいはPrintImmediateメソッドを呼ぶことによって、ロゴは印刷されます。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	不正な*Location*が指定されました。
その他	ResultCodeの項目を参照してください。

参照　「印字データとエスケープシーケンス」

TransactionPrint メソッド　*第1.11版で追加、第1.7版で更新*

形式　LONG TransactionPrint(LONG *Station*, LONG *Control*);

パラメータ	説明
Station	使用するステーション。PTR_S_JOURNAL、PTR_S_RECEIPTまたはPTR_S_SLIPです。
Control	一括処理。値は以下を参照。

値	意味
PTR_TP_TRANSACTION	一括処理の開始。
PTR_TP_NORMAL	バッファデータを印字して一括処理を終了する。

説明　一括処理モードに入る時または抜ける時にこのメソッドを呼び出します。

*Control*がPTR_TP_TRANSACTIONならば、一括処理モードに入ります。その後の**PrintNormal**、**CutPaper**、**RotatePrint**、**PrintBarCode**、**PrintBitmap**、**MarkFeed**、**DrawRuledLine** への呼び出しは、*Control*パラメータにPTR_TP_NORMALをセットして**TransactionPrint**が呼び出されるまで、印刷データをバッファリングします。(プリンタでバッファリングされるか、サービスオブジェクトでバッファリングされるかは、プリンタ能力に依存します。)（この場合、前記のメソッドの印刷データはバッファリングされるだけで、印刷は開始しません。また、**AsyncMode**プロパティの値は、そのオペレーションに影響しません。つまり、その要求に**OutputID**は割り当てられませんし、**OutputCompleteEvent**も通知しません。）

*Control*がPTR_TP_NORMALならば、一括処理モードを抜けます。**PrintNormal**、**CutPaper**、**RotatePrint**、**PrintBarCode**、**PrintBitmap**、**MarkFeed**、**DrawRuledLine** によりデータがバッファされていたならば、そのデータは印刷されます。一括処理全体は、一つのメッセージとして処理されます。このメソッドは、**AsyncMode**がFALSEならば、同期的に実行され、**AsyncMode**がTRUEならば、非同期的に実行されます。

ClearOutputメソッドを呼び出すことによって、一括処理モードは取り消されます。バッファされた印刷行も削除されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味

OPOS_SUCCESS　メソッドは正常終了しました。

OPOS_E_ILLEGAL　指定したステーションが存在しないか（**CapJrnPresent**、 **CapRecPresent**、**CapSlpPresent**プロパティ参照）あるいは**CapTransaction**プロパティがFALSEです。

OPOS_E_BUSY　出力中なので実行できません。（**AsyncMode**がFALSEでControlパラメータがPTR_TP_NORMALの時のみ返されます。）

OPOS_E_EXTENDED　**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN:
プリンタカバーが開いています。（**AsyncMode**がFALSEでControlパラメータがPTR_TP_NORMALの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_JRN_EMPTY:
指定されたジャーナルは、用紙がありません。（**AsyncMode**がFALSEでControlパラメータがPTR_TP_NORMALの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_JRN_CARTRIDGE_EMPTY:
ジャーナルのカートリッジは空です。（ **AsyncMode**がFALSEでControlパラメータがPTR_TP_NORMALの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_ JRN_CARTRIDGE_REMOVED:
ジャーナルのカートリッジは取り外されています。（ **AsyncMode**がFALSEでControlパラメータがPTR_TP_NORMALの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_JRN_HEAD_CLEANING:
ジャーナルのヘッドは、クリーニング中です。（ **AsyncMode**がFALSEでControlパラメータがPTR_TP_NORMALの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_EMPTY:
指定されたレシートは、用紙がありません。（**AsyncMode**がFALSEでControlパラメータがPTR_TP_NORMALの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_CARTRIDGE_EMPTY:
レシートのカートリッジは空です。
（ **AsyncMode**がFALSEでControlパラメータがPTR_TP_NORMALの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_CARTRIDGE_REMOVED:
レシートのカートリッジは取り外されています。（ **AsyncMode**がFALSEでControlパラメータがPTR_TP_NORMALの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_REC_HEAD_CLEANING:
レシートのヘッドは、クリーニング中です。
（ **AsyncMode**がFALSEでControlパラメータがPTR_TP_NORMALの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_EMPTY:
用紙が挿入されていないスリップが指定されました。
（**AsyncMode**がFALSEでControlパラメータがPTR_TP_NORMALの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_CARTRIDGE_EMPTY:
スリップのカートリッジは空です。
（ **AsyncMode**がFALSEでControlパラメータがPTR_TP_NORMALの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_CARTRIDGE_REMOVED:
スリップのカートリッジは取り外されています。（ **AsyncMode**がFALSEでControlパラメータがPTR_TP_NORMALの時のみ返されます。）

ResultCodeExtended = OPOS_EPTR_SLP_HEAD_CLEANING:
スリップのヘッドは、クリーニング中です。
（ **AsyncMode**がFALSEでControlパラメータがPTR_TP_NORMALの時のみ返されます。）

その他　　ResultCodeの項目を参照してください。

参照　　CapTransactionプロパティ、CutPaperメソッド、PrintBarCodeメソッド、PrintBitmapメソッド、PrintNormalメソッド、RotatePrintメソッド、MarkFeedメソッド、DrawRuledLineメソッド

ValidateData メソッド　　*第1.11版で追加、第1.9版で更新*

形式　LONG ValidateData(LONG *Station*, BSTR *Data*);

パラメータ	説明
Station	使用するステーション。PTR_S_JOURNAL、PTR_S_RECEIPTまたはPTR_S_SLIPです。
Data	判定されるデータ。印刷可能データとエスケープシーケンスを含みます。 詳細は**BinaryConversion**プロパティを参照してください。

説明　**PrintNormal**、**PrintTwoNormal**、**PrintImmediate**メソッドを呼び出す前に、指定したステーションにとって、おそらく一つ以上のエスケープシーケンスを含むデータシーケンスが、有効か否かを判定する時に呼び出します。

このメソッドは、全く印刷しませんが、そのステーションの能力を判定するために使用されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	データは有効です。
OPOS_E_ILLEGAL	一つ以上のエスケープシーケンスが範囲外です。しかし、コントロールは有効な選択肢を選ぶことができます。また、エスケープシーケンスが、ページモードや、右90度または左90度の回転印刷モードでサポートされていない場合も、この値が格納されます。
OPOS_E_FAILURE	一つ以上のエスケープシーケンスがサポートされていません。選ぶことのできる選択肢はありません。

OPOS_E_ILLEGALが返されるケースは、

エスケープシーケンス	状況
用紙カット	パーセンテージ'#'は正確にサポートされていません。コントロールはサポートされている最も近い値を選択します。

フィードと用紙カット	パーセンテージ'#'は正確にサポートされていません。コントロールはサポートされている最も近い値を選択します。
フィードとカットとスタンプ印刷	パーセンテージ'#'は正確にサポートされていません。コントロールはサポートされている最も近い値を選択します。
単位フィード	フィード量'#'は正確にサポートされていません。コントロールはサポートされている最も近い値を選択します。
逆フィード	逆フィード行数'#'は大きすぎます。コントロールはサポートされている最大値を選択します。
アンダーライン	太さ'#'は正確にサポートされていません。コントロールはサポートされている最も近い値を選択します。
網掛け文字	パーセンテージ'#'は正確にサポートされていません。コントロールはサポートされている最も近い値を選択します。
横倍率	倍率'#'はサポートされていません。コントロールはサポートされている最も近い値を選択します。
縦倍率	倍率'#'はサポートされていません。コントロールはサポートされている最も近い値を選択します。
カスタムカラー	カラー'#'はサポートされていません。コントロールはサポートされている最も近い値を選択します。
RGBカラー	カラー'#'はサポートされていません。コントロールはサポートされている最も近い値を選択します。

データ	状況
*data1*CR*data*2LF	（CRはキャリッジリターン、LFはラインフィードです。）データ*data1*を印刷後その行に残るために、サービスオブジェクトは、改行印字してから逆フィードを行います。データ*data2*はデータ*data1*に上書きされます。

OPOS_E_FAILUREが返されるケースは、

エスケープシーケンス	状況
(一般)	エスケープシーケンスの書式が無効です。
用紙カット	サポートされていません。
フィードと用紙カット	サポートされていません。
フィードと用紙カットとスタンプ印刷	サポートされていません。
スタンプ印刷	サポートされていません。
ビットマップ印刷	ビットマップ印刷がサポートされていないか、ビットマップ番号'#'が範囲外です。
逆フィード	サポートされていません。
フォントタイプ指定	フォント番号'#'がサポートされていません。
ボールド	サポートされていません。
アンダーライン	サポートされていません。
イタリック	サポートされていません。
反転文字	サポートされていません。
サブスクリプト	サポートされていません。
スーパースクリプト	サポートされていません。
網掛け文字	サポートされていません。
縦横1倍角	サポートされていません。
横倍角	サポートされていません。
縦倍角	サポートされていません。
縦横倍角	サポートされていません。
RGBカラー	サポートされていません。

データ	状況
data1CRdata2LF	（CRはキャリッジリターン、LFはラインフィードです。）データを印刷後その行に残ることができません。データ*data1*は1行に印刷され、データ*data2*は次の行に印刷されます。

参照　　「印字データとエスケープシーケンス」

イベント

ErrorEvent イベント　　*第1.8版で更新*

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

パラメータ	説明
ResultCode	エラーイベントを起こした原因のコード。値は**ResultCode**参照。
ResultCodeExtended	エラーイベントを起こした原因の拡張コード。下記の値を参照。
ErrorLocus	OPOS_EL_OUTPUTに設定します。非同期出力中に、エラーが起こりました。
pErrorResponse	エラーイベント応答へのポインタ。下記の値を参照。

もし、*ResultCode*がOPOS_E_EXTENDEDであれば、*ResultCodeExtended*は次に示す値のいずれかに設定されます。

値	意味
OPOS_EPTR_COVER_OPEN	プリンタカバーが開いています。
OPOS_EPTR_JRN_EMPTY	ジャーナル用紙がありません。
OPOS_EPTR_REC_EMPTY	レシート用紙がありません。
OPOS_EPTR_SLP_EMPTY	スリップ用紙が挿入されていません。
OPOS_EPTR_JRN_CARTRIDGE_REMOVED	ジャーナルステーションカートリッジはありません。
OPOS_EPTR_JRN_CARTRIDGE_EMPTY	ジャーナルカートリッジは空です。
OPOS_EPTR_JRN_HEAD_CLEANING	ジャーナルステーションヘッドはクリーニング中です。

OPOS_EPTR_REC_CARTRIDGE_REMOVED
レシートステーションカートリッジはありません。

OPOS_EPTR_REC_CARTRIDGE_EMPTY
レシートカートリッジは空です。

OPOS_EPTR_REC_HEAD_CLEANING
レシートステーションヘッドはクリーニング中です。

OPOS_EPTR_SLP_CARTRIDGE_REMOVED
スリップステーションカートリッジはありません。

OPOS_EPTR_SLP_CARTRIDGE_EMPTY
スリップカートリッジは空です。

OPOS_EPTR_SLP_HEAD_CLEANING
スリップステーションヘッドはクリーニング中です。

*pErrorResponse*によって指定された位置の内容は、OPOS_ER_RETRYのデフォルト値にプリセットされます。アプリケーションは、次に示す値の内の一つを設定します。

値	意味
OPOS_ER_RETRY	その非同期出力を再度試みます。エラー状態は抜けます。
OPOS_ER_CLEAR	その非同期出力を含むすべてのバッファリングされたデータを削除します（**ClearOutput**メソッドと同様の効果があります）。エラー状態は抜けます。

説明　エラーを察知し、POSプリンタコントロールの状態がエラー状態に移った時に通知されます。

参照　「**ステータス、結果コード、状態モデル**」

StatusUpdateEvent イベント　*第1.9版で更新*

形式　void StatusUpdateEvent (LONG *Status*);

*Status*は、次に示す値のいずれかに設定されます。

値	意味
PTR_SUE_COVER_OPEN	プリンタカバーが開いています。
PTR_SUE_COVER_OK	プリンタカバーが閉じています。
PTR_SUE_JRN_EMPTY	ジャーナル用紙がありません。
PTR_SUE_JRN_NEAREMPTY	ジャーナル用紙がニアエンドです。
PTR_SUE_JRN_PAPEROK	ジャーナル用紙が用意できました。
PTR_SUE_REC_EMPTY	レシート用紙がありません。
PTR_SUE_REC_NEAREMPTY	レシート用紙がニアエンドです。
PTR_SUE_REC_PAPEROK	レシート用紙が用意できました。
PTR_SUE_SLP_EMPTY	スリップ用紙がありません。スリップ用紙がスリップステーションの挿入口にて検出されていません。（詳細については、モデルの章を参照してください。）
PTR_SUE_SLP_NEAREMPTY	スリップ用紙がニアエンドです。
PTR_SUE_SLP_PAPEROK	スリップ用紙が挿入されました。

PTR_SUE_IDLE　全ての非同期出力が、成功するか、または削除されることによって終了しました。POSプリンタの**State**は現在OPOS_S_IDLEです。**FlagWhenIdle**プロパティは、このイベントが通知するためにTRUEでなければなりません。そして、POSプリンタコントロールは、イベントが通知する前に、そのプロパティをFALSEへ自動的にリセットします。

第 1.5 版以降 – カートリッジ状態報告

CartridgeNotifyプロパティがPTR_CN_ENABLEDであれば、次のような*Status*パラメータの*値の* **StatusUpdateEvents**が発生します。

値	意味
PTR_SUE_JRN_CARTRIDGE_EMPTY	ジャーナルカートリッジは取り替えが必要です。カートリッジは空か存在しません。
PTR_SUE_JRN_CARTRIDGE_NEAREMPTY	ジャーナルカートリッジはニアエンドです。
PTR_SUE_JRN_HEAD_CLEANING	ジャーナルカートリッジは、クリーニングを開始しました。
PTR_SUE_JRN_CARTDRIGE_OK	すべてのジャーナルカートリッジは、使用できます。
PTR_SUE_REC_CARTRIDGE_EMPTY	レシートカートリッジは取り替えが必要です。カートリッジは空か存在しません。
PTR_SUE_REC_CARTRIDGE_NEAREMPTY	レシートカートリッジはニアエンドです。
PTR_SUE_REC_HEAD_CLEANING	レシートカートリッジは、クリーニングを開始しました。
PTR_SUE_REC_CARTDRIGE_OK	すべてのレシートカートリッジは、使用できます。

PTR_SUE_SLP_CARTRIDGE_EMPTY
スリップカートリッジは取り替えが必要です。カートリッジは空か存在しません。

PTR_SUE_SLP_CARTRIDGE_NEAREMPTY
スリップカートリッジはニアエンドです。

PTR_SUE_SLP_HEAD_CLEANING
スリップカートリッジは、クリーニングを開始しました。

PTR_SUE_SLP_CARTDRIGE_OK
すべてのスリップカートリッジは、使用できます。

*電源状態通知の**StatusUpdateEvent**の値*
「第2章 共通プロパティ、メソッド、イベント」の**StatusUpdateEvent**の記述を参照。

第 1.8 版以降 – カバー（ステーション別）状態の報告

第1.8版以降、ステーション別のカバーに対するStatusUpdateEventをサポートします。プリンタがひとつのカバーしか持っていない、或いは、プリンタがどのカバーが開いたか検出／報告できない場合、従来のPTR_SUE_COVER_OPENとPTR_SUE_COVER_OKのイベントのみが通知されます。

複数のカバーをサポートしているプリンタでは、現行のアプリケーションとの互換性のために従来のイベントも通知されます。これらの場合、最初にステーション別のイベントが通知され、従来のイベントがその後で通知されます。

２つ以上のカバーが開いた場合、従来のPTR_SUE_COVER_OPENイベントは最初のカバーが開いた後に一度だけ通知されます。
PTR_SUE_COVER_OKのイベントは、全てのカバーが閉じた後に一度だけ通知されます。

イベントの*Status*属性には、状態変化を示す下記の値が追加されています。

値	意味
PTR_SUE_JRN_COVER_OPEN	ジャーナルカバーが開いています。
PTR_SUE_JRN_COVER_OK	ジャーナルカバーが閉じています。
PTR_SUE_REC_COVER_OPEN	レシートカバーが開いています。

PTR_SUE_REC_COVER_OK　レシートカバーが閉じています。

PTR_SUE_SLP_COVER_OPEN　スリップカバーが開いています。

PTR_SUE_SLP_COVER_OK　スリップカバーが閉じています。

例A：２つのカバーセンサーがありますが、どちらかのカバーが開いた時に“カバーが開いた”ことを報告するプリンタを想定してください。以下は、操作と通知されるべきStatusUpdateEventです。

操作	StatusUpdateEvent
フロントカバーを開ける	PTR_SUE_COVER_OPEN
リアカバーを開ける	(no additional SUE)
フロントカバーを閉じる	(no additional SUE)
リアカバーを閉じる	PTR_SUE_COVER_OK例B：２つのカバーセンサーがあり、その状態を独立して報告するプリンタを想定してください。以下は、操作と通知されるべき**StatusUpdateEvent**です。

操作	StatusUpdateEvent
スリップカバーを開ける	PTR_SUE_SLP_COVER_OPEN、次にPTR_SUE_COVER_OPEN
レシートカバーを開ける	PTR_SUE_REC_COVER_OPEN
スリップカバーを閉じる	PTR_SUE_SLP_COVER_OK
レシートカバーを閉じる	PTR_SUE_REC_COVER_OK、次にPTR_SUE_COVER_OK

このステータスの報告は、第1.7版以前の仕様で作成されたアプリケーションの移行を可能にします。加えて、第1.8版以降の仕様で作成されたアプリケーションに追加機能を提供します：

新しいステータスを無視する、あるいは第1.7以前の仕様で作成されたアプリケーションは、PTR_SUE_COVER_OPENとPTR_SUE_COVER_OKの **StatusUpdateEvent**に引続き応答します。（アプリケーションが予期しないステータスを無視すると想定しています。）

新しいステータスをサポートするために開発されたアプリケーションは、ステーション別のステータス(PTR_SUE_xxx_COVER_OK)に応答することが可能です。通常のステータス(PTR_SUE_COVER_OK)は追加情報を提供しません。このアプリケーションは、前述のステーション別のステータスなしで、通常のステータスを受け取った場合、通常のステータスを処理します。

第 1.9 版以降 – ファームウェアアップデート状態／進捗報告

第1.9版で追加されたファームウェアアップデート機能で、非同期でファームウェアをアップデートしている際の、状態や進捗を報告するための*Status*パラメータの値が追加されました。

説明　重要な状態の変化が起こった時に通知されます。

第　9 章
スキャナ（バーコードリーダ）

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.2	Boolean	R/W	Open
BinaryConversion	1.2	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.0	String	R	Open
Claimed	1.0	Boolean	R	Open
DataCount	1.2	Long	R	Open
DataEventEnabled	1.0	Boolean	R/W	Open
DeviceEnabled	1.0	Boolean	R/W	Open & Claim
FreezeEvents	1.0	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.0	Long	R	*適用外*
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.0	Long	R	無し
ResultCodeExtended	1.0	Long	R	Open
State	1.0	Long	R	無し
ControlObjectDescription	1.0	String	R	無し
ControlObjectVersion	1.0	Long	R	無し
ServiceObjectDescription	1.0	String	R	Open
ServiceObjectVersion	1.0	Long	R	Open
DeviceDescription	1.0	String	R	Open
DeviceName	1.0	String	R	Open
専用				
DecodeData	1.2	Boolean	R/W	Open
ScanData	1.0	String	R	Open

ScanDataLabel	1.2	String	R	Open
ScanDataType	1.2	Long	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.0	無し
Close	1.0	Open
ClaimDevice	1.0	Open
ReleaseDevice	1.0	Open & Claim
CheckHealth	1.0	Open, Claim, & Enable
ClearInput	1.0	Open & Claim
ClearInputProperties	1.10	Open & Claim
ClearOutput	1.0	*適用外*
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.0	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.0	Open, Claim, & Enable
DirectIOEvent	1.0	Open, Claim
ErrorEvent	1.0	Open, Claim, & Enable
OutputCompleteEvent	1.0	*適用外*
StatusUpdateEvent	1.3	Open, Claim, & Enable

概説

スキャナコントロールのOLEプログラマティクIDは“OPOS.Scanner”です。

機能

スキャナコントロールは、次の機能を持っています。

バーコードラベルからエンコードデータを読み込みます。

モデル

スキャナコントロールは、イベント駆動入力の一般的な入力モデルに従います。

コントロールは入力を受信したとき、DataEventを発生させます。

AutoDisableプロパティがTRUEならば、DataEventのキューイング時にコントロールは自動的にディセーブルとなります。

DataEventEnabledプロパティがTRUEならば、キューイングしたDataEventはアプリケーションに通知されます。コントロールはこのイベントを発生させる直前に、データをプロパティにコピーし、さらなるデータイベントを発生させないために、DataEventEnabledプロパティをFALSEに設定します。これにより、アプリケーションが現在の入力の処理や関連するプロパティの処理をしている間、コントロールはその後の入力データをキューイングするようになります。アプリケーションが現在の入力データの処理を終え、次のデータ処理の準備ができたとき、DataEventEnabledプロパティをTRUEに設定することにより、再びデータイベントが通知されるようになります。

入力データの読込み中または処理中に、コントロールにおいてエラーが発生した場合、ErrorEventが発行され、DataEventEnabledプロパティがTRUEのならば、アプリケーションに通知されます。

DataCountプロパティを読むことでコントロールによってキューイングされたデータイベント数が得られます。

コントロールにキューイングされたすべての入力データはClearInputメソッドを呼ぶことにより削除できます。

DataEventもしくはErrorEvent発生により入力された全てのデータプロパティは、ClearInputPropertiesメソッドを呼ぶ事によりデフォルト値に戻す事が出来ます。

スキャンされたデータは、**ScanData**プロパティに格納されます。アプリケーションが**DecodeData**プロパティにTRUEを設定したならば、データは**ScanDataLabel**と**ScanDataType**にデコードされます。

デバイスの共有

スキャナは下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、デバイスの読み込みを始める前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の必要条件については、「一覧」の表を参照してください。

プロパティ

DecodeDataプロパティ　R/W　　*第1.2版にて追加*

形式　　BOOL DecodeData;

説明　　TRUEならば、コントロールはScanDataをScanDataLabelとScanDataTypeにデコードします。

このプロパティは、Openメソッドにより FALSEに初期化されます。

戻り値　　本プロパティ設定時、次の値がResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

ScanData プロパティ

形式　　**BSTR ScanData;**

説明　　スキャナから読み込まれたバーコードデータが設定されます。
このデータのフォーマットは**BinaryConversion**プロパティの値に依存します。詳細は**BinaryConversion**プロパティを参照してください。

スキャナデータは、一般にスキャナから通知された形式です。しかし、アプリケーションのための有効な情報を含まないことから、スキャナが独自に付加したデータ（メッセージヘッダー，トレーラ情報）は取り除かれます。

ヘッダー情報はSTXキャラクタなどのプレフィックスキャラクタです。トレーラ情報はETXやCRキャラクタなどのターミネータキャラクタとスキャナによって生成されたブロックチェックキャラクタです。

もしUPC-Aの ’A’のようなシンボルキャラクタがスキャナから通知されるならば、**ScanData**はシンボルキャラクタを含みます。もしラベルにチェックデジットが含まれていて スキャナから通知されるならば、**ScanData**はチェックデジットも含みます。（スキャナの機種や設定により、シンボルキャラクタ，チェックデジットが含まれたり含まれなかったりします。スキャナコントロールは、それらが含まれていれば通知しますが、含まれていなければ 生成したり計算はしません。）
いくつかの商品には補足バーコードが付けられています。このバーコードは一般的にメインバーコードの右側に置かれて、２または５文字の情報からなります。スキャナが補足バーコードとメインバーコードが付いた商品を読み込んだ場合、補足バーコードはメインバーコードに付加され、結果として一つのラベルとしてアプリケーションに通知されます。(スキャナにより、補足バーコードの読取りを、イネーブルまたはディセーブルに設定することができるかもしれません。)

いくつかの商品には多段バーコードが付けられています。これらのバーコードは一般的に縦に並べられており、同じまたは別のシンボルキャラクタです。スキャナが多段バーコードが付いた商品を読み込むと、各段のラベルは分割したデータとしてアプリケーションに通知されます。これは、多段バーコードの標準化が現時点ではなされていないことが理由です。また、個々のバーコードデータから成る多段バーコードのすべての組み合わせを確定することはできません。このために、アプリケーションは返されたデータに基づいて多段バーコードが読み込まれたことを決定する必要があります。（スキャナにより、多段バーコードの読取りをサポートしているか、サポートしていないかもしれません）

本プロパティは、**DataEvent**が通知される直前にコントロールによって設定されます。

ScanDataLabelプロパティ　　*第1.10版にて更新*

形式　　BSTR ScanDataLabel;

説明　　デコードされたバーコードデータが設定されます。
このデータのフォーマットはBinaryConversionプロパティの値に依存します。詳細はBinaryConversionプロパティを参照してください。

DecodeDataプロパティがFALSEの時、ScanDataLabelには空の文字列(“”)が設定されます。

DecodeDataプロパティがTRUEの時、コントロールはScanDataを以下のようにScanDataLabelにデコードします。

- スキャナが生成したシンボルキャラクタが存在する場合、これは取り除かれます。
- バーコードタイプがUPC/EANと特定できるならば（シンボル識別子はスキャナにて用意される）、チェックディジットがこのプロパティの中に存在しなければなりません。スキャナハードウェアがUPC/EANチェックディジットを通知しない場合、サービスオブジェクトは、完全なUPC/EANバーコードになるようにそれを計算してこのプロパティに付加しなければなりません。
- 可変長バーコードにおいて、バーコード長が存在する場合、これは取り除かれます。

たとえば、”5 018374 827715”と印刷されたEAN-13バーコードを読込んで、スキャナから受信した場合、ScanDataに以下のように格納されます。

スキャナからの受信データ	ScanData	コメント
5018374827715	5018374827715	バーコードのみ
501837482771<CR>	501837482771	チェックデジット無しで、キャリッジリターン有り
F5018374827715<CR>	F5018374827715	スキャナ依存のシンボルキャラクタと、キャリッジリターン有り
*<STX>*F5018374827715*<ETX>*	F5018374827715	ヘッダー、シンボルキャラクタ、トレーラ有り

これらのケースでは（そして他のバリエーションでも）
ScanDataLabelにはいつも ”5018374827715”文字列が設定され、ScanDataTypeにはSCAN_SDT_EAN13が設定されなければなりません。

本プロパティは、**DataEvent**が通知される直前にコントロールによって設定されます。

ScanDataTypeプロパティ　　*第1.15.1版にて更新*

形式　　**LONG ScanDataType；**

説明　　デコードされたバーコードのラベルタイプが設定されます。

DecodeDataプロパティがFALSEの時、**ScanDataType**にはSCAN_SDT_UNKNOWNがセットされます。

DecodeDataプロパティがTRUEの時、コントロールはスキャンしたラベルタイプの識別を試みます。スキャナコントロールのヘッダファイル（OposScan.h）には、SCAN_SDTの名称で始まるいくつかのシンボルが定義されています。

以下のラベルタイプは、本リリースで定義されたものです。

値	ラベルタイプ
1次元バーコードのシンボル	
SCAN_SDT_UPCA	UPC-A
SCAN_SDT_UPCA_S	UPC-A 補足バーコード付き
SCAN_SDT_UPCE	UPC-E
SCAN_SDT_UPCE_S	UPC-E 補足バーコード付き
SCAN_SDT_UPCD1	UPC-D1
SCAN_SDT_UPCD2	UPC-D2
SCAN_SDT_UPCD3	UPC-D3
SCAN_SDT_UPCD4	UPC-D4
SCAN_SDT_UPCD5	UPC-D5
SCAN_SDT_EAN8	EAN 8 (= JAN 8)
SCAN_SDT_JAN8	JAN 8 (= EAN 8)
SCAN_SDT_EAN8_S	EAN 8 補足バーコード付き
SCAN_SDT_EAN13	EAN 13 (= JAN 13)
SCAN_SDT_JAN13	JAN 13 (= EAN 13)
SCAN_SDT_EAN13_S	EAN 13 補足バーコード付き

SCAN_SDT_EAN128	EAN-128 EAN-128は廃止され、GS1-128に置き換えられました。SCAN_SDT_EAN_128は廃止され、同じ値を持つSCAN_SDT_GS1128に置き換えられました。
SCAN_SDT_TF	Standard (or discrete) 2 of 5
SCAN_SDT_ITF	Interleaved 2 of 5
SCAN_SDT_Codabar	Codabar (NW-7)
SCAN_SDT_Code39	Code 39
SCAN_SDT_Code93	Code 93
SCAN_SDT_Code128	Code 128
SCAN_SDT_OCRA	OCR “A”
SCAN_SDT_OCRB	OCR “B”

値	ラベルタイプ
1次元バーコードのシンボル	- ***第1.8版にて追加***
SCAN_SDT_RSS14	14桁GTIN のみ – 第1.12版より廃止予定; SCAN_SDT_GS1DATABAR(同じ値)に置き換え
SCAN_SDT_RSS_EXPANDED	14桁GTIN +追加定義されたフィールド (例えば、価格、重量) – 第1.12版より廃止予定; SCAN_SDT_GS1DATABAR_E(同じ値)に置き換え

値	ラベルタイプ
1次元バーコードのシンボル	- ***第1.12版にて追加***
SCAN_SDT_GS1DATABAR	GS1 DataBar オムニディレクショナル(ノーマルまたはスタックト)
SCAN_SDT_GS1DATABAR_E	GS1 DataBar エクスパンデッド (ノーマルまたはスタックト)

値	ラベルタイプ
1次元バーコードのシンボル	- ***第1.14版にて追加***

SCAN_SDT_ITF_CK　チェック済みのチェックディジットを含む Interleaved 2 of 5

SCAN_SDT_GS1DATABAR_TYPE2　GS1 DataBar Limited

SCAN_SDT_AMES　Ames Code

SCAN_SDT_TFMAT　Matrix 2 of 5

SCAN_SDT_CODE39_CK　チェック済みのチェックキャラクタを含むCode39

SCAN_SDT_Code32　Mod 32 によるチェックキャラクタを含む Code39

SCAN_SDT_CodeCIP　Code39 CIP

SCAN_SDT_TRIOPTIC39　Tri-Optic Code39

SCAN_SDT_ISBT128　ISBT-128

SCAN_SDT_Code11　Code 11

SCAN_SDT_MSI　MSI Code

SCAN_SDT_PLESSEY　Plessey Code

SCAN_SDT_TELEPEN　Telepen

値	ラベルタイプ

*1次元バーコードのシンボル　-　**第1.15.1版にて追加***

SCAN_SDT_GS1128　GS1-128
EAN-128は同じ値を持つGS1-128に置き換えられました。
SCAN_SDT_EAN128は廃止され、
SCAN_SDT_GS1128に置き換えられました。

SCAN_SDT_DWCODE　Digital Watermarking

値	ラベルタイプ

*組み合わせコードのシンボル　-　**第1.8版にて追加***

SCAN_SDT_CCA　コンポジットコンポーネント A
最大56文字のデータ

SCAN_SDT_CCB　コンポジットコンポーネント B
最大338文字のデータ

SCAN_SDT_CCC　　コンポジットコンポーネント C
最大2361文字のデータ

値	ラベルタイプ
*組み合わせコードのシンボル　- **第1.14版にて追加***	
SCAN_SDT_TLC39	TLC-39

コンポジットコンポーネントは UPC, EAN, GS1 DataBar など、いくつかのラベルタイプの、どれでも１つと組み合わされます。コンポジットコンポーネントは1次元コンポーネントと同時に読み込まれます。その場合、**ScanDataType**にSCAN_SDT_CCA, SCAN_SDT_CCB, SCAN_SDT_CCCが設定された**DataEvent**イベントが通知されます。次に必ず1次元コンポーネントの**DataEvent**イベントが通知されます。（言い換えればサービスは同時に2つの**DataEvent**イベントをキューイングします：最初にコンポジットコンポーネント、次に1次元コンポーネント）。関連する2つの**DataEvent**イベントのデータを連結するのはアプリケーション開発者の責任です。

値	ラベルタイプ
２次元コードのシンボル	
SCAN_SDT_PDF417	PDF 417
SCAN_SDT_MAXICODE	MAXICODE

値	ラベルタイプ
*２次元コードのシンボル　- **第1. 11版にて追加***	
SCAN_SDT_DATAMATRIX	Data Matrix
SCAN_SDT_QRCODE	QR Code
SCAN_SDT_UQRCODE	Micro QR Code
SCAN_SDT_AZTEC	Aztec
SCAN_SDT_UPDF417	Micro PDF 417

値	ラベルタイプ
*２次元コードのシンボル　- **第1. 14版にて追加***	
SCAN_SDT_GS1DATAMATRIX	GS1 DataMatrix
SCAN_SDT_GS1QRCODE	GS1 QR Code

SCAN_SDT_CODE49 Code 49

SCAN_SDT_CODE16k Code 16K

SCAN_SDT_CodablockA Codablock A

SCAN_SDT_CodablockF Codablock F

SCAN_SDT_Codablock256 Codablock 256

SCAN_SDT_HANXIN Han Xin Code

値	ラベルタイプ
*２次元コードのシンボル - **第1.15.1版にて追加***	
SCAN_SDT_DOTCODE	DotCode

値	ラベルタイプ
*郵便コードのシンボル - **第1. 14版にて追加***	
SCAN_SDT_AusPost	Australian Post
SCAN_SDT_CanPost	Canada Post
SCAN_SDT_ChinaPost	China Post
SCAN_SDT_DutchKix	Dutch Post
SCAN_SDT_InfoMail	InfoMail
SCAN_SDT_JapanPost	Japan Post
SCAN_SDT_KoreanPost	Korean Post
SCAN_SDT_SwedenPost	Sweden Post
SCAN_SDT_UkPost	UK Post BPO
SCAN_SDT_UsIntelligent	US Intelligent Mail
SCAN_SDT_UsPlanet	US Planet Code
SCAN_SDT_PostNet	US Postnet

値	ラベルタイプ
特別なケース	
SCAN_SDT_OTHER	このタイプより大きいか、または等しいならば、サービスオブジェクトはOPOSで未定義のシンボルキャラクタを通知しました。

SCAN_SDT_UNKNOWN　サービスオブジェクトはバーコードシンボルを識別できませんでした。**ScanDataLabel**は、実際のバーコードタイプに適切な形式で格納されていないかもしれません。

本プロパティは、**DataEvent**が通知される直前にコントロールによって設定されます。

イベント

DataEvent イベント

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

*Status*パラメータはゼロが設定されます。

説明　　デバイスからアプリケーションに、入力データを送るために通知されます。このイベントが通知される前に、スキャナデータが**ScanData, ScanDataLabel, ScanDataType**プロパティに設定されます。

ErrorEvent イベント

形式　　**void ErrorEvent (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**ResultCode**の項目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**ResultCodeExtended**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_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が、すでにバッファリングされたデータの一部は利用できます。

*pErrorResponse*パラメータが示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_CLEAR	バッファリングされている入力データをクリアし、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、**DataEventEnabled**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、**DataEventEnabled**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　スキャナデータを読み込もうとしている間に、エラーを検出すると通知されます。

正しいアプリケーションシーケンシングが行われるよう、**DataEventEnabled**プロパティがTRUEになるまで、入力エラーイベントは通知されません。

参照　「**ステータス、結果コード、状態モデル**」

第　１０章
POSキーボード

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.2	Boolean	R/W	Open
BinaryConversion	1.2	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.0	String	R	Open
Claimed	1.0	Boolean	R	Open
DataCount	1.2	Long	R	Open
DataEventEnabled	1.0	Boolean	R/W	Open
DeviceEnabled	1.0	Boolean	R/W	Open & Claim
FreezeEvents	1.0	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.0	Long	R	*適用外*
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.0	Long	R	無し
ResultCodeExtended	1.0	Long	R	Open
State	1.0	Long	R	無し
ControlObjectDescription	1.0	String	R	無し
ControlObjectVersion	1.0	Long	R	無し
ServiceObjectDescription	1.0	String	R	Open
ServiceObjectVersion	1.0	Long	R	Open
DeviceDescription	1.0	String	R	Open
DeviceName	1.0	String	R	Open
専用				
CapKeyUp	1.2	Boolean	R	Open
EventTypes	1.2	Long	R/W	Open

POSKeyData	1.0	Long	R	Open
POSKeyEventType	1.2	Long	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.0	無し
Close	1.0	Open
ClaimDevice	1.0	Open
ReleaseDevice	1.0	Open & Claim
CheckHealth	1.0	Open, Claim, & Enable
ClearInput	1.0	Open & Claim
ClearInputProperties	1.10	*適用外*
ClearOutput	1.0	*適用外*
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.0	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.0	Open, Claim, & Enable
DirectIOEvent	1.0	Open & Claim
ErrorEvent	1.0	Open, Claim, & Enable
OutputCompleteEvent	1.0	*適用外*
StatusUpdateEvent	1.3	Open, Claim, & Enable

概説

POSキーボードコントロールのOLEプログラマティクIDは、“OPOS.POSKeyboard”です。

ここで扱うPOSキーボードとは、PCのシステムキーボードではなく、RS-232Cポートなどの別のインターフェースに付加的に接続されるPLUキーボードやメニューキーボードなどです。

したがって、外観は”POSキーボード”であっても、Windowsから見てそれがシステムキーボードである場合は、基本的には、この仕様の対象外です。

システムキーボードインターフェースに接続された”POSキーボード”の入力には、通常のWindowsAPIを使用します（標準入力系ActiveXコントロールなど）。

機能

POSキーボードコントロールは、次の機能を持っています。

POSキーボードからキーを読み込みます。

> 上記のように、POSキーボードは、システムキーボードとは別のインターフェースに付加的に接続されるキーボードの場合が一般的ですが、実装によってはシステムキーボードの一部または全部のキーからなる仮想的なキーボードであってもかまいません。

POSキーボードの1つとして表示装置を備え、入力項目を変更可能なデバイスがありますが、本仕様の規定外機能とし、**DirectIO**メソッドにより制御します。

モデル

POSキーボードコントロールは、イベント駆動入力の一般的な入力モデルに従います。

POSキーボード押下によるデータ入力を受け取ると、コントロールはDataEventをキューイングします。

AutoDisableプロパティがTRUEならば、DataEventのキューイング時にコントロールは自動的にディセーブルとなります。

DataEventEnabledプロパティがTRUEならば、キューイングしたDataEventはアプリケーションに通知されます。イベントを通知する直前にコントロールはデータをプロパティにコピーし、DataEventEnabledプロパティをFALSEに設定して次のデータイベントを抑止します。これにより、アプリケーションが現在の入力と関連プロパティを処理している間、以降の入力データをコントロールによってキューイングします。アプリケーションは、現在の入力データの処理を終えて次のデータ処理の準備ができたとき、DataEventEnabledプロパティをTRUEに設定してイベント通知を再び許可します。

入力データの読込中または処理中に、コントロールにおいてエラーが発生した場合、ErrorEventがキューイングされ、DataEventEnabledプロパティがTRUEならばアプリケーションに通知されます。

DataCountプロパティを読むことでコントロールによってキューイングされたデータイベント数が得られます。

キューイングされたすべての入力データは、ClearInputメソッドを呼び出して削除できます。

キーボードのコード変換

POSキーボードコントロールは、キーボード固有の“内部キーコード”からデータイベントによって返される“ユーザ定義キーコード”に変換するための手段を提供しなければなりません。この変換はエンドユーザにより変更ができる必要があります。

デバイスの共有

POSキーボードは、下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、デバイスの読み込みを始める前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の必要条件については、「一覧」の表を参照してください。

プロパティ

CapKeyUp プロパティ　　*第1.2版にて追加*

形式　　**BOOL　CapKeyUp;**

説明　　TRUE：KEYUPとKEYDOWNの両方のイベントを発生させることができます。これは、**EventTypes**プロパティの設定に依存します。
FALSE：KEYDOWNイベントのみの発生が可能です。

EventTypes プロパティ　R/W　　*第1.2版にて追加*

形式　　**LONG　EventTypes;**

説明　　アプリケーションが受信したいイベントタイプを選択します。

値	意味
KBD_ET_DOWN	KEYDOWNイベントを発生します。
KBD_ET_DOWN_UP	KEYDOWNとKEYUPの両方のイベントを発生します。

第1.1.1版　　：KEYDOWNイベントのみ通知可能です。

第1.2版以降：KEYDOWNとKEYUPイベントが通知可能です。

このプロパティは、**Open**メソッドにより、 KBD_ET_DOWNに初期化されます。

戻り値　　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	不正な値が設定されたか、またはコントロールはその値をサポートしていません。

POSKeyData プロパティ

形式　　LONG　POSKeyData;

説明　　POSキーボードから読み込まれた1キーデータが設定されます。

システム開発者がPOSキーボードのSO内に、内部キーコードをアプリケーションが期待するコードに変換する機能を含める設計をすれば、アプリケーションは、この値がデバイスに依存しない様に扱うことが可能です。

第1.1.1版　　：　**POSKeyData**プロパティの上位ワードを論理キーコード、下位ワードをスキャンコードと定義し、またそのコードはシステムキーボードのコードと一致する必要はありません。

第1.2版以降：　任意キーコードへの変換がエンドユーザー設定により可能であることが要求されます。

このプロパティは、**DataEvent**が通知される直前にコントロールによって設定されます。

POSKeyEventType プロパティ　　*第1.2版にて追加*

形式　　LONG　POSKeyEventType;

説明　　このプロパティは、最新のキーボードイベントのタイプを保持します。そのタイプとは、キーが押されたか、離されたか、です。

値	意味
KBD_KET_KEYDOWN	**POSKeyData**プロパティのキーが押されたことを示します。
KBD_KET_KEYUP	**POSKeyData**プロパティのキーが離されたことを示します。

このプロパティは、**DataEvent**が通知される直前にコントロールによって設定されます。

イベント

DataEvent イベント

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

*Status*パラメータは、ゼロが設定されます。

説明　　デバイスからアプリケーションに、入力データを送るために通知されます。　このイベントが通知される直前に、論理キーコードは**POSKeyData**プロパティに、イベントタイプは**POSKeyEventType**プロパティに設定されます。

ErrorEvent イベント

形式　　**void ErrorEvent (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**ResultCode**の項目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**ResultCodeExtended**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_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が、すでにバッファリングされたデータは利用できます。

*pErrorResponse*パラメータが示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_CLEAR	バッファリングされている入力データをクリアし、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、**DataEventEnabled**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、**DataEventEnabled**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　POSキーボードデータを読み込もうとしている間に、エラーを検出すると通知されます。

正しいアプリケーションシーケンシングが行われるよう、**DataEventEnabled**プロパティがTRUEになるまで、入力エラーイベントは通知されません。

参照　「**ステータス、結果コード、状態モデル**」

第　１１章
コインディスペンサ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.2	Boolean	R/W	*適用外*
BinaryConversion	1.2	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.0	String	R	Open
Claimed	1.0	Boolean	R	Open
DataCount	1.2	Long	R	*適用外*
DataEventEnabled	1.0	Boolean	R/W	*適用外*
DeviceEnabled	1.0	Boolean	R/W	Open & Claim
FreezeEvents	1.0	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.0	Long	R	*適用外*
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.0	Long	R	無し
ResultCodeExtended	1.0	Long	R	Open
State	1.0	Long	R	無し
ControlObjectDescription	1.0	String	R	無し
ControlObjectVersion	1.0	Long	R	無し
ServiceObjectDescription	1.0	String	R	Open
ServiceObjectVersion	1.0	Long	R	Open
DeviceDescription	1.0	String	R	Open
DeviceName	1.0	String	R	Open
専用				
CapEmptySensor	1.0	Boolean	R	Open

CapJamSensor	1.0	Boolean	R	Open
CapNearEmptySensor	1.0	Boolean	R	Open
DispenserStatus	1.0	Long	R	Open, Claim, & Enable

メソッド

共通	*版数*	*使用可能条件*
Open	1.0	無し
Close	1.0	Open
ClaimDevice	1.0	Open
ReleaseDevice	1.0	Open & Claim
CheckHealth	1.0	Open, Claim, & Enable
ClearInput	1.0	*適用外*
ClearInputProperties	1.10	*適用外*
ClearOutput	1.0	*適用外*
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.0	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用		
AdjustCashCounts	1.11	Open, Claim, & Enable
DispenseChange	1.0	Open, Claim, & Enable
ReadCashCounts	1.11	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.0	*適用外*
DirectIOEvent	1.0	Open, Claim
ErrorEvent	1.0	*適用外*
OutputCompleteEvent	1.0	*適用外*
StatusUpdateEvent	1.0	Open, Claim, & Enable

概説

コインディスペンサコントロールのOLEプログラマティックIDは“OPOS.CoinDispenser”です。

機能　　*第1.11版で更新*

コインディスペンサは、次の機能を持っています。

釣り銭額を指定してデバイスから払い出すメソッドをサポートします。

次の機能はオプションとして実装されます。

コインディスペンサの状態通知、これは釣り銭エンプティ、釣り銭ニアエンプティ、硬貨スロット詰まりの状態を通知します。

第1.11版より、ReadCashCountsメソッドにより報告されるデータで、硬貨枚数の相違の発生（もしくは可能性）を報告します。

モデル　　*第1.11版で更新*

コインディスペンサの一般的なモデルは、次のようなものです。

コインディスペンサは、払い出し用硬貨を蓄える複数のスロットから構成されます。コインディスペンサコントロールを利用するプログラマは、硬貨スロットの個別的な制御は行わず、払い出す釣り銭合計を指定してメソッドを呼び出します。各種スロットから適正な釣り銭を払い出すことはコインディスペンサのデバイスかコントロールの責任となります。

第1.11版以降

現金が追加された際に、硬貨枚数合計を追加することにより、プログラム的にデバイスに現金を設定します。

直接ハードウェアから、もしくは払い出しとデバイスへの追加をサービスがトレースすることにより、デバイスの硬貨枚数を読み取ります。

デバイスの共有

コインディスペンサは次のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、プロパティにアクセスしたり、釣り銭を払い出したり、状態変化を受け取る前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の必要条件については、「一覧」の表を参照してください。

プロパティ

CapEmptySensor プロパティ

形式　　BOOL CapEmptySensor;

説明　　TRUE：コインディスペンサは硬貨のエンプティを通知します。

FALSE：通知しません。

このプロパティはOpenメソッドにより初期化されます。

CapJamSensor プロパティ

形式　　BOOL CapJamSensor;

説明　　TRUE：コインディスペンサは硬貨詰まりや機器の異常を通知します。

FALSE：通知しません。

このプロパティはOpenメソッドにより初期化されます。

CapNearEmptySensor プロパティ

形式　　BOOL CapNearEmptySensor;

説明　　TRUE：コインディスペンサは硬貨のニアエンプティを通知します。

FALSE：通知しません。

このプロパティはOpenメソッドにより初期化されます。

DispenserStatus プロパティ

形式　LONG DispenserStatus;

説明　コインディスペンサの現在の状態を示します。次のいずれかとなります。

値	意味
COIN_STATUS_OK	硬貨払い出し可能。この値はコインディスペンサで異常が検出されない時に設定されます。
COIN_STATUS_EMPTY	硬貨がエンプティで払い出しができません。
COIN_STATUS_NEAREMPTY	ニアエンプティですが、釣り銭の払い出しは可能です。
COIN_STATUS_JAM	機器障害が発生しました。

このプロパティはイネーブルの設定で初期化され、イネーブルの間は最新の状態となります。このプロパティは自動釣り銭機の **DeviceStatus** と同じものです。

メソッド

AdjustCashCounts メソッド　　*第1.11版で追加*

形式　　LONG AdjustCashCounts (BSTR *CashCounts*);

パラメータ	説明
CashCount	*CashCount*パラメータは、初期化する現金の金種と枚数を指定します。

説明　　最初の設置後にコンディスペンサの開始現金有高を設定したり、入出金操作などの補充・回収後の現金有高の補正を行うため、このメソッドは呼び出されます。コインディスペンサ内の正確な現金合計を自動的に判定できないデバイスでは、このメソッドの呼び出しが必要です。デバイスが正確な合計額を判定できるなら、このメソッドの呼び出しは無視されます。アプリケーションは最初にReadCashCountsを呼び出して現在の有高を取得し、実際の有高に補正します。さらに、アプリケーションはこのメソッドを呼び出し、ディスペンサ内の現在の有高を設定するでしょう。

すべての現金有高をリセットするには、各々の金種をゼロに設定します。

例えば、通貨が日本円で*CashCounts*パラメータに“1:80,5:77,50:54,100:0,500:87”を設定してAdjustCashCountsメソッドを呼び出せば、0枚の1円と、77枚の5円と、54枚の50円と、0枚の100円と、87枚の500円の硬貨がコインディスペンサに有ります。

戻り値　　次の値のいずれかが戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは成功しました。
その他	ResultCodeの項目を参照してください。

参照　　ReadCashCountsメソッド

DispenseChange メソッド

形式　　LONG DispenseChange (LONG *Amount*);

パラメータ	説明
Amount	払い出し金額の合計。

説明　コインディスペンサから釣り銭を払い出します。*Amount*パラメータで示される値は払い出す（セントや円といった）通貨単位での合計額です。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	指定された釣り銭が正常に払い出されました。
OPOS_E_ILLEGAL	*Amount*パラメータの値が0となっています、もしくは負の値となっています、もしくはデバイスで払い出せる金額を超えています。
その他	**ResultCode**の項目を参照してください。

ReadCashCounts メソッド　*第1.11版で追加*

形式　**LONG ReadCashCounts (BSTR*** *pCashCounts*, **BOOL*** *pDiscrepancy*);

パラメータ	意味
pCashCounts	現金枚数データを格納するエリアを示します。
pDiscrepancy	TRUE： *pCashCounts*で報告された現金枚数に含まれない現金が有ります。 FALSE：現金有り高は*pCashCounts*の内容と一致します。

説明　*pCashCounts*の文字列形式はアスキー文字列です。文字列はコンマで区切られたユニットの集合です。*pCashCounts*の各々のユニットは、金種と個数をコロン（“:”）で区切って示します。

例えば、通貨が日本円で、**ReadCashCounts**メソッドの呼び出し結果として*pCashCounts*パラメータに返される文字列が、
　1:80,5:77,10:0,50:54,100:0,500:87
なら、80枚の1円硬貨と、77枚の5円硬貨と、54枚の50円硬貨、87枚の500円硬貨がコインディスペンサに有ります。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に終了しました。
その他	**ResultCode**の項目を参照してください。

イベント

StatusUpdateEvent イベント

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

*Status*パラメータはコインディスペンサの状態が設定されます。

値	意味
COIN_STATUS_OK	硬貨払い出しの準備完。この値はコインディスペンサで異常が検出されない時に設定されます。
COIN_STATUS_EMPTY	硬貨がエンプティで払い出しができません。
COIN_STATUS_NEAREMPTY	ニアエンプティですが、釣り銭の払い出しはまだできます。
COIN_STATUS_JAM	機器障害が発生しました。
*電源通知の**StatusUpdateEvent**の値*	「第2章 共通プロパティ、メソッド、イベント」の**StatusUpdateEvent**の記述を参照。

説明　　コインディスペンサの状態が変化した時に通知されます。

コインディスペンサは、機能プロパティに記された値でサポートされる検出機能を持つ場合にのみ、状態変化イベントを通知します。

第　12 章
MICR（磁気インクリーダ）

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.2	Boolean	R/W	Open
BinaryConversion	1.2	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.0	String	R	Open
Claimed	1.0	Boolean	R	Open
DataCount	1.2	Long	R	Open
DataEventEnabled	1.0	Boolean	R/W	Open
DeviceEnabled	1.0	Boolean	R/W	Open & Claim
FreezeEvents	1.0	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.0	Long	R	*適用外*
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.0	Long	R	無し
ResultCodeExtended	1.0	Long	R	Open
State	1.0	Long	R	無し
ControlObjectDescription	1.0	String	R	無し
ControlObjectVersion	1.0	Long	R	無し
ServiceObjectDescription	1.0	String	R	Open
ServiceObjectVersion	1.0	Long	R	Open
DeviceDescription	1.0	String	R	Open
DeviceName	1.0	String	R	Open

専用				
CapValidationDevice	1.0	Boolean	R	Open
RawData	1.0	String	R	Open
AccountNumber	1.0	String	R	Open
Amount	1.0	String	R	Open
BankNumber	1.0	String	R	Open
EPC	1.0	String	R	Open
SerialNumber	1.0	String	R	Open
TransitNumber	1.0	String	R	Open
CheckType	1.0	Long	R	Open
CountryCode	1.0	Long	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.0	無し
Close	1.0	Open
ClaimDevice	1.0	Open
ReleaseDevice	1.0	Open & Claim
CheckHealth	1.0	Open, Claim, & Enable
ClearInput	1.0	Open & Claim
ClearInputProperties	1.10	Open & Claim
ClearOutput	1.0	*適用外*
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.0	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用		
BeginInsertion	1.0	Open, Claim, & Enable
EndInsertion	1.0	Open, Claim, & Enable
BeginRemoval	1.0	Open, Claim, & Enable
EndRemoval	1.0	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.0	Open, Claim, & Enable

DirectIOEvent	1.0	Open & Claim
ErrorEvent	1.0	Open, Claim, & Enable
OutputCompleteEvent	1.0	*適用外*
StatusUpdateEvent	1.3	Open, Claim, & Enable

概説

MICRコントロールのOLEプログラマティクIDは “OPOS.MICR”です。

詳細はUPOS仕様書Version1.14を参照ください。

第 13 章

秤

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.3	Boolean	R/W	Open
BinaryConversion	1.2	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.0	String	R	Open
Claimed	1.0	Boolean	R	Open
DataCount	1.3	Long	R	Open
DataEventEnabled	1.3	Boolean	R/W	Open
DeviceEnabled	1.0	Boolean	R/W	Open & Claim
FreezeEvents	1.0	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.0	Long	R	*適用外*
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.0	Long	R	無し
ResultCodeExtended	1.0	Long	R	Open
State	1.0	Long	R	無し
ControlObjectDescription	1.0	String	R	無し
ControlObjectVersion	1.0	Long	R	無し
ServiceObjectDescription	1.0	String	R	Open
ServiceObjectVersion	1.0	Long	R	Open
DeviceDescription	1.0	String	R	Open
DeviceName	1.0	String	R	Open
専用				
CapDisplay	1.2	Boolean	R	Open

CapDisplayText	1.3	Boolean	R	Open
CapPriceCalculating	1.3	Boolean	R	Open
CapStatusUpdate	1.9	Boolean	R	Open
CapTareWeight	1.3	Boolean	R	Open
CapZeroScale	1.3	Boolean	R	Open
AsyncMode	1.3	Boolean	R/W	Open
MaxDisplayTextChars	1.3	Long	R	Open
MaximumWeight	1.0	Long	R	Open
SalesPrice	1.3	Currency	R	Open, Claim, & Enable
ScaleLiveWeight	1.9	Long	R	Open
StatusNotify	1.9	Long	R/W	Open
TareWeight	1.3	Long	R/W	Open, Claim, & Enable
UnitPrice	1.3	Currency	R/W	Open, Claim, & Enable
WeightUnit	1.0	Long	R	Open
ZeroValid	1.13	Boolean	R/W	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.0	無し
Close	1.0	Open
ClaimDevice	1.0	Open
ReleaseDevice	1.0	Open & Claim
CheckHealth	1.0	Open, Claim, & Enable
ClearInput	1.3	Open & Claim
ClearInputProperties	1.10	*適用外*
ClearOutput	1.0	*適用外*
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.0	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用		
DisplayText	1.3	Open, Claim, & Enable
ReadWeight	1.0	Open, Claim, & Enable
ZeroScale	1.3	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.3	Open, Claim, & Enable
DirectIOEvent	1.0	Open, Claim
ErrorEvent	1.3	Open, Claim, & Enable
OutputCompleteEvent	1.0	*適用外*
StatusUpdateEvent	1.3	Open, Claim, & Enable

概説

秤コントロールのOLEプログラマティクIDは、“OPOS.Scale”です。

UPOS仕様書1.13版までを掲載しています。

UPOS仕様書1.14版以降の変更は、UPOS仕様書を参照ください。

機能

秤コントロールは、次の機能を持っています。

商品の重量をアプリケーションに通知します。通知する重量の単位（グラム、キログラム、オンス、ポンド）は秤に依存します。

秤は以下のような追加的機能を持つ場合もあります。

秤に表示装置があり、現在の重量を表示する機能、あるいは現在の重量と何らかのアプリケーションで指定した文字列を表示する機能。

売上価格（重量 × 単価）を計算し、売上価格を返す機能（この機能は現在主にヨーロッパで使用されています）。

アプリケーションから風袋の重量を設定する機能。

アプリケーションからゼロアジャストをする機能。

以下の機能が第1.9版で追加になりました。

秤コントロールは、二つの異なる意図のために重さを取得して使用されます。価格を計算するための法定の重さと、カスタマディスプレイに表示する計量中の重さです。第1.9版以前に、よいインターフェースが法定の重さを得ることでアプリケーションのために提供されています。しかし、現在ある軽量中の重さを得るためのインターフェースがありませんでした。第1.9版で以下の追加された機能が、秤の状態と計量中の重さ得るためにインターフェースを制定します。

秤の重さの状態を更新する機能のプロパティ：CapStatusUpdate

秤の重さの状態をイベントで通知することを許可するためのプロパティ：StatusNotify

ScaleLiveWeightプロパティに含まれている値は、今の秤の重さでカスタマディスプレイを更新するために使用されます。

秤の重さの状態のために、ReadWeihtメソッドとStatusUpdateEventを拡張します。

モデル

秤の一般的なモデルは以下の通りです。

秤は載せられた物の重量を返します。

最も重要なメソッドはReadWeightです。 デフォルトではこのメソッドは非同期で動作します。 秤が重量を計測した後、メソッドの引数の *pWeightData* で示される所に重量が格納されます。 エラーもしくはタイムアウトならばReadWeightはエラーコードを返します。

第1.3版以降 － 非同期入力

もし**AsyncMode**がTRUEならば**ReadWeight**は非同期に実行されます。 **ReadWeight**メソッドによりイベント駆動入力が起動され、すぐにアプリケーションに戻ります。 引数で指定されたタイムアウト時間は重量が安定するまでアプリケーションが待つことが可能な時間です。

以下に追加事項を記述します。

- もしイベント駆動入力の初期化中にエラーが発生した場合（例えば秤がオフラインの場合）はエラーコードがReadWeightに返されます。 それ以外の場合はReadWeightが成功のステータスを返し、秤は非同期で計測を開始します。
- 計測中の重量が安定すれば、DataEventが発生し、重量が*Status*に格納されます。
- もしエラーが発生すれば（重量が安定しないタイムアウトを含む）ErrorEventが発生し、エラーコードを返します。 アプリケーションのイベントハンドラーではレスポンスのパラメーターがOPOS_ER_RETRY（ *pErrorResponse*で示される ）である場合は再実行することが可能です。
- 非同期のReadWeightを多重に発行することはできません。
- 非同期のReadWeightをClearInputでキャンセルすることが可能です。

売上価格が計算できる秤では、アプリケーションは**ReadWeight**メソッドを発行する前に**UnitPrice**を設定する必要があります。 コントロールは重量を得た後（非同期の場合は**DataEvent**をアプリケーションに通知する直前）に**SalesPrice**に計算された価格をセットします。

デバイスの共有

秤は次のように排他的に使用すべきデバイスです。

デバイスをオープン後、プロパティは読み取り可能となります。

アプリケーションは、デバイスをイネーブルにする前に、排他アクセス権を獲得しなければなりません。

アプリケーションは、デバイスの読み込みを始める前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の必要条件は「一覧」の表を参照してください。

プロパティ

AsyncMode プロパティ R/W　　*第1.3で追加*

形式　　**BOOL AsyncMode;**

説明　　TRUE:　**ReadWeight**が非同期的に実行されます。

FALSE: 同期的に実行されます。

このプロパティは、**Open**メソッドによりFALSEに初期化されます。

戻り値　　次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティが正常にセットされました。

参照　　**ReadWeight**メソッド

CapDisplay プロパティ　　*第1.2版で追加*

形式　　**BOOL CapDisplay;**

説明　　TRUE: 秤が現在の「重さを表示する装置」を持ちます。

FALSE: 持ちません。

このプロパティをアプリケーションで判断し、秤に表示装置が無い場合は別の表示装置に表示することが考えられます。

このプロパティは、**Open**メソッドにより初期化されます。

CapDisplayText プロパティ　　*第1.3版で追加*

形式　　BOOL CapDisplayText;

説明　　TRUE: 秤が現在の重量やその他載せられた物に対する情報の文字列を表示できる表示装置を持ちます。

FALSE: 持ちません。

このプロパティは、Openメソッドにより初期化されます。

参照　　MaxDisplayTextCharsメソッド

CapPriceCalculating プロパティ　　*第1.3版で追加*

形式　　BOOL CapPriceCalculating;

説明　　TRUE:　単価と重量から売上価格を計算する機能を秤は持ちます。

FALSE: 秤は重量のみ返します。

売上価格が計算できる秤では、データを受信するターミナルではなく秤で計算を行います。 売上価格が計算できる秤では、ReadWeightメソッドを呼ぶ前にUnitPriceをセットする必要があります。

このプロパティは、Openメソッドにより初期化されます。

参照　　ReadWeightメソッド、WeightUnitプロパティ、UnitPriceプロパティ、SalesPriceプロパティ

CapStatusUpdate プロパティ　　*第1.9版で追加*

形式　　BOOL CapStatusUpdate;

説明　　TRUE:　秤はStatusUpdateEventで秤の重さの状態を通知する機能を有します。デバイスが処理可能な状態にされると、直ちにStatusUpdateEventが秤の今の状態をアプリケーションに通知するために作成されます。

このプロパティは、Openメソッドにより初期化されます。

参照　　ScaleLiveWeightプロパティ、StatusNotifyプロパティ

CapTareWeight プロパティ　　*第1.3版で追加*

形式　　BOOL CapTareWeight;

説明　　TRUE:　風袋の重量を設定できる機能を秤は持ちます。

FALSE: 持ちません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　　**TareWeight**プロパティ

CapZeroScale プロパティ　　*第1.3版で追加*

形式　　BOOL CapZeroScale;

説明　　TRUE:　アプリケーションよりゼロアジャスト機能を秤は持ちます。

FALSE: 持ちません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　　**ZeroScale**メソッド

MaxDisplayTextChars プロパティ　　*第1.3版で追加*

形式　　LONG MaxDisplayTextChars;

説明　　秤に装備されている秤に載せられているものの情報を表示する表示装置が最大表示可能な文字数を示します。もし**CapDisplayText**がFALSEならば文字列を表示する装置をサポートしませんので**MaxDisplayTextChars**はゼロです。

このプロパティは、**Open**メソッドにより初期化されます。

参照　　**CapDisplayText**プロパティ

MaximumWeight プロパティ

形式　　LONG MaximumWeight;

説明　　秤で計量可能な重量の、最大値が設定されます。計量単位は**WeightUnit**プロパティで設定されます。

MaximumWeightによって設定される値は、小数点の位置が、数字の下3桁目と下4桁目の間にあると想定しています。例えば 12345 という値の場合 12.345 となり、5 の場合は 0.005 となります。

このプロパティは、**Open**メソッドにより初期化されます。

SalesPrice プロパティ　　*第1.3版で追加*

形式　　CURRENCY SalesPrice;

説明　　売上価格が計算できる秤で計算された売上価格がセットされます。秤が**UnitPrice**と測定された重量を乗算して値を求めます。

ReadWeightメソッドから戻ってくる前に（同期動作）、あるいは**DataEvent**が通知される前に（非同期動作）このプロパティはコントロールによってセットされます。

もし**CapPriceCalculating**がFALSEである場合は売上価格が計算できない秤であり**SalesPrice**は常にゼロになります。

このプロパティは、**Open**メソッド後にデバイスが最初にイネーブルされたとき、初期化されます。（1.5版以前では、**Open**メソッドで初期化されるとここで記述されていました。1.5版では、一覧の節に適合するように修正されました。）

参照　　**ReadWeight**メソッド、**WeightUnit**プロパティ、**CapPriceCalculating**プロパティ、**UnitPrice**プロパティ

ScaleLiveWeight プロパティ　　*第1.10版で変更*

形式　　LONG ScaleLiveWeight;

説明　　StatusUpdateEventの*Status*がSCAL_SUE_STABLE_WEIGHTかゼロにセットされているならば、秤によって計測された重さの戻り値が格納されます。

重さは、１／１０００で割り当てられると仮定されます。たとえば、１２３４５の正確な値は１２．３４５であらわされ、５の正確な値は、０．００５で表現されます。

アプリケーションは表示する目的にのみこのプロパティを使用することを提案します。売上げの目的で使用する重さのために、アプリケーションはReadWeightメソッドを呼ぶことを提案します。

参照　　CapStatusUpdateプロパティ、StatusNotifyプロパティ

StatusNotify プロパティ R/W　　*第1.9版で追加*

形式　　**LONG StatusNotify;**

説明　　デバイスの機能を示す**CapStatusUpdate**プロパティがTrueならば、秤の重さの状態を通知することがアプリケーションによってセットされることが出来ます。

有効な値は次の通りです。

値	意味
SCAL_SN_DISABLED	OPOSコントロールは、アプリケーションにいかなる秤の重さの状態の通知を行いません。あるいは、関連する*ErrorCodeExtended*の値にセットしません。秤の重さの状態を通知する**StatusUpdateEvent**が発行されることはありませんし、**SalesLiveWeight**に値がセットされることもありません。
SCAL_SN_ENABLED	**DeviceEnabled**がTrueをセットされたときに、OPOSコントロールは、秤の重さ情報の通知で**StatusUpdateEvent**を発行し、**ScaleLiveWeight**プロパティが始めに更新されます。機能のレベルは、**CapStatusUpdate**プロパティに依存します。

このプロパティは、デバイスが無効な状態（**DeviceEnabled**がFalseです）で値がセットされ、。**Open**メソッドによってSCAL_SN_DISABLEDで初期化されます。この値は、初期段階でデバイスとの共通性を提供します。

戻り値　　プロパティがセットされた時は次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティが正常にセットされました。
OPOS_E_ILLEGAL	以下のエラーのいづれかが発生します。 デバイスが既に有効です。 **CapStatusUpdate**がFalseです。
その他	**ResultCode**の項目を参照してください。

参照　　**CapStatusUpdate**プロパティ、**ScaleLiveWeight**プロパティ

TareWeight プロパティ R/W *第1.3版で追加*

形式 LONG TareWeight;

説明 秤の風袋の重量を保持します。 **TareWeight**プロパティにセットされている重量は、小数点の位置が、数字の下3桁目と下4桁目の間にあると想定しています。 例えば 12345 という値の場合 12.345 となり、5 の場合は 0.005 となります。 計量の単位は**WeightUnit**プロパティにセットされています。 もし**CapTareWeight**プロパティがFALSEの場合はデバイスが風袋をサポートしていませんので**TareWeight**は常にゼロになります。

TareWeightプロパティで示される風袋の重量は**ReadWeight**メソッドで返される重量には含まれません。

このプロパティは、**Open**メソッド後にデバイスが最初にイネーブルされたとき、秤のデフォルト値に初期化されます（通常はゼロです）。（1.5版以前では、**Open**メソッドで初期化されるとここで記述されていました。1.5版では、一覧の節に適合するように修正されました。）

戻り値 プロパティがセットされた時は次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティが正常にセットされました。
OPOS_E_ILLEGAL	**CapTareWeight**がFALSEの場合か、異常な風袋の重量がセットされました。
その他	**ResultCode**の項目を参照してください。

参照 **CapTareWeight**プロパティ、**ReadWeight**メソッド、**WeightUnit**プロパティ

ZeroValid プロパティ R/W　　*第1.13で追加*

形式　　BOOL ZeroValid;

説明　　TRUE:　**ReadWeight**は有効な規定値として０を返します。

FALSE: **ReadWeight**は０の値を返しません。

このプロパティは、**Open**メソッドによりFALSEに初期化されます。

参照　　**ReadWeight**メソッド

UnitPrice プロパティ R/W　　*第1.3版で追加*

形式　　**CURRENCY UnitPrice;**

説明　　秤に測定する物の単価を保持します。 売上価格が計算できる秤ではこのプロパティを計量する前にセットします。 秤そのものが**UnitPrice**と**ReadWeight**メソッドの引数で示される*pWeightData*を乗算して**SalesPrice**を求め、セットします。 このプロパティには単価のみをセットします。

もし**CapPriceCalculating**がFALSEの場合、秤は売上価格が計算できません。 この場合、単価設定の機能はサポートされていないため、**UnitPrice**には常にゼロがセットされます。

このプロパティは、**Open**メソッド後にデバイスが最初にイネーブルされたとき、ゼロに初期化されます。（1.5版以前では、**Open**メソッドで初期化されるとここで記述されていました。1.5版では、一覧の節に適合するように修正されました。）

戻り値　　次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティが正常にセットされました。
OPOS_E_ILLEGAL	**CapPriceCalculating**がFALSEの場合か異常な単価がセットされました。
その他	**ResultCode**の項目を参照してください。

参照　　**ReadWeight**メソッド、**WeightUnit**プロパティ、**CapPriceCalculating**プロパティ、**SalesPrice**プロパティ

WeightUnit プロパティ

形式　　LONG WeightUnit;

LONG WeightUnits;　　(WeightUnit[9]と同義)

説明　　秤データの計量単位が設定されます。

有効な値は次の通りです。

値	意味
SCAL_WU_GRAM	単位はグラムです。
SCAL_WU_KILOGRAM	単位はキログラムです。(= 1000グラム)
SCAL_WU_OUNCE	単位はオンスです。
SCAL_WU_POUND	単位はポンドです。(= 16オンス)

このプロパティは、**Open**メソッドにより秤固有の計量単位に初期化されます。

[9] 秤コントロールオブジェクトは**WeightUnit**プロパティをサポートしなければなりません。 **WeightUnit**に加えてAPG第1.1版以前での誤記に従い、同等の機能をもつ**WeightUnits**プロパティもサポートすることをお薦めします。

メソッド

DisplayText メソッド　　*第1.7版で更新*

形式　LONG DisplayText (BSTR *Data*);

パラメータ	説明
Data	表示すべき文字列 このデーターのフォーマットは**BinaryConversion**のプロパティに従います。詳細は**BinaryConversion**プロパティを参照してください。

説明　このメソッドにより秤に付属するテキスト表示装置に文字列を表示します。 空白の文字列（“”）を送付することにより表示装置をクリアすることができます。

もし**CapDisplayText**がFALSEの場合、デバイスは文字列表示装置をサポートしていませんので、**DisplayText**メソッドは常に失敗します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効なテキストが指定されました。テキスト長が**MaxDisplayTextChars**を超えています。 あるいは文字列表示が不可能です。
その他	**ResultCode**の項目を参照してください。

参照　**CapDisplay**プロパティ、**CapDisplayText**プロパティ、**MaxDisplayTextChars**プロパティ

ReadWeight メソッド　　*第1.10版で更新*

形式　　**LONG ReadWeight (LONG*** *pWeightData*, **LONG** *Timeout*);

パラメータ	説明
pWeightData	**AsyncMode**がFALSEの場合は計量結果のポインタ。 TRUEの場合はゼロ。
Timeout	秤の計量値が安定するまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-1)、0、正の値が指定できます。

説明　　秤から計量値を読み取ります。

第１.０版～第１.２版

計量は同期的に動作し、メソッドは計量が完了してから戻ります。 重量は*pWeightData*に示されるところに返されます。

第１.３版以降

もし**AsyncMode**がFALSEの場合、第１.２版と同様に**ReadWeight**は同期的に実行されます。

もし**AsyncMode**がTRUEの場合、 **ReadWeight**は非同期的に実行されます。 メソッドは計量を開始させ、すぐに戻ってきます。 もしメソッドの戻り値が成功のステータスならば、計量は正常に開始されていて、**DataEvent**が計量の完了時に通知され、計量した重量が*Status*にセットされています。

計量値は、小数点の位置が、数字の下3桁目と下4桁目の間にあると想定しています。例えば 12345 という値の場合 12.345 となり、5 の場合は 0.005 となります。

*Timeout*が0の場合、秤が安定していればOPOS_SUCCESSを返し、安定していなければ、OPOS_E_TIMEOUTを返します。また、*Timeout*がOPOS_FOREVER(-1)の場合は、タイムアウトは発生せず、秤が安定するまで無限に待ちます。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	有効な計量値が読み込まれ通知されました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。
OPOS_E_BUSY	非同期の**ReadWright**メソッドが実行されています。
OPOS_E_TIMEOUT	設定されている*Timeout*ミリ秒の間に、秤の"0"でない計量値が安定しませんでした。(**AsyncMode**がFALSEの場合のみ)
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_ESCAL_OVERWEIGHT: 計量値が**MaximumWeight**を超えています。
以下のデバイス固有のエラーコードが**第 *1.9*版で追加**されました。	
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_ESCAL_UNDER_ZERO 計量値が**MaximumWeight**を超えています。
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_ESCAL_SAME_WAIGHT: 計量値が**MaximumWeight**を超えています。
その他	**ResultCode**の項目を参照してください。

参照 **UnitPrice**プロパティ、**WeightUnit**プロパティ、**CapPriceCalculating**プロパティ、**SalesPrice**プロパティ、**TareWeight**プロパティ

ZeroScale メソッド *第1.3版で追加*

形式　LONG ZeroScale ();

説明　秤のゼロアジャストを行います。 これは初期時に「校正」する、あるいは「風袋の設定」のために用います。

CapZeroScaleがTRUEの場合のみ実行が可能です。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	秤は**ZeroScale**をサポートしていません。
OPOS_E_BUSY	非同期の**ReadWright**メソッドが実行されています。
その他	**ResultCode**の項目を参照してください。

参照　**CapZeroScale**プロパティ

イベント

DataEvent イベント *第1.3版で追加*

形式　　void DataEvent (LONG *Status*);

*Status*には重量が格納されます。

説明　　非同期動作の場合で**ReadWeight**で開始された計量が終了したことをアプリケーションに通知します。

もし売上価格が計算できる秤では、このイベントが通知される前に単価を**UnitPrice**プロパティにセットしておけば、計算された売上価格が**SlaesPrice**プロパティに格納されます。

ErrorEvent イベント *第1.3版で追加*

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

パラメータ	説明
ResultCode	エラーイベントを起こした要因のコード。 値は**ResultCode**の項目を参照してください。
ResultCodeExtended	エラーイベントを起こした要因の拡張コード。 値は**ResultCodeExteded**参照。
ErrorLocus	エラーの所在を示します。 下記の値参照。
pErrorResponse	エラーイベント応答へのポインタ。 下記の値を参照。

*ErrorLocus*は次に示す値のいずれかに設定されます。

値	意味
OPOS_EL_INPUT	イベント駆動入力で、入力データ処理をしている時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時にエラーが発生しましたが、すでにバッハリングされたデータの一部は利用できます。

*pErrorResponse*パラメータが示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_CLEAR	バッファリングされている入力データをクリアし、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の場合のみ使用します。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、**DataEventEnabled**プロパティの設定値に従って別の**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、**DataEventEnabled**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値です。

説明　秤よりデータを読み込もうとしているときにエラーが発生すると、本イベントを通知します。

入力のエラーイベントは、**DataEventEnabled**プロパティがTRUEになるまで実行されません。そのためアプリケーションを適切な順番で実行させることができます。

参照　「**ステータス、結果コード、状態モデル**」参照

StatusUpdateEvent イベント

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

秤の電源状態に変化があるとアプリケーションに通知されます。

StatusNotifyプロパティがSCAL_SN_ENABLEDならば、スケールデバイスで計測値に変化があると、イベントがアプリケーションに通知されます。

CapStatusNotifyプロパティがTRUEならば、計測が可能になり、秤の最新の値をアプリケーションに通知するために、即座に**StatusUpdateEvent**が生成されます。

値	意味
SCAL_SUE_STABLE_WEIHT	秤の計測値は安定しています。**ScaleLiveWeight**プロパティは、イベントが通知される前に更新されています。
SCAL_SUE_WEIGHT_UNSTABLE	秤の計測値が安定していません。
SCAL_SUE_WEIGHT_ZERO	秤の計測値は０です。
SCAL_SUE_OVERWEIGHT	秤の計測範囲を超えています。
SCAL_SUE_NOT_READY	計測する準備が出来ていません。
SCAL_SUE_WEIGHT_UNDER_ZERO	秤の計測値が０以下です。

説明　　*電源通知の**StatusUpdateEvent**の値*
「第2章 共通プロパティ、メソッド、イベント」の**StatusUpdateEvent***の記述を参照。*

参照　　**ScaleLiveWeightght**プロパティ、**StatusNotify**プロパティ

第　１４章
シグニチャ・キャプチャ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.2	Boolean	R/W	Open
BinaryConversion	1.2	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.0	String	R	Open
Claimed	1.0	Boolean	R	Open
DataCount	1.2	Long	R	Open
DataEventEnabled	1.0	Boolean	R/W	Open
DeviceEnabled	1.0	Boolean	R/W	Open & Claim
FreezeEvents	1.0	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.0	Long	R	*適用外*
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.0	Long	R	無し
ResultCodeExtended	1.0	Long	R	Open
State	1.0	Long	R	無し
ControlObjectDescription	1.0	String	R	無し
ControlObjectVersion	1.0	Long	R	無し
ServiceObjectDescription	1.0	String	R	Open
ServiceObjectVersion	1.0	Long	R	Open
DeviceDescription	1.0	String	R	Open
DeviceName	1.0	String	R	Open
専用				
CapDisplay	1.0	Boolean	R	Open
CapRealTimeData	1.2	Boolean	R	Open

CapUserTerminated	1.0	Boolean	R	Open
MaximumX	1.0	Long	R	Open
MaximumY	1.0	Long	R	Open
RawData	1.0	String	R	Open, Claim, & Enable
RealTimeDataEnabled	1.2	Boolean	R/W	Open
TotalPoints	1.0	Long	R	Open, Claim, & Enable
PointArray	1.0	String	R	Open, Claim, & Enable

メソッド

共通	*版数*	*使用可能条件*
Open	1.0	無し
Close	1.0	Open
ClaimDevice	1.0	Open
ReleaseDevice	1.0	Open & Claim
CheckHealth	1.0	Open, Claim, & Enable
ClearInput	1.0	Open & Claim
ClearInputProperties	1.10	Open & Claim
ClearOutput	1.0	*適用外*
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.0	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用		
BeginCapture	1.0	Open, Claim, & Enable
EndCapture	1.0	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.0	Open, Claim, & Enable
DirectIOEvent	1.0	Open, Claim
ErrorEvent	1.0	Open, Claim, & Enable
OutputCompleteEvent	1.0	*適用外*
StatusUpdateEvent	1.3	Open, Claim, & Enable

概説

シグニチャ・キャプチャ(Signature Capture)コントロールのOLEプログラマティックIDは、“OPOS.SigCap”です。

詳細はUPOS仕様書Version1.14を参照ください。

第　15　章
自動釣り銭機

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.2	Boolean	R/W	*適用外*
BinaryConversion	1.2	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.2	String	R	Open
Claimed	1.2	Boolean	R	Open
DataCount	1.5	Long	R	Open
DataEventEnabled	1.5	Boolean	R/W	Open
DeviceEnabled	1.2	Boolean	R/W	Open & Claim
FreezeEvents	1.2	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.2	Long	R	*適用外*
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.2	Long	R	無し
ResultCodeExtended	1.2	Long	R	Open
State	1.2	Long	R	無し
ControlObjectDescription	1.2	String	R	無し
ControlObjectVersion	1.2	Long	R	無し
ServiceObjectDescription	1.2	String	R	Open
ServiceObjectVersion	1.2	Long	R	Open
DeviceDescription	1.2	String	R	Open
DeviceName	1.2	String	R	Open

専用

CapDeposit	1.5	Boolean	R	Open
CapDepositDataEvent	1.5	Boolean	R	Open
CapDiscrepancy	1.2	Boolean	R	Open
CapEmptySensor	1.2	Boolean	R	Open
CapFullSensor	1.2	Boolean	R	Open
CapJamSensor	1.11	Boolean	R	Open
CapNearEmptySensor	1.2	Boolean	R	Open
CapNearFullSensor	1.2	Boolean	R	Open
CapPauseDeposit	1.5	Boolean	R	Open
CapRealTimeData	1.11	Boolean	R	Open
CapRepayDeposit	1.5	Boolean	R	Open
AsyncMode	1.2	Boolean	R/W	Open
AsyncResultCode	1.2	Long	R	Open, Claim, & Enable
AsyncResultCodeExtended	1.2	Long	R	Open, Claim, & Enable
CurrencyCashList	1.2	String	R	Open
CurrencyCode	1.2	String	R/W	Open
CurrencyCodeList	1.2	String	R	Open
CurrentExit	1.2	Long	R/W	Open
CurrentService	1.11	Long	R/W	Open
DepositAmount	1.5	Long	R	Open
DepositCashList	1.5	String	R	Open
DepositCodeList	1.5	String	R	Open
DepositCounts	1.5	String	R	Open
DepositStatus	1.5	Long	R	Open, Claim, & Enable
DeviceExits	1.2	Long	R	Open
DeviceStatus	1.2	Long	R	Open, Claim, & Enable
ExitCashList	1.2	String	R	Open
FullStatus	1.2	Long	R	Open, Claim, & Enable
RealTimeDataEnabled	1.11	Boolean	R/W	Open, Claim, & Enable
ServiceCount	1.11	Long	R	Open
ServiceIndex	1.11	Long	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.2	無し
Close	1.2	Open
ClaimDevice	1.2	Open
ReleaseDevice	1.2	Open & Claim
CheckHealth	1.2	Open, Claim, & Enable
ClearInput	1.5	Open & Claim
ClearInputProperties	1.10	*適用外*
ClearOutput	1.2	*適用外*
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.2	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用		
AdjustCashCounts	1.11	Open, Claim, & Enable
BeginDeposit	1.5	Open, Claim, & Enable
DispenseCash	1.2	Open, Claim, & Enable
DispenseChange	1.2	Open, Claim, & Enable
EndDeposit	1.5	Open, Claim, & Enable
FixDeposit	1.5	Open, Claim, & Enable
PauseDeposit	1.5	Open, Claim, & Enable
ReadCashCounts	1.2	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.5	Open, Claim&Enable
DirectIOEvent	1.2	Open, Claim
ErrorEvent	1.2	*適用外*
OutputCompleteEvent	1.2	*適用外*
StatusUpdateEvent	1.2	Open, Claim, & Enable

概説

自動釣り銭機コントロールのOLEプログラマティクIDは“OPOS.CashChanger”です。

このデバイスは第1.2版にて追加されました。

機能 *第1.11版で更新*

自動釣り銭機は次の機能を持っています。

自動釣り銭機で利用できる金種と、対応する枚数を報告します。

指定された合計の現金を、紙幣、硬貨、もしくは両者で、デバイスからユーザー指定の払い出し口へ払い出します。

指定された枚数の金種を、紙幣、硬貨、もしくは両者で、デバイスからユーザー指定の払い出し口へ払い出します。

デバイス内の障害状態を報告します。

複数の通貨をサポートします。

また、次の機能は、オプションとして実装されます。

自動釣り銭機の金種の充填レベルを報告します。示される状態としては、エンプティ、ニアエンプティ、フル、ニアフルがあります。

ReadCashCountsメソッドで報告されるデータに、現金枚数不一致がありそう（もしくは、ある）かどうかが報告されます。

第1.5版以降 — オプションとして入金機能をサポート

入金開始から入金終了の間に入金された紙幣／硬貨は入金されるたびにその金種と対応する枚数を報告します。

第1.11版以降 — キャッシュデバイスサブサービス利用のサポート

自動釣り銭機サービスの全機能を利用できるようにするため、他のキャッシュデバイスをサブサービスとして利用できます。サブサービスからの情報を抽出するためのプロパティが追加されています。

モデル　*第1.11版で更新*

自動釣り銭機の一般的なモデルは次の通りです。

自動釣り銭機は、硬貨、紙幣、硬貨と紙幣の組み合わせといった各種の現金タイプをサポートします。各通貨毎にサポートされる現金タイプは、CurrencyCashListプロパティの金種リストに記述されています。

自動釣り銭機デバイスは、現金投入口、現金を格納する複数のスロットや収納庫、現金払い出し口といった現金処理機能を援助する各種の要素から構成されています。

第1.5版より前の仕様では、*現金の払い出しのみ*をプログラムで制御できました。デバイスが自分自身の現金有り高を決定できない場合、（例えば、現金補充のための）デバイスでの現金の受け取りや回収は、AdjustCashCountsメソッドにより制御されます。アプリケーションは金種毎の現在の枚数を読み出すためにReadCashCountsメソッドを呼び出すことができますが、デバイスに何時・どれくらい追加されたかを制御することはできません。

自動釣り銭機は各種の払い出し口を持つことがあります。払い出し口の個数はDeviceExitsプロパティで規定されます。アプリケーションはCurrentExitプロパティを設定して払い出し口を選択します。現在の払い出し口に払い出せる金種は、ExitCashListプロパティで表されます。CurrentExitの値が1なら、払い出し口は基本払い出し口となり、小売り販売に伴う顧客への現金払い出し処理に通常使用されます。CurrentExitの値が1より大きいなら、払い出し口はその他払い出し口_となります。その他払い出し口は、基本払い出し口を出口としない、多量の、もしくは別金種の現金の払い出しといった特殊な目的に通常使用されます。

CurrentExitプロパティで指定される払い出し口へ現金を払い出すには、DispenseChangeもしくはDispenseCashメソッドを使用します。DispenseChangeメソッドでアプリケーションは合計金額を指定して払い出しますが、各種スロットもしくは収納庫から適切な現金を払い出すことは、自動釣り銭機デバイスまたはコントロールの責任となります。DispenseCashメソッドでアプリケーションは払い出す金種毎の枚数を指定します。

現金の払い出しは、AsyncModeプロパティの値に従い、同期的、もしくは非同期的に実行されます。

> AsyncModeがFALSEなら、現金払い出しメソッドは同期的に実行され、払い出しメソッドはアプリケーションに終了ステータスを返します。
>
> AsyncModeがTRUEで、DispenseChangeもしくはDispenseCashがOPOS_SUCCESSを返したら、メソッドは非同期的に実行され、終了はCHAN_STATUS_ASYNCを*Status*の値に持つStatusUpdateEventイベントで示されます。

AsyncResultCodeとAsyncResultCodeExtendedプロパティの値は、同期的な払い出しが選択された際に返されるResultCodeとResultCodeExtendedプロパティの値と同じです。

非同期的な自動釣り銭機の操作を多重に行うことは禁止されています。同時に処理される非同期的なメソッドは1つだけです。

ReadCashCountsは、非同期メソッドが実行中の間、不正な現金枚数を報告する可能性があるため、実行できません。

自動釣り銭機は複数の通貨をサポートできます。CurrencyCodeプロパティは、CurrencyCodeListの一覧から選択された通貨が設定されます。CurrencyCashList、ExitCashListプロパティとDispenseCash、DispenseChange、ReadCashCountsメソッドは、現在設定されている通貨上で動作します。

現金スロット（もしくは現金収納庫）の状態は、エンプティとニアエンプティを示すDeviceStatusプロパティと、フルとニアフルを示すFullStatusプロパティに設定されます。1つ以上の現金スロットがエンプティならDeviceStatusプロパティはCHAN_STATUS_EMPTYとなり、1以上の現金スロットがフルならFullStatusプロパティはCHAN_STATUS_FULLとなります。

第1.5版以降　―オプションの入金機能サポート

入金機能モデルは下記のとおりです。

DispenseChangeメソッドとDispenseCashメソッドは入金機能とは独立しています。AsynModeプロパティの値は入金機能に影響を与えません。

CapDeposit プロパティがTRUEなら入金機能がサポートされており、BeginDepositメソッド、FixDepositメソッドとEndDepositメソッドを使うことができます。入金が動作中の間、出金はできません。また、出金が動作中の間、入金はできません。

アプリケーションがBeginDepositメソッドを呼び出した時、DepositCountsプロパティとDepositAmountプロパティはゼロに初期化されます。

自動釣り銭機が現金を受け付けた時、DepositCountsプロパティとDepositAmountプロパティの値は更新されます。さらに、CapDepositDataEventプロパティがTRUEならDataEventイベントが報告されます。

CapPauseDepositプロパティがTRUEなら、アプリケーションはCHAN_DEPOSIT_PAUSEパラメータのPauseDepositメソッドを呼び出して入金を一時停止することができます。その時、DepositCountsとDepositAmountプロパティの値が確定します。アプリケーションが入金を再開したいなら、CHAN_DEPOSIT_RESTARTのパラメータ値でPauseDepositメソッドを再度呼び出します。DepositCountsプロパティかDepositAmountプロパティを読み取り、アプリケーションは正確な入金を得ることができ、入金合計を販売合計と比較した結果をチェックできます。入金合計≧販売合計なら、アプリケーションはFixDepositメソッドを呼び出します。入金合計＜販売合計なら、アプリケーションはCHAN_DEPOSIT_RESTARTでPauseDepositメソッドを呼び出します。入金総額を得るため、アプリケーションはPauseDepositメソッドを繰り返し呼び出せます。

FixDepositメソッドが呼び出された時、自動釣り銭機は入金を停止し、DepositCountsプロパティとDepositAmountプロパティの値が確定します。

必要なら、アプリケーションはFixDepositメソッドの後のDepositCountsプロパティとDepositAmountプロパティを使用して釣り銭額を計算しなくてはなりません。FixDepositメソッドが呼び出された後、アプリケーションは入金を再開することはできず、入金を完了するためにEndDepositメソッドを呼び出さなくてはなりません。

入金プロセスを終了させるために、アプリケーションはEndDepositメソッドを呼び出さなくてはなりません。

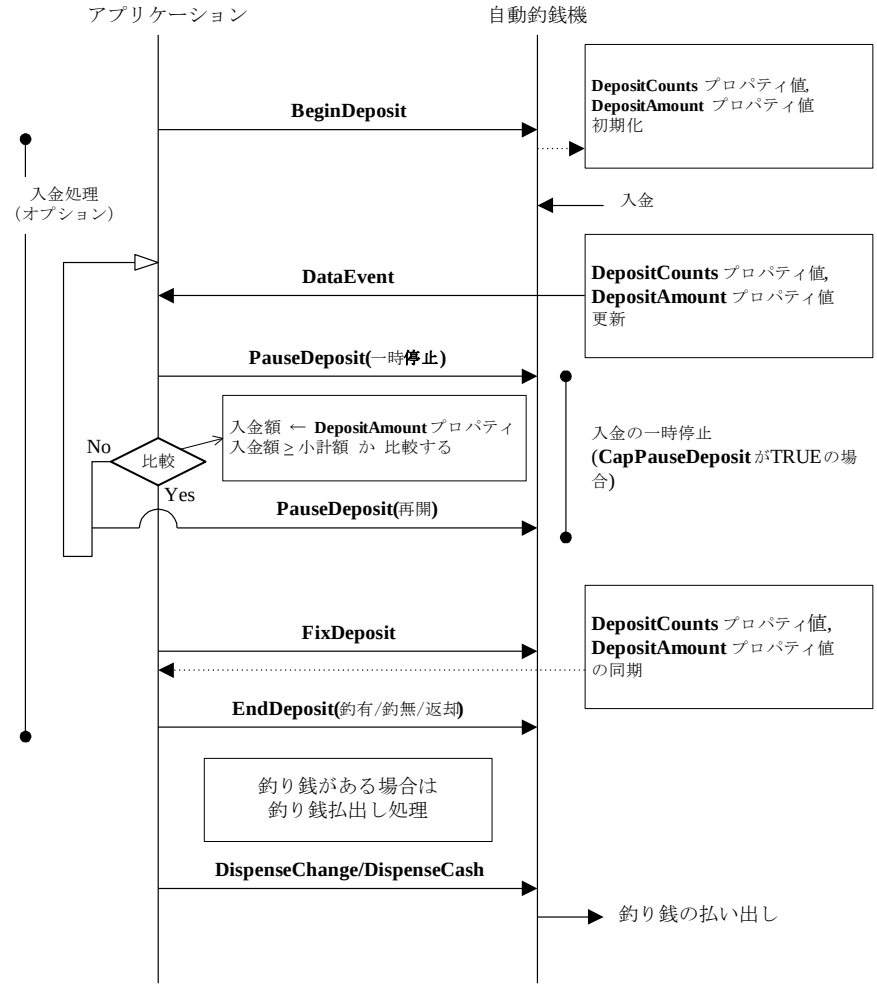
EndDepositメソッドの*Success*パラメータの値は次の値をとります。

- CHAN_DEPOSIT_CHANGE：アプリケーションは現金を払い出します。これは販売の釣り銭払い出しに典型的に用いられます。
- CHAN_DEPOSIT_NOCHANGE： アプリケーションは現金を払い出しません。典型的には、入金総合計が販売額合計に等しいことを示します。
- CHAN_DEPOSIT_REPAY： アプリケーションは入金を全て払い戻します。この典型は販売の中止です。
  CHAN_DEPOSIT_REPAYはCapRepayDepositがTRUEの時にのみ使用できます。CapRepayDepositがFALSEならアプリケーションはDispenseChangeかDispenseCashメソッドを呼び出して入金金額を払い出さなくてはなりません。
  自動釣り銭機によっては、自動釣り銭機のメカニズムにより入金された現金自身を使用しない場合があります。

ClearInputメソッドが実行された時、入金に伴いキューされた全てのDataEventはクリアされますが、DepositCountsプロパティとDepositAmountプロパティはクリアされません。

入金プロセスで、ErrorEventイベントとAutoDisableプロパティは使用されません。

入金プロセスは下図のダイアグラムの様になります。



第1.11版以降 — キャッシュデバイスサブサービス利用のサポート

キャッシュデバイスのサブサービスモデルは以下の通りです：

自動釣り銭機サービスへ機能を提供するコインディスペンサや硬貨入金機や紙幣ディスペンサや紙幣入金機、その他のデバイスハードウェアへアクセスする自動釣り銭機に対し、自動釣り銭機はサブサービスとしてこれらのキャッシュデバイスを利用可能にします。自動釣り銭機サービスに対する呼び出しにより、対応するサブサービスの呼び出しが実行されます。このため、オープン呼び出しは全てのサブサービスのオープンメソッドを引き起こし、クレーム呼び出しもクレーム呼び出しを引き起こし、その他も同様です。あるプロパティはディスペンサに対して有効ですが、他のプロパティは入金機にのみ有効となります。サブサービスとの通信結果を解析して解釈し、アプリケーションへの報告するかは、自動釣り銭機サービスで統合されます。たとえば、オープン呼び出しが一つのサブサービスで失敗した場合、アプリケーションまでエラーが返るでしょう。サービスからサブサービスへのプロパティとメソッドのマッピングは、次のようになります。

自動釣り銭機	コインディスペンサ	紙幣ディスペンサ	硬貨入金機	紙幣入金機
CapDeposit				
CapDepositDataEvent				
CapDiscrepancy	X	X	X	X
CapEmptySensor	X	X		
CapJamSensor	X	X	X	X
CapFullSensor			X	X
CapNearEmptySensor	X	X		
CapNearFullSensor			X	X
CapPauseDeposit			X	X
CapRealTimeData			X	X
CapRepayDeposit				
AsyncMode		X		
AsyncResultCode		X		
AsyncResultCodeExtended		X		
CurrencyCashList		X		
CurrencyCode		X	X	X
CurrencyCodeList		X		
CurrentExit		X		
CurrentService				
DepositAmount			X	X
DepositCashList			X	X

DepositCodeList			X	X
DepositCounts			X	X
DepositStatus			X	X
DeviceExits		X		
DeviceStatus	**DispenserStatus**	X		
ExitCashList		X		
FullStatus			X	X
ServiceCount				
ServiceIndex				
RealTimeDataEnabled			X	X
BeginDeposit()			X	X
DispenseCash()		X		
DispenseChange( )	X			
EndDeposit()			X	X
FixDeposit( )			X	X
PauseDeposit()			X	X
ReadCashCounts()	X	X	X	X

ServiceCountは、自動釣り銭機で使用されるサブサービスの個数を表します。

ServiceIndexは、各々のサブサービスのインデックスをバイト毎に表すプロパティです。

サブサービスのプロパティやメソッドにアクセスする必要があるなら、CurrentServiceプロパティを必要なインデックスに設定し、アプリケーションは特定のサブサービスへ適切な情報を要求します。

デバイスの共有

自動釣り銭機は次の通り、排他的な利用をするデバイスです。

アプリケーションはデバイスをイネーブルにする前に、デバイスの排他アクセス権を獲得しなければなりません。

アプリケーションは、プロパティにアクセスしたり、払い出しや回収を行ったり、状態更新イベントを受け取る前に、デバイスの排他アクセス権を獲得しイネーブルにしなければなりません。

使用上の必要条件の詳細は、「一覧」の表を参照してください。

プロパティ

AsyncMode プロパティ　R/W

形式　**BOOL AsyncMode;**

説明　TRUE：**DispenseCash**、**DispenseChange**メソッドは、非同期的に実行されます。
FALSE：これらのメソッドは同期的に実行されます。

このプロパティは、**Open**メソッドによりFALSEに初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

参照　**DispenseCash**メソッド、**DispenseChange**メソッド、**AsyncResultCode**プロパティ、**AsyncResultCodeExtended**プロパティ

AsyncResultCode プロパティ

形式　**LONG AsyncResultCode;**

説明　最後の非同期の払い出しの処理が完了した時（**DispenseCash**メソッドあるいは**DispenseChange**メソッドが**AsyncMode**=TRUEで呼び出された時）に終了ステータスを返します。このプロパティは、CHAN_STATUS_ASYNCの*Status*値の**StatusUpdateEvent**イベントで通知する前に、コントロールにより設定されます。

このプロパティの値は、メソッドが同期的に実行された際の**ResultCode**プロパティの値と同じです。

参照　**DispenseCash**メソッド、**DispenseChange**メソッド、**AsyncMode**プロパティ

AsyncResultCodeExtended プロパティ

形式　　LONG AsyncResultCodeExtended;

説明　　最後の非同期の払い出しの処理が完了した時（DispenseCashメソッドあるいはDispenseChangeメソッドがAsyncMode=TRUEで呼び出された時）に終了ステータスを返します。このプロパティは、CHAN_STATUS_ASYNCの*Status*値のStatusUpdateEventイベントで通知する前に、コントロールにより設定されます。このプロパティの値は、メソッドが同期的に実行された際のResultCodeExtendedプロパティの値と同じです。

参照　　DispenseCashメソッド、DispenseChangeメソッド、AsyncModeプロパティ

CapDeposit プロパティ　　第1．5版で追加

形式　　BOOL CapDeposit;

説明　　TRUE:自動釣り銭機は入金機能をサポートします。

FALSE:サポートしていません。

このプロパティは、Openメソッドにより初期化されます。

参照　　BiginDepositメソッド,EndDepositメソッド,FixDepositメソッド,PauseDepositメソッド

CapDepositDataEvent プロパティ　　第1．5版で追加

形式　　BOOL CapDepositDataEvent;

説明　　TRUE：自動釣り銭機は、入金が発生したときに入金イベントの報告を行う機能があります。
FALSE：報告する機能はありません。

このプロパティは、Openメソッドにより初期化されます。

参照　　BiginDepositメソッド,EndDepositメソッド,FixDepositメソッド,PauseDepositメソッド

CapDiscrepancy プロパティ

形式　　BOOL CapDiscrepancy;

説明　　TRUE：**ReadCashCounts**メソッドは*pDiscrepancy*に有効な値を設定します。
FALSE：サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　　**ReadCashCounts**メソッド

CapEmptySensor プロパティ

形式　　BOOL CapEmptySensor;

説明　　TRUE：自動釣り銭機は、エンプティになっている現金スロットがあることを報告します。
FALSE：サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　　**DeviceStatus**プロパティ、**StatusUpdateEvent**イベント

CapFullSensor プロパティ

形式　　BOOL CapFullSensor;

説明　　TRUE：自動釣り銭機は、フルになっている現金スロットがあることを報告します。
FALSE：サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　　**FullStatus**プロパティ、**StatusUpdateEvent**イベント

CapJamSensor プロパティ

形式　　BOOL CapJamSensor;

説明　　TRUE：コインディスペンサは硬貨詰まりや機器の異常を通知します。

FALSE：通知しません。

このプロパティはOpenメソッドにより初期化されます。

CapNearEmptySensor プロパティ

形式　　BOOL CapNearEmptySensor;

説明　　TRUE：自動釣り銭機は、ニアエンプティになっている現金スロットがあることを報告します。
FALSE：サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　　DeviceStatusプロパティ、StatusUpdateEventイベント

CapNearFullSensor プロパティ

形式　　BOOL CapNearFullSensor;

説明　　TRUE：自動釣り銭機は、ニアフルになっている現金スロットがあることを報告します。
FALSE：サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　　FullStatusプロパティ、StatusUpdateEventイベント

CapPauseDeposit プロパティ　　第1．5版で追加

形式　　BOOL CapPauseDeposit;

説明　　TRUE：自動釣り銭機は、入金機構を一時停止する機能があります。
FALSE：一時停止する機能はありません。

このプロパティは、Openメソッドにより初期化されます。

参照　　PauseDepositメソッド

CapRealTimeData プロパティ　　第1．11版で追加

形式　　BOOL CapRealTimeData;

説明　　TRUE：自動釣り銭機は、（“リアルタイムで”）入金データを報告できます。
FALSE：報告できません。

このプロパティは、Openメソッドにより初期化されます。

参照　　RealTimeDataEnabledプロパティ

CapRepayDeposit プロパティ　　第1．5版で追加

形式　　BOOL CapRepayDeposit;

説明　　TRUE：自動釣り銭機は、紙幣と硬貨を挿入口または払い出し口に返却する機能があります。
FALSE：返却する機能はありません。

このプロパティは、Openメソッドにより初期化されます。

参照　　EndDepositメソッド

CurrencyCashList プロパティ

形式　　BSTR CurrencyCashList;

説明　　CurrencyCodeプロパティで示される通貨での、自動釣り銭機がサポートする金種を表す文字列です。

文字列は、自動釣り銭機で使用できる硬貨の金種を示すコンマで区切られたASCII数字で始まり、ASCIIセミコロン(“;”)から、紙幣の金種を示すコンマで区切られたASCII数字が続きます。セミコロン(“;”)が無ければ、全ての金種は硬貨を表します。

下記は、日本での、CurrencyCashListの値の例です。

- “1,5,10,50,100,500” —
  1、5、10、50、100、500円硬貨。
- “1,5,10,50,100,500;1000,5000,10000” —
  1、5、10、50、100、500円硬貨と、1000、5000、10000円紙幣。
- “;1000,5000,10000” —
  1000、5000、10000円紙幣。

このプロパティはOpenメソッドにより初期化され、CurrencyCodeが設定された時に更新されます。

参照　　CurrencyCodeプロパティ

CurrencyCode プロパティ　R/W

形式　BSTR CurrencyCode;

説明　自動釣り銭機の操作で現在使用されている通貨コードが設定されます。

このプロパティはOpenメソッドで適当な値に初期化されます。有効な値は、CurrencyCodeListプロパティに記される通貨の中の一つです。

戻り値　本プロパティ設定時、次の値のいずれかがResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	このプロパティは正常に設定されました。
OPOS_E_ILLEGAL	CurrencyCodeList以外の値が指定されました。

参照　CurrencyCodeListプロパティ

CurrencyCodeList プロパティ

形式　BSTR CurrencyCodeList;

説明　通貨コードの文字列を表します。

文字列は、コンマで区切られたASCII 3 文字のISO 4217通貨コードにより構成されます。

このプロパティはOpenメソッドで初期化されます。

例えば、文字列が”JPY,USD”なら、自動釣り銭機は日本とU.S.の通貨単位をサポートします。

参照　CurrencyCodeプロパティ

CurrentExit プロパティ　R/W

形式　LONG CurrentExit;

説明　現在の払い出し口を示します。値が1なら基本の（もしくは、通常の）払い出し口を示し、1より大きい値ならその他の払い出し口を示します。適切な値は1から**DeviceExits**までの間です。

このプロパティは**Open**メソッドで1に初期化されます。

下の例は、日本での典型的なプロパティ値集合のサンプルです。**CurrencyCode**は“JPY”で、**CurrencyCodeList**は“JPY”です。

- 自動釣り銭機は硬貨をサポート ； 一つのみの払い出し口 ：
  CurrencyCashList = “1,5,10,50,100,500”
  DeviceExits = 1
  CurrentExit = 1：**ExitCashList** = “1,5,10,50,100,500”
- 自動釣り銭機は硬貨と紙幣の両方をサポート ； その他払い出し口は大量の紙幣払い出しに使用 ：
  CurrencyCashList = “1,5,10,50,100,500;1000,5000,10000”
  DeviceExits = 2
  CurrentExit = 1：**ExitCashList** = “1,5,10,50,100,500;1000,5000”
  CurrentExit = 2：**ExitCashList** = “;1000,5000,10000”
- 自動釣り銭機は紙幣をサポート ； その他払い出し口は大量の紙幣払い出しに使用 ：
  CurrencyCashList = “;1000,5000,10000”
  DeviceExits = 2
  CurrentExit = 1：**ExitCashList** = “;1000,5000”
  CurrentExit = 2：**ExitCashList** = “;1000,5000,10000”

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	このプロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な**CurrentExit**の値が指定されました。

参照　**CurrencyCashListプロパティ**、**DeviceExitsプロパティ**、**ExitCashListプロパティ**

CurrentService プロパティ　R/W　第1．11版で追加

形式　LONG CurrentService;

説明　現在のサービスを示します。値が0なら基本サービスを示し、0より大きくServiceCount以下なら統合サービスの情報要求に使用されます。有効な値は0からServiceCountまでです。ReadCashCountsメソッドやすべての共通・専用プロパティは、CurrentServiceが0以上でも利用できます。CurrentService、ServiceCount、ServiceIndexは常に基本サービスの状態を反映します。

下の例は、硬貨入金機とコインディスペンサと、唯一の紙幣自動釣り銭機から構成されるサービスを、自動釣り銭機サービスから利用した場合です。アプリケーションはStatusUpdateEventでジャムを検出しています。紙幣自動釣り銭機とコインディスペンサだけがジャムを検出できます。

- 基本サービスの値をチェック：
  CurrentService = 0
  ServiceCount = 3
  ServiceIndex = 50528769 (X'03030201')
  DeviceStatus = CHAN_STATUS_JAM
  DeviceServiceDecription = "Integrated Cash Changer Service 1.11.05"
- コインディスペンサの情報を取得するためにサービスを変更：
  CurrentService = 2
  ServiceCount = 3
  ServiceIndex = 50528769 (X'03030201')
  DeviceStatus = CHAN_STATUS_OK
  DeviceServiceDecription = "Pennybrite Coin Dispenser Service"
- コインディスペンサはOKのようだ。紙幣自動釣り銭機は？：
  CurrentService = 3
  ServiceCount = 3
  ServiceIndex = 50528769 (X'03030201')
  DeviceStatus = CHAN_STATUS_JAM
  DeviceServiceDecription = "Benjamin Bill Changer Service"

このプロパティはOpenメソッドで0に初期化されます。

戻り値　本プロパティ設定時、次の値のいずれかがResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	このプロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効なCurrentServiceの値が指定されました。

参照　ServiceCountプロパティ、ServiceIndexプロパティ

DepositAmount プロパティ　第1．5版で追加

形式　LONG DepositAmount;

説明　入金された合計金額を表します。

例えば、CurrencyCodeプロパティが日本円（JPY）で、DepositAmountプロパティで表される値が、
18057
なら、18,057円がBeginDepositメソッドの呼び出し以降の入金された現金金額となります。

参照　CurrencyCodeプロパティ

DepositCashList プロパティ　第1．5版で追加

形式　BSTR DepositCashList;

説明　CurrencyCodeプロパティで示される通貨での、自動釣り銭機がサポートする金種を表す文字列です。入金機構がサポートされていない場合は空の文字列になります。

文字列は、自動釣り銭機で使用できる硬貨の金種を示すコンマで区切られたASCII数字で始まり、ASCIIセミコロン(“;”)から、紙幣の金種を示すコンマで区切られたASCII数字が続きます。セミコロン(“;”)が無ければ、全ての金種は硬貨を表します。

下記は、日本での、DepositCashListの値の例です。

- “1,5,10,50,100,500” —
  1、5、10、50、100、500円硬貨。
- “1,5,10,50,100,500;1000,5000,10000” —
  1、5、10、50、100、500円硬貨と、1000、5000、10000円紙幣。
- “;1000,5000,10000” —
  1000、5000、10000円紙幣。

このプロパティはOpenメソッドにより初期化され、CurrencyCodeが設定された時に更新されます。

参照　CurrencyCodeプロパティ

DepositCodeList プロパティ　　第1．5版で追加

形式　　BSTR DepositCodeList;

説明　　入金の通貨コードの文字列を表します。入金機構がサポートされていない場合は空の文字列になります。

文字列は、コンマで区切られたASCII 3 文字のISO 4217通貨コードにより構成されます。

例えば、文字列が”JPY,USD”なら、自動釣り銭機は日本とU.S.の通貨単位をサポートします。

このプロパティはOpenメソッドで初期化されます。

参照　　CurrencyCodeプロパティ

DepositCounts プロパティ　　第1．5版で追加

形式　　BSTR DepositCounts;

説明　　入金された金種の文字列を表します。文字列のフォーマットは**DispenseCash**メソッド内の*CashCounts*と同じです。文字列内の金種は**DepositCashList**プロパティと同じで、同じ順序で並んでいます。入金機構がサポートされていない場合は空の文字列になります。

例えば、**CurrencyCode**プロパティが日本円（JPY）で、**DepositCounts**プロパティで表される文字列が、
1:80,5:77,10:0,50:54,100:0,500:87
なら、80枚の1円硬貨と、77枚の5円硬貨と、54枚の50円硬貨、87枚の500円硬貨が**BeginDeposit**メソッドの呼び出し以降の入金された現金枚数となります。

参照　　**CurrencyCode**プロパティ

DepositStatus プロパティ　　第1．5版で追加

形式　　LONG DepositStatus;

説明　　自動釣り銭機の入金機構の現在の状態を示します。次のいずれかとなります。

値	意味
CHAN_STATUS_DEPOSIT_START	入金可能状態です。
CHAN_STATUS_DEPOSIT_END	入金停止状態です。
CHAN_STATUS_DEPOSIT_NONE	入金機能がありません。
CHAN_STATUS_DEPOSIT_COUNT	入金計数中または返却中です。
CHAN_STATUS_DEPOSIT_JAM	機器障害が発生しました。

このプロパティはイネーブルの設定で初期化され、イネーブルの間は最新の状態になります。このプロパティはCHAN_STATUS_DEPOSIT_ENDに初期化されます。デバイスが入金をサポートしなければ、CHAN_STATUS_DEPOSIT_NONEに初期化されます。

DeviceExits プロパティ

形式　　LONG DeviceExits;

説明　　現金の払い出し口の個数が設定されます。

このプロパティは**Open**メソッドで初期化されます。

参照　　CurrentExitプロパティ

DeviceStatus プロパティ

形式　　LONG DeviceStatus;

説明　　自動釣り銭機の現在の状態を示します。次のいずれかとなります。

値	意味
CHAN_STATUS_OK	自動釣り銭機の現在の状態は正常です。
CHAN_STATUS_EMPTY	エンプティの現金スロットが有ります。
CHAN_STATUS_NEAREMPTY	ニアエンプティの現金スロットが有ります。
CHAN_STATUS_JAM	機器障害が発生しました。

このプロパティは初期化された後、デバイスがイネーブルの間は最新の状態となります。複数の状態が発生した時、優先順位は障害、エンプティ、ニアエンプティの順となります。

ExitCashList プロパティ

形式　　BSTR ExitCashList;

説明　　CurrentExitプロパティで示された払い出し口へ払い出せる金種の文字列です。サポートされる金種は、CurrencyCashListと同じかその一部です。文字列のフォーマットはCurrencyCashListと同じです。

このプロパティはOpenメソッドで初期化され、CurrencyCodeまたはCurrentExitが設定された時に更新されます。

参照　　CurrencyCodeプロパティ、CurrencyCashListプロパティ、CurrentExitプロパティ

FullStatus プロパティ

形式　　LONG FullStatus;

説明　　現金スロットの現在のフル状態を示します。次のいずれかとなります。

値	意味
CHAN_STATUS_OK	ニアフル、フルの現金スロットは有りません。
CHAN_STATUS_FULL	フルの現金スロットが有ります。
CHAN_STATUS_NEARFULL	ニアフルの現金スロットが有ります。

このプロパティは初期化された後、デバイスがイネーブルの間は最新の状態となります。

RealTimeDataEnabled プロパティ　R/W　　第1．11版で追加

形式　　BOOL RealTimeDataEnabled;

説明　　TRUEでCapRealTimeDataがTRUEなら、データイベントの度にDepositAmountとDepositCountsプロパティを更新します。FALSEなら、FixDepositが呼び出される時に金額が収集されてDepositAmountとDepositCountsが更新されます。RealTimeDataEnabledを設定しても、継続してBeginDepositが実行されなければ、システムの動作に影響を与えません。このプロパティにより、BeginDepositとEndDepositの間の状態変化に対する混乱を防ぎます。

このプロパティはOpenメソッドで初期化されます。

戻り値　　本プロパティ設定時、次の値のいずれかがResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	このプロパティは正常に設定されました。
OPOS_E_ILLEGAL	CapRealTimeDataがFALSEのため設定できません。

参照　　CapRealTimeDataプロパティ、DepositAmountプロパティ、DepositCountsプロパティ、BeginDepositメソッド、EndDepositメソッド、FixDepositメソッド

ServiceCount プロパティ　　第1．11版で追加

形式　　LONG ServiceCount;

説明　　自動釣り銭機サービスで使用される統合サービスの個数です。このサービスが他のサービスを使用しない場合、この値は0になります。

このプロパティはOpenメソッドで初期化されます。

参照　　CurrentServiceプロパティ、ServiceIndexプロパティ

ServiceIndex プロパティ　　第1．11版で追加

形式　　LONG ServiceIndex;

説明　　この値は4つのバイトに分割され、それぞれの統合サービスタイプのサービスインデックスを示します。下表はプロパティがどのように分割されるかを示します： 値が0なら統合サービスは利用されていません。

High Word		Low Word	
Hight Byte 紙幣ディスペンサ	Low Byte 紙幣入金機	Hight Byte コインディスペンサ	Low Byte 硬貨入金機

このプロパティはOpenメソッドで初期化されます。

参照　　CurrentServiceプロパティ、ServiceCountプロパティ

メソッド

AdjustCashCounts メソッド　　第1.11版で追加

形式　LONG AdjustCashCounts (BSTR *CashCounts*);

パラメータ	説明
CashCount	*CashCount*パラメータは、初期化する現金の金種と枚数を指定します。

説明　最初の設置後にコンディスペンサの開始現金有高を設定したり、入出金操作などの補充・回収後の現金有高の補正を行うため、このメソッドは呼び出されます。コインディスペンサ内の正確な現金合計を自動的に判定できないデバイスでは、このメソッドの呼び出しが必要です。デバイスが正確な合計額を判定できるなら、このメソッドの呼び出しは無視されます。アプリケーションは最初に**ReadCashCounts**を呼び出して現在の有高を取得し、実際の有高に補正します。さらに、アプリケーションはこのメソッドを呼び出し、ディスペンサ内の現在の有高を設定するでしょう。

すべての現金有高をリセットするには、各々の金種をゼロに設定します。

例えば、通貨が日本円で*CashCounts*パラメータに“1:80,5:77,50:54,100:0,500:87”を設定して**AdjustCashCounts**メソッドを呼び出せば、0枚の1円と、77枚の5円と、54枚の50円と、0枚の100円と、87枚の500円の硬貨がコインディスペンサに有ります。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは成功しました。
その他	**ResultCode**の項目を参照してください。

参照　**ReadCashCounts**メソッド

BeginDeposit メソッド　　第1．5版で追加

形式　LONG BeginDeposit ();

説明　自動釣り銭機の入金処理を開始します。

このメソッドの呼び出しで以下のプロパティ値は初期化されます。

DepositCountsプロパティは各金種の枚数値が0にセットされます。

DepositAmountプロパティは0がセットされます。

CapDepositDataEventプロパティ値がTRUEなら、このメソッドの呼び出し後に自動釣り銭機が現金を受け付けるたびに**DataEvent**イベントにて入金イベントが報告されます。但し、入金処理が一時停止状態の場合は入金イベントは報告されません。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	自動釣り銭機は入金機能をサポートしていません。または、呼び出しシーケンスが正しくありません。
その他	**ResultCode**の項目を参照してください。

参照　**DepositCounts**プロパティ、**DepositAmount**プロパティ、**CapDepositDataEvent**プロパティ、**EndDeposit**メソッド,**FixDeposit**メソッド、**PauseDeposit**メソッド

DispenseCash メソッド

形式　LONG DispenseCash (BSTR *CashCounts*);

パラメータ	意味
CashCount	金種:枚数, ..;..金種:枚数のフォーマットで表された払い出す金種と枚数です。";"の前は硬貨を表し、";"の後ろは紙幣を表します。";"が無ければ、全ての金種は硬貨を表します。

説明　**CurrentExit**で指定された払い出し口に、自動釣り銭機から現金を払い出します。払い出す現金は金種と枚数のペアで指定されます。

AsyncModeがFALSEならこのメソッドは同期的に実行され、**AsyncMode**がTRUEなら非同期的に実行されます。

通貨として日本円を使用した際の、*CashCounts*の例は下記の通りです。

- "10:5,50:1,100:3,500:1"
  5枚の10円硬貨と、1枚の50円硬貨と、3枚の100円硬貨と、1枚の500円硬貨を払い出します。
- "10:5,100:3;1000:10"
  5枚の10円硬貨と、3枚の100円硬貨と、10枚の1000円紙幣を払い出します。
- ";1000:10,10000:5"
  10枚の1000円紙幣と、5枚の10000円紙幣を払い出します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	指定の現金は正常に払い出されました、または**DispenseCash**メソッドは非同期的に実行されました。
OPOS_E_BUSY	非同期メソッドが実行中のため、現金は払い出せません。
OPOS_E_ILLEGAL	*CashCounts*パラメータの値は、現在の払い出し口に対して不正です。または、入金動作が実行中のため現金は払い出せません
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_ECHAN_OVERDISPENSE： 現金不足で、指定の現金を払い出せません。

その他　　ResultCodeの項目を参照してください。

参照　　AsyncModeプロパティ、CurrentExitプロパティ

DispenseChange メソッド

形式　LONG DispenseChange (LONG *Amount*);

パラメータ	意味
Amount	払い出す釣り銭の合計です。利用できる現金から、どの硬貨、紙幣の組み合わせで払い出し要求にこたえるかは、自動釣り銭機の責任となります。

説明　**CurrentExit**で指定された払い出し口に、自動釣り銭機から指定合計額の現金を払い出します。

AsyncModeがFALSEならこのメソッドは同期的に実行され、**AsyncMode**がTRUEなら非同期的に実行されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	指定の釣り銭は正常に払い出されました、または**DispenseChange**メソッドは非同期的に実行されました。
OPOS_E_BUSY	非同期メソッドが実行中のため、指定の釣り銭は払い出せません。
OPOS_E_ILLEGAL	下記エラーが発生しました。
	*Amount*がゼロか負です。
	現在の払い出し口の**ExitCashList**の値では*Amount*を払い出せません。
	入金動作が実行中のため指定の釣り銭が払い出せません。
	OPOS_E_EXTENDED**ResultCodeExtended** = OPOS_ECHAN_OVERDISPENSE：現金不足で、指定の釣り銭を払い出せません。
その他	**ResultCode**の項目を参照してください。

参照　**AsyncModeプロパティ**、**CurrentExitプロパティ**

EndDeposit メソッド　　第1．5版で追加

形式　　LONG EndDeposit (LONG *Success*);

*Success*パラメータには、入金した現金の処理を設定します。この値は、以下のいずれかの値です。

値	意味
CHAN_DEPOSIT_CHANGE	入金した現金を収納。釣り銭あり。
CHAN_DEPOSIT_NOCHANGE	入金した現金を収納。釣り銭なし。
CHAN_DEPOSIT_REPAY	現金投入口または現金払出し口に返却。

説明　　自動釣り銭機の入金処理を終了します。

このメソッドを呼び出す前に、アプリケーションは釣り銭額の計算をする必要があります。入金合計＞販売合計なら、CHAN_DEPOSIT_CHANGEを*Success*パラメータとしてこのメソッドを呼び出します。入金合計≦販売合計なら、CHAN_DEPOSIT_NOCHANGEを*Success*パラメータとしてこのメソッドを呼び出します。
もし*Success*パラメータがCHAN_DEPOSIT_REPAYなら、入金した現金を格納せずに現金投入口または現金払い出し口に返却します。

このメソッドを呼び出す前には必ず**FixDeposit**メソッドにて入金の確定を行ってください。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	次に示すいずれかのエラーが発生しています。 ・自動釣り銭機は入金機能をサポートしていません。 ・呼び出しシーケンスが正しくありません。このメソッドを呼び出す前に**BeginDeposit**メソッドおよび**FixDeposit**メソッドを呼び出す必要があります。
その他	**ResultCode**の項目を参照してください。

参照　DepositCountsプロパティ、DepositAmountプロパティ、CapDepositDataEventプロパティ、BeginDepositメソッド、FixDepositメソッド、PauseDepositメソッド

FixDeposit メソッド　第1．5版で追加

形式　LONG FixDeposit ();

説明　自動釣り銭機の入金確定を行います。このとき、自動釣り銭機は入金処理を停止し、DepositCountsプロパティとDepositAmountプロパティの値が確定します。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	次に示すいずれかのエラーが発生しています。 ・自動釣り銭機は入金機能をサポートしていません。 ・呼び出しシーケンスが正しくありません。このメソッドを呼び出す前にBeginDepositメソッドを呼び出す必要があります。
その他	ResultCodeの項目を参照してください。

参照　DepositCountsプロパティ、DepositAmountプロパティ、BeginDepositメソッド、EndDepositメソッド、PauseDepositメソッド

PauseDeposit メソッド　　第1．5版で追加

形式　　LONG PauseDeposit (LONG *Control*);

このメソッドは入金処理の一時停止を行うか、入金処理を再開するときに使います。*Control*パラメータの値は、以下のいずれかの値です。

値	意味
CHAN_DEPOSIT_PAUSE	入金の一時停止。
CHAN_DEPOSIT_RESTART	入金の再開。

説明　　入金処理を一時停止または再開します。

*Control*がCHAN_DEPOSIT_PAUSEなら、入金処理を一時停止します。このメソッドがCHAN_DEPOSIT_RESTARTで呼び出されるまで、入金処理は一時停止状態になります。また、入金処理が一時停止の状態で、**FixDeposit**メソッドを呼び出すことが出来ます。

入金処理が一時停止するとき、**DepositCounts**プロパティと**DepositAmount**プロパティは自動釣り銭機の現在の状態に更新されます。入金処理が再開されるまで、上記プロパティ値は更新されません。

*Control*がCHAN_DEPOSIT_RESTARTなら、入金処理は再開されます。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	次に示すいずれかのエラーが発生しています。
	自動釣り銭機は入金機能をサポートしていません。
	呼び出しシーケンスが正しくありません。このメソッドを呼び出す前に**BeginDeposit**メソッドを呼び出す必要があります。
	すでに一時停止している状態で、CHAN_DEPOSIT_PAUSEが発行されたか、一時停止していない状態で、CHAN_DEPOSIT_RESTARTが発行された。
その他	**ResultCode**の項目を参照してください。

参照　DepositCountsプロパティ、DepositAmountプロパティ、CapDepositDataEventプロパティ、CapPauseDepositプロパティ、BeginDepositメソッド、EndDepositメソッド、FixDepositメソッド

ReadCashCounts メソッド

形式　LONG ReadCashCounts (BSTR* *pCashCounts*, BOOL* *pDiscrepancy*);

パラメータ	意味
pCashCounts	現金枚数データを格納するエリアを示します。
pDiscrepancy	TRUE： *pCashCounts*で報告された現金枚数に含まれない現金が有ります。 FALSE：現金有り高は*pCashCounts*の内容と一致します。

説明　*pCashCounts*で指定される文字列のフォーマットはDispenseCashメソッド内の*CashCounts*と同じです。*pCashCounts*内の金種はCurrencyCashListプロパティと同じで、同じ順序で並んでいます。

例えば、通貨が日本円で、ReadCashCountsメソッドの呼び出し結果として*pCashCounts*パラメータに返される文字列が、
　1:80,5:77,10:0,50:54,100:0,500:87
なら、80枚の1円硬貨と、77枚の5円硬貨と、54枚の50円硬貨、87枚の500円硬貨が自動釣り銭機に有ります。

CapDiscrepancyがFALSEなら、*pDiscrepancy*は常にFALSEです。

通常、*pCashCounts*パラメータから計算される現金有り高は、自動釣り銭機内の現金有り高に一致します。しかし、自動釣り銭機内に計数不可能な現金があるといった場合に、食い違いが発生する場合があります。例えば、現金の検出・監視の能力を超える現金スロットのオーバーフローなどがあります。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に終了しました。
OPOS_E_BUSY	非同期メソッド実行中のため、金種枚数は読み出せませんでした。
その他	ResultCodeの項目を参照してください。

参照　　CapDiscrepancyプロパティ、CurrencyCashListプロパティ、DispenseCashメソッド

イベント

DataEvent イベント　　第1．11版で修正

形式　　void DataEvent (LONG *Status*);

*Status*パラメータには、0が設定されます。

説明　　自動釣り銭機に入金された時に通知されます。

StatusUpdateEvent イベント

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

*Status*パラメータは、自動釣り銭機の状態が設定されます。

値	意味
CHAN_STATUS_EMPTY	エンプティの現金スロットが有ります。
CHAN_STATUS_NEAREMPTY	ニアエンプティの現金スロットが有ります。
CHAN_STATUS_EMPTYOK	エンプティ、ニアエンプティの状態が解除されました。
CHAN_STATUS_FULL	フルの現金スロットが有ります。
CHAN_STATUS_NEARFULL	ニアフルの現金スロットが有ります。
CHAN_STATUS_FULLOK	フル、ニアフルの状態が解除されました。
CHAN_STATUS_JAM	機器障害が発生しました。
CHAN_STATUS_JAMOK	機器障害は回復しました。
CHAN_STATUS_ASYNC	非同期メソッドの実行が終了しました。
*電源通知の**StatusUpdateEvent**の値*	「第2章 共通プロパティ、メソッド、イベント」の**StatusUpdateEvent**の記述を参照。

説明　　自動釣り銭機の状態が変化した時に通知されます。

デバイスがフル、ニアフル、エンプティ、ニアエンプティ状態の検出センサを持ち、これらの状態の機能プロパティが設定されている時にのみ、自動釣り銭機は釣り銭の充填レベルを**StatusUpdateEvent**で通知できます。

障害が発生したなら、障害状態が通知されます。これは非同期メソッド実行の終了と同じです。

非同期メソッド実行の終了ステータスは、**AsyncResultCode**と**AsyncResultCodeExtended**プロパティに設定されます。

第　16 章
トーンインジケータ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.2	Boolean	R/W	*適用外*
BinaryConversion	1.2	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.2	String	R	Open
Claimed	1.2	Boolean	R	Open
DataCount	1.2	Long	R	*適用外*
DataEventEnabled	1.2	Boolean	R/W	*適用外*
DeviceEnabled	1.2	Boolean	R/W	Open
FreezeEvents	1.2	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.2	Long	R	Open
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.2	Long	R	無し
ResultCodeExtended	1.2	Long	R	Open
State	1.2	Long	R	無し
ControlObjectDescription	1.2	String	R	無し
ControlObjectVersion	1.2	Long	R	無し
ServiceObjectDescription	1.2	String	R	Open
ServiceObjectVersion	1.2	Long	R	Open
DeviceDescription	1.2	String	R	Open
DeviceName	1.2	String	R	Open
専用				
AsyncMode	1.2	Boolean	R/W	Open & Enable
CapPitch	1.2	Boolean	R	Open

CapVolume	1.2	Boolean	R	Open
CapMelody	1.13	Long	R	Open
Tone1Pitch	1.2	Long	R/W	Open & Enable
Tone1Volume	1.2	Long	R/W	Open & Enable
Tone1Duration	1.2	Long	R/W	Open & Enable
Tone2Pitch	1.2	Long	R/W	Open & Enable
Tone2Volume	1.2	Long	R/W	Open & Enable
Tone2Duration	1.2	Long	R/W	Open & Enable
InterToneWait	1.2	Long	R/W	Open & Enable
MelodyType	1.13	Long	R/W	Open & Enable
MelodyVolume	1.13	Long	R/W	Open & Enable

メソッド

共通	*版数*	*使用可能条件*
Open	1.2	無し
Close	1.2	Open
ClaimDevice	1.2	Open
ReleaseDevice	1.2	Open & Claim
CheckHealth	1.2	Open & Enable (注)
ClearInput	1.2	*適用外*
ClearInputProperties	1.10	*適用外*
ClearOutput	1.2	Open
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.2	Open
ResetStatistics	1.8	Open & Enable (注)
RetrieveStatistics	1.8	Open & Enable (注)
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open & Enable (注)
専用		
Sound	1.2	Open & Enable (注)
SoundImmediate	1.2	Open & Enable (注)

（注） トーンインジケータを排他アクセスしているアプリケーションがないこと。

イベント

名称	*版数*	*発生条件*
DataEvent	1.2	*適用外*
DirectIOEvent	1.2	Open

ErrorEvent	1.2	Open & Enable
OutputCompleteEvent	1.2	Open & Enable
StatusUpdateEvent	1.2	Open & Enable

概説

トーンインジケータ・コントロールのOLEプログラマティクIDは“OPOS.ToneIndicator”です。

機能

トーンインジケータは次の機能を基本機能として実装しています。

PCシステムのスピーカー、その他のハードウェアのトーンデバイスを鳴らします。多くの場合PCのスピーカーは利用できないかオペレータが聞き取りやすい場所にない場合があります。

簡単な音程と音量の調整機能を持ち、２種類のトーンのインジケータを鳴らします。

Win32のビープ機能と同様に同期的に一度だけのインジケータ機能を提供します。

モデル　　*第1.13版で更新*

トーンインジケータデバイスは、POS用ハードウェアプラットフォームに外部PC標準スピーカ機能と同様な機能を提供する場合に使われます。多くのPOSシステムではそのようなデバイスを持っています。

このデバイスは、「サイレン」音、又は(且つ)「メロディ」音を生成するようなサウンドをサポートします。

- 「サイレン」音は２種類のサウンドをサポートします。
- 「メロディ」音は数種類のサウンドをサポートします。

インジケータは通常非同期に開始され、ユーザーが音を認識するのを待っている間に、アプリケーションが他の機能を使用できます。非同期に音を鳴らし続けたり、途中で止めたりするオプションもあります。

インジケータが非同期に開始されたときは、全ての音が鳴り終わった時に**OutputCompleteEvent**が発生します。これによってアプリケーションは音が止まったことを認識できます。たとえば、キャッシュドロワが開いているときに指定された回数だけ音を鳴らすことができます。アプリケーションはキャッシュドロワが閉まったら音を止めたり、もしそうでない場合に**OutputCompleteEvent**で戻ってきた場合には

オペレータに注意を喚起するために再度大きな音を鳴らすことができます。

トーンインジケータは一般的な出力モデルに沿っています。非同期出力は次のように処理されます。

- トーンインジケータコントロールはその要求をバッファリングし、**OutputID**プロパティをこの要求のための識別子として設定し、できるかぎり早く制御を戻します。デバイスがその要求を正常に完了した時、トーンインジケータコントロールは**OutputCompleteEvent**を通知します。このイベントのパラメータは、完了した要求の**OutputID**を含みます。
- **Sound**メソッドはハードの問題によるエラーステータスは返しません。これらのエラーは**ErrorEvent**によってのみ通知されます。エラーステータスはトーンインジケータコントロールが他のアプリケーションによって排他アクセスされていたり、イネーブルされていなかったり、パラメータが不正だったり、要求がキューに追加できなかった場合にのみ返されます。最初の二つのケースは、アプリケーションのエラーです。一方、最後のケースは重大なシステムリソースの例外です。
- 非同期の要求が実行されている間に、エラーが発生すると、**ErrorEvent**が通知されます。
- トーンインジケータコントロールは、非同期出力がFIFOで実行されることを保証しています。
- **ClearOutput**メソッドを呼ぶことによって、トーンインジケータコントロールによってバッファリングされた全ての出力は、削除されます。**OutputCompleteEvent**は、削除された出力に対しては通知されません。このメソッドは、（可能ならば）処理中の出力も中止します。

「サイレン」音と「メロディ」音の切り替えは、**MelodyType**プロパティで行います。**MelodyType**プロパティがTONE_MT_NONEの場合は「サイレン」音を鳴らします。**MelodyType**プロパティがTONE_MT_NONE以外の場合は「メロディ」音を鳴らします。

「メロディ」音を鳴らす場合は、**ToneXPitch**、**ToneXVolume**、**ToneXDuration**、**InterToneWait**の設定は無視されます。

例

非同期に２種類の音をセットし、それを100回鳴らします。各々の音は750ミリ秒の間隔で50%の音量で鳴らし、各々の音の間にはポーズはない設定です。

```
Indicator.Tone1Pitch = 500
Indicator.Tone1Volume = 50
Indicator.Tone1Duration = 750
Indicator.Tone2Pitch = 800
Indicator.Tone2Volume = 50
Indicator.Tone2Duration = 750
Indicator.InterToneWait = 0

Indicator.AsyncMode = True
Indicator.Sound 100, 0
```

同期的に音を鳴らします。単純なビープ音で、500ミリ秒音を鳴らして500ミリ秒音を消します。

```
Indicator.Tone1Pitch = 500
Indicator.Tone1Volume = 50
Indicator.Tone1Duration = 500
Indicator.Tone2Pitch = 0  ' turn off second tone
Indicator.InterToneWait = 0  ' no wait after tone-1

Indicator.AsyncMode = False
Indicator.Sound 100, 500
```

次の例では両方の音を０にしているためエラーとなります。

```
Indicator.Tone1Pitch = 0  ' turn off first tone
Indicator.Tone2Pitch = 0  ' turn off second tone

Indicator.Sound 100, 0
```

Soundメソッドは任意の期間音を鳴らし続けるためにも使用できます。*NumberOfCycles*パラメータがOPOS_FOREVERで指定されていた場合、音が鳴り続けます。停止は明確に実施する必要があります。

```
Indicator.Tone1Pitch = 500
Indicator.Tone1Volume = 50
Indicator.Tone1Duration = 500
Indicator.Tone2Pitch = 0  ' turn off second tone
Indicator.InterToneWait = 0  ' no wait after tone-1

Indicator.AsyncMode = True
Indicator.Sound OPOS_FOREVER, 500
```

ClearOutputまたは**SoundImmediate**メソッドを使用して鳴り続けている音を消します。

```
…
Indicator.AsyncMode = True
Indicator.Sound OPOS_FOREVER, 500
…
Indicator.ClearOutput
   or
Indicator.SoundImmediate
```

SoundImmediateメソッドは、また**InterToneWait**の間隔で２つの音を一度鳴らします。音は同期的に鳴ります。これはWin32 APIで提供されている通常のBeep関数と同等です。

```
Indicator.Tone1Pitch = 500
Indicator.Tone1Volume = 50
Indicator.Tone1Duration = 500
Indicator.Tone2Pitch = 0  ' turn off second tone
Indicator.InterToneWait = 0  ' no wait after tone-1

Indicator.AsyncMode = True
Indicator.Sound 1, 0   ' asynchronous beep
Indicator.SoundImmediate  ' synchronous beep
```

デバイスの共有

トーンインジケータは共有可能なデバイスです。デバイスを共有する際のルールは次のとおりです。

デバイスをオープンし、イネーブルにした後、アプリケーションはすべてのプロパティとメソッドを使用することができます。また、ステータスアップデートイベントを受け取ることができます。

複数のアプリケーションがデバイスをオープンし、イネーブルにした場合、そのすべてのアプリケーションはプロパティとメソッドを扱うことができます。ステータスアップデートイベントはそれらすべてのアプリケーションに通知されます。

あるアプリケーションがトーンインジケータの排他アクセス権を獲得した場合そのアプリケーションでのみSoundおよびSoundImmediateメソッドを呼び出すことができます。この機能を使用することは、起動時にデバイスの排他アクセス権を獲得したメインのアプリケーションのためにトーンインジケータを限定するのに効果的です。

非同期のサウンドを起動したアプリケーションだけがそれに対応するOutputComplete及びErrorEventイベントを受け取ります。

1つのアプリケーションがサウンドを実行している時に別のアプリケーションがデバイスの排他アクセス権を獲得しサウンドを実行した場合、最初のアプリケーションからのサウンドの実行は割り込まれます。最初のアプリケーションが同期のSoundメソッドを実行していたのであればOPOS_E_CLAIMEDが返されます。非同期のSoundメソッドを実行していたのであればエラーは通知されず、単に最初のサウンドの実行が中止されます。

使用上の必要条件については「一覧」を参照してください。

プロパティ

AsyncMode プロパティ　R/W

形式　　**BOOL AsyncMode;**

説明　　TRUE：**Sound**メソッドは非同期で動作します。
FALSE：**Sound**メソッドは同期で動作します。

このプロパティは、**Open**メソッド後にデバイスが最初にイネーブルされたとき、FALSEに初期化されます。（1.5版以前では、**Open**メソッドで初期化されるとここで記述されていました。1.5版では、他のデバイスとの整合性をとるため修正されました。）

戻り値　　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

CapMelody プロパティ　　*第1.13版で追加*

形式　　**LONG CapMelody;**

説明　　メロディ機能を示します。

このプロパティには、利用可能なメロディの数を示します。メロディ機能が無い場合は0が設定されます。

このプロパティは、**Open**メソッドにより初期化されます。

CapPitch プロパティ

形式　　**BOOL CapPitch;**

説明　　TRUE：ハードウェアで音程を変更する機能を持ちます。
FALSE：持ちません

このプロパティは、**Open**メソッドにより初期化されます。

CapVolume プロパティ

形式　　**BOOL CapVolume;**

説明　　TRUE：ハードウェアで音量を変更する機能を持ちます。
FALSE：持ちません。

このプロパティは、**Open**メソッドにより初期化されます。

InterToneWait プロパティ　R/W

形式　　**LONG InterToneWait;**

説明　　音１と音２の間の無音の間隔（ミリ秒）
音２の後に続く次の音１までの間隔は**Sound**メソッドの
*InterSoundWait*パラメータで設定します。

このプロパティは、**Open**メソッド後にデバイスが最初にイネーブルされたとき、０に初期化されます。（1.5版以前では、**Open**メソッドで初期化されるとここで記述されていました。1.5版では、他のデバイスとの整合性をとるため修正されました。）

戻り値　　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

MelodyType プロパティ　R/W　　*第1.13版で追加*

形式　　LONG MelodyType;

説明　　メロディ音を示します。

値	意味
TONE_MT_NONE	メロディ音を鳴らしません。(サイレン音を鳴らします。)
TONE_MT_TYPE1(=1)	TYPE1のメロディ音です。
TONE_MT_TYPE2(=2)	TYPE2のメロディ音です。
TONE_MT_TYPE3(=3)	TYPE3のメロディ音です。
TONE_MT_TYPE4(=4)	TYPE4のメロディ音です。
TONE_MT_TYPE5(=5)	TYPE5のメロディ音です。

デバイスが6種類以上のメロディ音をサポートしている場合は、6以上の値が指定できます。

このプロパティは、**Open**メソッド後にデバイスが最初にイネーブルされたときTONE_MT_NONEに初期化されます。

戻り値　　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

MelodyVolume プロパティ　R/W　　*第1.13版で追加*

形式　LONG MelodyVolume;

説明　メロディ音の音量をデバイスの能力のパーセント値で表します。0以下では無音、100以上で最大になります。

デバイスがユーザ定義の音量をサポートしていない場合には（**CapVolume**プロパティがFALSE）、0より大きい値が設定されていてもハードウェアに依存した音が鳴ります。

このプロパティは、**Open**メソッド後にデバイスが最初にイネーブルされたとき、100に初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

Tone1Duration プロパティ　R/W

形式　LONG Tone1Duration;

説明　音１の長さ（ミリ秒）０以下の値の場合には音は鳴りません

このプロパティは、**Open**メソッド後にデバイスが最初にイネーブルされたとき、０に初期化されます。（1.5版以前では、**Open**メソッドで初期化されるとここで記述されていました。1.5版では、他のデバイスとの整合性をとるため修正されました。）

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

Tone1Pitch プロパティ　R/W

形式　　LONG Tone1Pitch;

説明　　音１の周波数（ヘルツ）です。０以下の値の場合には音は鳴りません。

デバイスがユーザ定義の音程をサポートしていない場合には、（**CapPitch**プロパティがFALSE）設定値が０より大きい値が設定されていてもハードウェアに依存した音が鳴ります。

このプロパティは、**Open**メソッド後にデバイスが最初にイネーブルされたとき、０に初期化されます。（1.5版以前では、**Open**メソッドで初期化されるとここで記述されていました。1.5版では、他のデバイスとの整合性をとるため修正されました。）

戻り値　　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

Tone1Volume プロパティ　R/W

形式　　**LONG Tone1Volume;**

説明　　音１の音量をデバイスの能力のパーセント値で表します。０以下では無音、100以上で最大になります。

デバイスがユーザ定義の音量をサポートしていない場合には（**CapVolume**プロパティがFALSE）、０より大きい値が設定されていてもハードウェアに依存した音が鳴ります。

このプロパティは、**Open**メソッド後にデバイスが最初にイネーブルされたとき、100に初期化されます。（1.5版以前では、**Open**メソッドで初期化されるとここで記述されていました。1.5版では、他のデバイスとの整合性をとるため修正されました。）

戻り値　　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

Tone2Duration プロパティ　R/W

形式　　**LONG Tone2Duration;**

説明　　音２の長さ（ミリ秒）０以下の値の場合には音は鳴りません

このプロパティは、**Open**メソッド後にデバイスが最初にイネーブルされたとき、０に初期化されます。（1.5版以前では、**Open**メソッドで初期化されるとここで記述されていました。1.5版では、他のデバイスとの整合性をとるため修正されました。）

戻り値　　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

Tone2Pitch プロパティ　R/W

形式　　LONG Tone2Pitch;

説明　　音２の周波数（ヘルツ）です。０以下の値の場合には音は鳴りません。

デバイスがユーザ定義の音程をサポートしていない場合には、（**CapPitch**プロパティがFALSE）設定値が０より大きい値が設定されていてもハードウェアに依存した音が鳴ります。

このプロパティは、**Open**メソッド後にデバイスが最初にイネーブルされたとき、０に初期化されます。（1.5版以前では、**Open**メソッドで初期化されるとここで記述されていました。1.5版では、他のデバイスとの整合性をとるため修正されました。）

戻り値　　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

Tone2Volume プロパティ　R/W

形式　　LONG Tone2Volume;

説明　　音２の音量をデバイスの能力のパーセント値で表します。０以下では無音、100以上で最大になります。

デバイスがユーザ定義の音量をサポートしていない場合には（**CapVolume**プロパティがFALSE）、０より大きい値が設定されていてもハードウェアに依存した音が鳴ります。

このプロパティは、**Open**メソッド後にデバイスが最初にイネーブルされたとき、100に初期化されます。（1.5版以前では、**Open**メソッドで初期化されるとここで記述されていました。1.5版では、他のデバイスとの整合性をとるため修正されました。）

戻り値　　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

メソッド

Sound メソッド　　*第1.13版で更新*

形式　　LONG Sound (LONG *NumberOfCycles*, LONG *InterSoundWait*);

パラメータ	説明
NumberOfCycles	OPOS_FOREVER：音を鳴らし続けます。 その他：指定回数繰り返し鳴らします。
InterSoundWait	*NumberOfCycles*sが１でない時、音の繰り返しの前（再度音１を鳴らす前）に *InterSoundWait*　ミリ秒の間、音を止めます。

説明　　音を鳴らす、または非同期に音を鳴らします。

AsyncModeがFALSEの場合は同期動作し、**AsyncMode**がTRUEの場合は非同期で動作します。

インジケータのサイクルは下記の通りです：

「サイレン」音

Tone1Durationプロパティ　＋
InterToneWaitプロパティ　＋
Tone2Durationプロパティ　＋
*InterSoundWait*パラメータ（最後の音のサイクルを除く）

「メロディ」音

MelodyTypeプロパティ　＋
*InterSoundWait*パラメータ（最後の音のサイクルを除く）

トーンインジケータを非同期で鳴らしはじめた後、以下のメソッドのうち、１つで音を止めることができます。（音を鳴らしはじめる時に使用した*NumberOfCycles*sパラメータの値がOPOS_FOREVERの場合、アプリケーションは継続して鳴っている音を止めるためにこれらのうちの１つを指定する必要があります。）

・**ClearOutput**メソッド

・**SoundImmediate**メソッド

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	インジケータが鳴らされたか鳴り始めました
OPOS_E_CLAIMED	他のアプリケーションがデバイスの排他アクセス権を獲得しサウンドを実行したことにより、このサウンドの実行が割り込まれたことを示します。(**AsyncMode**プロパティがFALSEの場合にのみ返されます。)
OPOS_E_ILLEGAL	次のエラーのいずれかが発生 ・*NumberOfCycles*正の数、０、OPOS_FOREVER以外が指定されています。 ・**AsyncMode**プロパティがFALSEの時に*NumberOfCycles*にOPOS_FOREVERが指定されています。 ・*InterSoundWait*に負の数が指定されています。
その他	**ResultCode**の項目を参照してください。

SoundImmediate メソッド

形式　**LONG SoundImmediate ( );**

説明　ハードウェアの音を同期で１回鳴らします。音１と音２の間は**InterToneWait**の間隔で鳴らされます。

非同期出力が未終了状態の場合、（あたかも**ClearOutput**メソッドがよばれたように）音を鳴らす前に終了させられます。
SoundImmediateメソッドは、主として非同期に音が鳴っている間にエラーイベント処理等で音を鳴らすという例外的な利用を目的とします。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	インジケータが鳴らされたか鳴り始めました
その他	**ResultCode**の項目を参照してください。

第　17　章
フィスカルプリンタ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.3	Boolean	R/W	*適用外*
BinaryConversion	1.3	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.3	String	R	Open
Claimed	1.3	Boolean	R	Open
DataCount	1.3	Long	R	*適用外*
DataEventEnabled	1.3	Boolean	R/W	*適用外*
DeviceEnabled	1.3	Boolean	R/W	Open ＆ Claim
FreezeEvents	1.3	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.3	Long	R	Open
PowerState	1.3	Long	R	Open
PowerNotify	1.3	Long	R/W	Open
ResultCode	1.3	Long	R	--
ResultCodeExtended	1.3	Long	R	Open
State	1.3	Long	R	--
ControlObjectDescription	1.3	String	R	--
ControlObjectVersion	1.3	Long	R	--
ServiceObjectDescription	1.3	String	R	Open
ServiceObjectVersion	1.3	Long	R	Open
DeviceDescription	1.3	String	R	Open
DeviceName	1.3	String	R	Open
専用				
CapAdditionalHeader	1.6	Boolean	R	Open
CapAdditionalLines	1.3	Boolean	R	Open

CapAdditionalTrailer	1.6	Boolean	R	Open
CapAmountAdjustment	1.3	Boolean	R	Open
CapAmountNotPaid	1.3	Boolean	R	廃止予定（第1.11版より）
CapChangeDue	1.6	Boolean	R	Open
CapCheckTotal	1.3	Boolean	R	Open
CapCoverSensor(2)	1.3	Boolean	R	Open
CapDoubleWidth	1.3	Boolean	R	Open
CapDuplicateReceipt	1.3	Boolean	R	Open
CapEmptyReceiptIsVoidable	1.6	Boolean	R	Open
CapFiscalReceiptStation	1.6	Boolean	R	Open
CapFiscalReceiptType	1.6	Boolean	R	Open
CapFixedOutput	1.3	Boolean	R	Open
CapHasVatTable	1.3	Boolean	R	Open
CapIndependentHeader	1.3	Boolean	R	Open
CapItemList	1.3	Boolean	R	Open
CapJrnEmptySensor(2)	1.3	Boolean	R	Open
CapJrnNearEndSensor(2)	1.3	Boolean	R	Open
CapJrnPresent(2)	1.3	Boolean	R	Open
CapMultiContractor	1.6	Boolean	R	Open
CapNonFiscalMode	1.3	Boolean	R	Open
CapOnlyVoidLastItem	1.6	Boolean	R	Open
CapOrderAdjustmentFirst	1.3	Boolean	R	Open
CapPackageAdjustment	1.6	Boolean	R	Open
CapPercentAdjustment	1.3	Boolean	R	Open
CapPositiveAdjustment	1.3	Boolean	R	Open
CapPostPreLine	1.6	Boolean	R	Open
CapPowerLossReport	1.3	Boolean	R	Open
CapPredefinedPaymentLines	1.3	Boolean	R	Open
CapReceiptNotPaid	1.3	Boolean	R	Open
CapRecEmptySensor(2)	1.3	Boolean	R	Open
CapRecNearEndSensor(2)	1.3	Boolean	R	Open
CapRecPresent(2)	1.3	Boolean	R	Open
CapRemainingFiscalMemory	1.3	Boolean	R	Open
CapReservedWord	1.3	Boolean	R	Open
CapSetHeader	1.3	Boolean	R	Open
CapSetPOSID	1.3	Boolean	R	Open
CapSetCurrency	1.6	Boolean	R	Open
CapSetStoreFiscalID	1.3	Boolean	R	Open
CapSetTrailer	1.3	Boolean	R	Open

CapSetVatTable	1.3	Boolean	R	Open
CapSlpEmptySensor(2)	1.3	Boolean	R	Open
CapSlpFiscalDocument	1.3	Boolean	R	Open
CapSlpFullSlip(2)	1.3	Boolean	R	Open
CapSlpNearEndSensor(2)	1.3	Boolean	R	Open
CapSlpPresent(2)	1.3	Boolean	R	Open
CapSlpValidation	1.3	Boolean	R	Open
CapSubAmountAdjustment	1.3	Boolean	R	Open
CapSubPercentAdjustment	1.3	Boolean	R	Open
CapSubtotal	1.3	Boolean	R	Open
CapTotalizerType	1.6	Boolean	R	Open
CapTrainingMode	1.3	Boolean	R	Open
CapValidateJournal	1.3	Boolean	R	Open
CapXReport	1.3	Boolean	R	Open
ActualCurrency	1.6	Long	R	Open, Claim, & Enable
AdditionalHeader	1.6	String	R/W	Open, Claim, & Enable
AdditionalTrailer	1.6	String	R/W	Open, Claim, & Enable
AmountDecimalPlaces	1.3	Long	R	Open, Claim, & Enable
AsyncMode	1.3	Boolean	R/W	Open
ChangeDue	1.6	String	R/W	Open
CheckTotal	1.3	Boolean	R/W	Open
ContractorId	1.6	Long	R/W	Open, Claim, & Enable
CountryCode	1.3	Long	R	Open, Claim, & Enable
CoverOpen(2)	1.3	Boolean	R	Open, Claim, & Enable
DateType	1.6	Long	R/W	Open, Claim, & Enable
DayOpened	1.3	Boolean	R	Open, Claim, & Enable
DescriptionLength	1.3	Long	R	Open
DuplicateReceipt	1.3	Boolean	R/W	Open
ErrorLevel	1.3	Long	R	Open
ErrorOutID	1.3	Long	R	Open, Claim, & Enable
ErrorState	1.3	Long	R	Open
ErrorStation	1.3	Long	R	Open
ErrorString	1.3	String	R	Open
FiscalReceiptStation	1.6	Long	R/W	Open, Claim, & Enable
FiscalReceiptType	1.6	Long	R/W	Open, Claim, & Enable
FlagWhenIdle	1.3	Boolean	R/W	Open
JrnEmpty(2)	1.3	Boolean	R	Open, Claim, & Enable
JrnNearEnd(2)	1.3	Boolean	R	Open, Claim, & Enable
MessageLength	1.3	Long	R	Open
MessageType	1.6	Long	R/W	Open

NumHeaderLines	1.3	Long	R	Open
NumTrailerLines	1.3	Long	R	Open
NumVatRates	1.3	Long	R	Open
PostLine	1.6	String	R/W	Open, Claim, & Enable
PredefinedPaymentLines	1.3	String	R	Open
PreLine	1.6	String	R/W	Open, Claim, & Enable
PrinterState	1.3	Long	R	Open, Claim, & Enable
QuantityDecimalPlaces	1.3	Long	R	Open, Claim, & Enable
QuantityLength	1.3	Long	R	Open, Claim, & Enable
RecEmpty(2)	1.3	Boolean	R	Open, Claim, & Enable
RecNearEnd(2)	1.3	Boolean	R	Open, Claim, & Enable
RemainingFiscalMemory	1.3	Long	R	Open, Claim, & Enable
ReservedWord(1)	1.3	String	R	Open
SlpEmpty(2)	1.3	Boolean	R	Open, Claim, & Enable
SlpNearEnd(2)	1.3	Boolean	R	Open, Claim, & Enable
SlipSelection	1.3	Long	R/W	Open, Claim, & Enable
TotalizerType	1.6	Long	R/W	Open, Claim, & Enable
TrainingModeActive	1.3	Boolean	R	Open, Claim, & Enable

メソッド

共通	*版数*	*使用可能条件*
Open	1.3	--
Close	1.3	Open
ClaimDevice	1.3	Open
ReleaseDevice	1.3	Open ＆ Claim
CheckHealth	1.3	Open, Claim, & Enable
ClearInput	1.3	*適用外*
ClearInputProperties	1.10	*適用外*
ClearOutput	1.3	Open および Claim
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.3	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用 - フィスカルの事前設定		
SetCcurrency	1.6	Open, Claim, & Enable
SetDate	1.3	Open, Claim, & Enable
SetHeaderLine	1.3	Open, Claim, & Enable
SetPOSID(1)	1.3	Open, Claim, & Enable
SetStoreFiscalID	1.3	Open, Claim, & Enable
SetTrailerLine	1.3	Open, Claim, & Enable
SetVatTable	1.3	Open, Claim, & Enable
SetVatValue	1.3	Open, Claim, & Enable
専用 - フィスカルレシート		
BeginFiscalReceipt	1.3	Open, Claim, & Enable
EndFiscalReceipt	1.3	Open, Claim, & Enable
PrintDuplicateReceipt	1.3	Open, Claim, & Enable
PrintRecCash	1.6	Open, Claim, & Enable
PrintRecItem	1.3	Open, Claim, & Enable
PrintRecItemAdjustment	1.3	Open, Claim, & Enable
PrintRecItemFuel	1.6	Open, Claim, & Enable
PrintRecItemFuelVoid	1.6	Open, Claim, & Enable
PrintRecItemRefund	1.12	Open, Claim, & Enable
PrintRecItemRefundVoid	1.12	Open, Claim, & Enable
PrintRecMessage	1.3	Open, Claim, & Enable
PrintRecNotPaid	1.3	Open, Claim, & Enable
PrintRecPackageAdjustment	1.6	Open, Claim, & Enable

PrintRecPackageAdjustVoid	1.6	Open, Claim, & Enable
PrintRecRefund	1.3	Open, Claim, & Enable
PrintRecRefundVoid	1.6	Open, Claim, & Enable
PrintRecSubtotal	1.3	Open, Claim, & Enable
PrintRecSubtotalAdjustment	1.3	Open, Claim, & Enable
PrintRecSubtotalAdjustVoid	1.6	Open, Claim, & Enable
PrintRecTaxID	1.6	Open, Claim, & Enable
PrintRecTotal	1.3	Open, Claim, & Enable
PrintRecVoid	1.3	Open, Claim, & Enable
PrintRecVoidItem	1.3	*適用外*

専用 - フィスカルドキュメント		
BeginFiscalDocument	1.3	Open, Claim, & Enable
EndFiscalDocument	1.3	Open, Claim, & Enable
PrintFiscalDocumentLine	1.3	Open, Claim, & Enable
専用 - 品目一覧		
BeginItemList(1)	1.3	Open, Claim, & Enable
EndItemList(1)	1.3	Open, Claim, & Enable
VerifyItem(1)	1.3	Open, Claim, & Enable
専用 - フィスカルレポート		
PrintPeriodicTotalsReport	1.3	Open, Claim, & Enable
PrintPowerLossReport	1.3	Open, Claim, & Enable
PrintReport	1.3	Open, Claim, & Enable
PrintXReport	1.3	Open, Claim, & Enable
PrintZReport	1.3	Open, Claim, & Enable
専用 - スリップ挿入		
BeginInsertion(2)	1.3	Open, Claim, & Enable
BeginRemoval(2)	1.3	Open, Claim, & Enable
EndInsertion(2)	1.3	Open, Claim, & Enable
EndRemoval(2)	1.3	Open, Claim, & Enable
専用 - ノンーフィスカル		
BeginFixedOutput(1)	1.3	Open, Claim, & Enable
BeginNonFiscal	1.3	Open, Claim, & Enable
BeginTraining	1.3	Open, Claim, & Enable
EndFixedOutput(1)	1.3	Open, Claim, & Enable
EndNonFiscal	1.3	Open, Claim, & Enable
EndTraining	1.3	Open, Claim, & Enable
PrintFixedOutput(1)	1.3	Open, Claim, & Enable
PrintNormal	1.3	Open, Claim, & Enable
専用 - データ要求		
GetData	1.3	Open, Claim, & Enable
GetDate	1.3	Open, Claim, & Enable
GetTotalizer	1.3	Open, Claim, & Enable
GetVatEntry(1)	1.3	Open, Claim, & Enable
専用 - エラー訂正		
ClearError	1.3	Open, Claim, & Enable
ResetPrinter	1.3	Open, Claim, & Enable

イベント

名前	*版数*	*発生条件*
DataEvent	1.3	*適用外*
DirectIOEvent	1.3	Open, Claim
ErrorEvent	1.3	Open, Claim, & Enable
OutputCompleteEvent	1.3	Open, Claim, & Enable
StatusUpdateEvent	1.3	Open, Claim, & Enable

(1) のマークがあるすべてのメソッドとプロパティは、一部の国固有のものであり、すべての国のフィスカル制度で必要になるわけではありません。(2) のマークがあるプロパティとメソッドは、POSプリンタデバイスから適合させたものであります。

概説

フィスカルプリンタ コントロールのOLEプログラムプログラマティクIDは、 “OPOS.FiscalPrinter”です。

このデバイスは、第1.3 版で追加されました。

詳細はUPOS仕様書Version1.14を参照ください。

第　１８章

PINパッド

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.3	Boolean	R/W	*適用外*
CapCompareFirmwareVersion	1.9	Boolean	R	Open
BinaryConversion	1.5	Long	R/W	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.3	String	R	Open
Claimed	1.3	Boolean	R	Open
DataCount	1.3	Long	R	Open
DataEventEnabled	1.3	Boolean	R/W	Open
DeviceEnabled	1.3	Boolean	R/W	Open & Claim
FreezeEvents	1.3	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.3	Long	R	*適用外*
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.3	Long	R	無し
ResultCodeExtended	1.3	Long	R	Open
State	1.3	Long	R	無し
ControlObjectDescription	1.3	String	R	無し
ControlObjectVersion	1.3	Long	R	無し
ServiceObjectDescription	1.3	String	R	Open
ServiceObjectVersion	1.3	Long	R	Open
DeviceDescription	1.3	String	R	Open
DeviceName	1.3	String	R	Open

専用				
CapMACCalculation	1.3	Boolean	R	Open
CapDisplay	1.3	Long	R	Open
CapLanguage	1.3	Long	R	Open
CapKeyboard	1.3	Boolean	R	Open
CapTone	1.3	Boolean	R	Open
AvailablePromptsList	1.3	String	R	Open
Prompt	1.3	Long	R/W	Open
AvailableLanguagesList	1.3	String	R	Open
PromptLanguage	1.3	Long	R/W	Open
AccountNumber	1.3	String	R/W	Open
Amount	1.3	Currency	R/W	Open
MerchantID	1.3	String	R/W	Open
TerminalID	1.3	String	R/W	Open
Track1Data	1.3	String	R/W	Open
Track2Data	1.3	String	R/W	Open
Track3Data	1.3	String	R/W	Open
Track4Data	1.5	String	R/W	Open
TransactionType	1.3	String	R/W	Open
MinimumPINLength	1.3	Long	R/W	Open
MaximumPINLength	1.3	Long	R/W	Open
PINEntryEnabled	1.3	Boolean	R	Open
EncryptedPIN	1.3	String	R	Open
AdditionalSecurity Information	1.3	String	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.3	無し
Close	1.3	Open
ClaimDevice	1.3	Open
ReleaseDevice	1.3	Open & Claim
CheckHealth	1.3	Open, Claim, & Enable
ClearInput	1.3	Open, Claim, & Enable
ClearInputProperties	1.10	Open & Claim
ClearOutput	1.3	*適用外*
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.3	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用		
BeginEFTTransaction	1.3	Open, Claim, & Enable
EndEFTTransaction	1.3	BeginEFTTransaction
EnablePINEntry	1.3	BeginEFTTransaction
ComputeMAC	1.3	BeginEFTTransaction
VerifyMAC	1.3	BeginEFTTransaction
UpdateKey	1.3	BeginEFTTransaction

イベント

名称	*版数*	*発生条件*
DataEvent	1.3	Open, Claim, & Enable
DirectIOEvent	1.3	Open & Claim
ErrorEvent	1.3	Open, Claim, & Enable
OutputCompleteEvent	1.3	*適用外*
StatusUpdateEvent	1.3	Open, Claim, & Enable

概説

PINパッドコントロールのOLEプログラマティックIDは、“OPOS.PINPad”です。

このデバイスは第 1.3 版で追加されました。

ここで扱うPINパッドは、次の機能を提供します。

顧客がPIN入力を行えるメカニズムを提供します。

EFT取り引きを行うホストコンピューターと通信を行うための暗号化エンジンとして動作します。

PINパッドは、一つ以上のPINパッドマネージメントシステムを実装することでこれらの機能を実行します。 PINパッドマネージメントシステムでは、PINパッドがPINの暗号化、メッセージ認証コードの計算、および暗号化キーの更新などのような機能を実行する方法を定義します。 PINパッドマネージメントシステムの例としては、Master-Session、DUKPT、APACS40、 HGEPOS、AS2805、JDEBIT2、その他多数があります。

機能

PINパッドコントロールは、次の機能を最低限持っています。

装置上のキーボードでのPIN入力を認識し、アプリケーションへ暗号化したPINを通知します。

PINパッドコントロールは、次の付加的な機能を有する場合があります。

メッセージ認証番号の計算。

選択されたPINパッドマネージメントシステムに従って暗号化キーの更新。

複数のPINパッドマネージメントシステムの対応。

アプリケーションからのPINパッド上のキーボード、ディスプレイ、およびトーン装置の使用。これらの装置のいずれかが使用可能の場合、アプリケーションは、POSキーボード、ラインディスプレイまたはトーンに対応するコントロールオブジェクトをオープンして使用します。

例外項目

当仕様では、以下への対応は含まれておりません。

初期暗号化キーのローディング。通常この操作は、少なくとも一つの暗号化キーを平文でダウンロードする必要があり、かつ安全な場所（一般的には、工場か金融施設のいずれか）で実行されなければなりません。従って、初期暗号化キーのローディングは当仕様の範囲外となります。しかし、当仕様では、PINパッド装置が商用店にインストールされる限りにおいては暗号化キーの更新機能を含みます。

完全なEFT機能。当仕様は、EFT処理アプリケーションが、単に周辺装置として使用するPINパッドの機能について言及しています。インテリジェントなPINパッドの範囲内で処理を実行するEFT処理アプリケーションの機能については特に定義しません。当仕様では、メッセージをEFTトランザクションホストコンピューターに送信されなければならないということをPINパッドアプリケーションが決定するアプリケーションへの対応は含みません。*従って、この仕様は、カナダ、ドイツ、オランダ、およびその他の国々には適用されません。ベンダーがPINパッドにてEFT機能を提供できるとして選んだPINパッドには適用されません。*

スマートカードリーダー。スマートカードリーダーを備えるPIN パッド装置もあります。この装置への対応は、 当仕様の将来の版に盛り込まれる場合もあります。当面はそのような追加機能の制御にはDirectIOメソッドを使うことができます。

用語の注意事項

PINパッドデバイスのために、デバイスでデータ交換を記述するのに使用される用語の明確化が必要です。「16進ASCII」は， 16進ASCII文字としてのバイトの「標準の」表現を示すのに使用されます。例えば、{0x15, 0xC7, 0xF0} は16進ASCIIでは”15C7F0”とあらわされます。

モデル

PINパッド装置は、PINパッドマネージメントシステムの制御のもとで暗号化を行います。複数のPINパッドマネージメントシステムに対応するPINパッド装置もあります。異なったEFTトランザクションホストコンピューターに対して複数の暗号化キー（集合）に対応するPINパッドマネージメントシステムもあります。よって、おのおののEFT取り引きにおいては、アプリケーションは使用すべきPINパッドマネージメントシステムとEFTトランザクションホストコンピューターを選択しなければなりません。
PINパッドマネージメントシステムによっては、暗号化のために一つ以上のEFT取り引き用パラメーターをPINパッド装置に提供しなければなりません。アプリケーションは、異なったPINパッドマネージメントシステムを要するEFTトランザクションホストコンピューターへの移行をより簡単に行えるようにするために、**すべての**EFT取り引き用パラメータープロパティの値を設定すべきです。

PINパッドコントロールをオープンし、排他制御権を獲得し、そして利用可能にした後、アプリケーションはおのおののEFT取り引きを処理するには以下の一般的なシナリオを使用すべきです。

アプリケーションはEFT取り引きのパラメーター(AccountNumber、Amount、MerchantID、TerminalID、Track1Data、 Track2Data、Track3Data、Track4Data 、 お よ び TransactionType 各 プ ロ パ テ ィ) を 設 定 し 、BeginEFTTransactionメソッドを実行しなければなりません。これは、EFT取り引きのための暗号化機能を司るサービスオブジェクトとPINパッドを初期化します。

PIN入力が必要であれば、EnablePINEntryメソッドを呼び出します。それから、 DataEventEnabledプロパティを設定しDataEventイベントを待ちます。

メッセージ認証コードが必要であれば、必要に応じてComputeMACとVerifyMACメソッドを使用します。

EFT取り引きのためのすべてのオペレーションを完了したことをコントロールに通知するためにEndEFTTransactionメソッドを実行します。

当仕様では，PINパッドのディスプレイの使用について2つのモデルをサポートします。**CapDisplay**プロパティが個々のデバイスにおいてどのモデルがサポートされるかを示します。

一つのモデルでは、アプリケーションが表示されるべきテキストの完全な制御を握ります。このモデルのためには、ディスプレイを制御するアプリケーションにより使用される関連付けられたOPOSラインディスプレイコントロールがあります。

もう一つのモデルでは、アプリケーションは表示されるテキストを自由にできません。その代わり、アプリケーションは、表示される予め定義されたメッセージのリストから選択することのみ可能です。このモデルのためには、ディスプレイを制御するために使用される一連のプロパティがあります。

デバイスの共有

PINパッドは、下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、デバイスの読み込みを始める前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の必要条件については、「一覧」の表を参照してください。

プロパティ

AccountNumber プロパティ　R/W

形式　BSTR AccountNumber;

説明　現在のEFT取り引きで使用されるアカウント番号。アプリケーションは、 **BeginEFTTransaction**を呼び出す前にこのプロパティを設定しなければなりません。 **BeginEFTTransaction**メソッドが呼び出された以降のこのプロパティ値を変更するすべての試みは、OPOS_E_ILLEGALの値が**ResultCode**に格納されることになります。

戻り値　このプロパティが設定されると、次のいずれかの値が**ResultCode**プロパティへ設定されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	このプロパティが設定される前に**BeginEFTTransaction**メソッドが呼び出されています。
その他	**ResultCode**を参照のこと。

AdditionalSecurityInformation プロパティ

形式　BSTR AdditionalSecurityInformation;

説明　このプロパティは、**DataEvent**イベント後の付加的なセキュリティ/暗号化情報になります。このプロパティは、Hex-ASCII文字列として書式化されます。この文字列の情報内容と内部フォーマットは、PINパッドマネジメントシステムにおいて異なります。例えば、PINパッドマネジメントシステムがDUKPTの場合には、このプロパティは“PINパッド連続番号”となります。PIN入力がキャンセルされた場合には、このプロパティは空の文字列となります。

Amount プロパティ　R/W

形式　CURRENCY Amount;

説明　現在のEFT取り引きの金額。アプリケーションは、**BeginEFTTransaction**を呼び出す前にこのプロパティを設定しなければなりません。**BeginEFTTransaction**メソッドが呼び出された以降のこのプロパティ値を変更するすべての試みは、 OPOS_E_ILLEGALの値が**ResultCode**に格納されることになります。

戻り値　このプロパティが設定されると、次のいずれかの値が**ResultCode**プロパティへ設定されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	このプロパティが設定される前に**BeginEFTTransaction**メソッドが呼び出されています。
その他	**ResultCode**を参照のこと。

AvailableLanguagesList プロパティ

形式　BSTR AvailableLanguagesList;

説明　このプロパティは、PINパッド装置において予め定義されるプロンプトによりサポートされる言語のカンマセパレート型のストリングです。言語は数値であり、Microsoft Language IDです。**CapLanguage** = PPAD_LANG_NONEの場合には、このプロパティは、空の文字列となります。

このプロパティは**Open**メソッドによって初期化されます。

AvailablePromptsList プロパティ

形式 BSTR AvailablePromptsList;

説明 このプロパティは**Prompt**プロパティに対してサポートされた値のカンマセパレート型のストリングです。

名称（値）	意味
PPAD_MSG_ENTERPIN (1)	ユーザーはPINパッド上でPIN番号を入力すべきです。
PPAD_MSG_PLEASEWAIT (2)	システムは処理中です。ユーザーは待つべきです。
PPAD_MSG_ENTERVALIDPIN (3)	入力されたPINは正しくありません。 ユーザーは正しいPIN番号を入力すべきです。
PPAD_MSG_RETRIESEXCEEDED (4)	ユーザーは正しいPIN番号の入力に失敗し、試行の最大値を超えました。
PPAD_MSG_APPROVED (5)	要求は承認されました。
PPAD_MSG_DECLINED (6)	EFTトランザクションホストは、要求された機能の実行を拒否しました。
PPAD_MSG_CANCELED (7)	要求はキャンセルされました。
PPAD_MSG_AMOUNTOK (8)	顧客は金額を承認するためにYes/Noを入力すべきです。
PPAD_MSG_NOTREADY (9)	PINパッドは顧客の使用には準備できていません。
PPAD_MSG_IDLE (10)	システムはアイドルです。
PPAD_MSG_SLIDE_CARD (11)	ユーザーは統合されたMSRからカードを読み込ませるべきです。
PPAD_MSG_INSERTCARD (12)	顧客は(スマート)カードを挿入すべきです。
PPAD_MSG_SELECTCARDTYPE (13)	顧客はカードのタイプ（典型的には、クレジットかデビット）を選択すべきです。

1000以上の値は、デバイス固有定義値として予約されています。

このプロパティは**Open**メソッドによって初期化されます。

CapDisplay プロパティ

形式　LONG CapDisplay;

説明　アプリケーションがPINパッドのディスプレイ装置へ行うことができるオペレーションを定義します。

値	意味
PPAD_DISP_UNRESTRICTED	アプリケーションは、メッセージを表示することにおいて制限無くPINパッドのディスプレイ装置を使用することができます。この場合、関連づけられたラインディスプレイコントロールオブジェクトが、PINパッドのディスプレイ装置へのインターフェースとなります。アプリケーションはディスプレイ装置を制御するために、ラインディスプレイのメソッドを呼び出さなければなりません。
PPAD_DISP_PINRESTRICTED	アプリケーションは、PIN入力中以外において制限無くPINパッドのディスプレイ装置を使用することができます。PINパッドは、PIN入力中，予め定義されたメッセージを表示します。PIN入力となっている間に関連付けされたラインディスプレイコントロールオブジェクトを使用した場合，ラインディスプレイコントロールはOPOS_E_BUSYを返します。
PPAD_DISP_RESTRICTED_LIST	アプリケーションは、表示するメッセージの文字列を指定することができません。予め定義されたメッセージの中から選択できます。関連付けされたラインディスプレイコントロールオブジェクトはありません。
PPAD_DISP_RESTRICTED_ORDER	アプリケーションは、表示するメッセージの文字列を指定することができません。予め定義されたメッセージの中から選択できます。選択は、予め定義された許容できる順序でなければなりません。関連付けされたラインディスプレイコントロールオブジェクトはありません。
PPAD_DISP_NONE	PINパッドは、PINパッドのディスプレイ装置を持っていません。

このプロパティは**Open**メソッドによって初期化されます。

CapLanguage プロパティ　　第1．9版で更新

形式　LONG CapLanguage;

説明　アプリケーションが予め定義されたメッセージの言語（例えば、英語、フランス語、アラビア語）を選択しなければならない能力を定義します。

値	説明
PPAD_LANG_NONE	PINパッドは、予め定義されるプロンプトメッセージをサポートしません。もし**CapDisplay**プロパティの値が、PPAD_DISP_UNRESTRICTED である場合、プロパティ値はこの値に設定されます。**PromptLanguage**プロパティへの設定は、**ResultCode**プロパティはOPOS_E_ILLEGALとなります。
PPAD_LANG_ONE	PINパッドは、一つの言語の予め定義されるプロンプトメッセージをサポートします。**PromptLanguage**プロパティへデフォルト値以外の設定は、**ResultCode**プロパティはOPOS_E_ILLEGALとなります。
PPAD_LANG_PINRESTRICTED	PINパッドは、PIN入力中にプロンプト言語を変更することはできません。アプリケーションは、**EnablePINEntry**を呼び出す前に**PromptLanguage**プロパティへ適切な値を設定しなければなりません。 **EnablePINEntry**がTRUEであるときに **PromptLanguage**へ値を設定すると、 **ResultCode**プロパティはOPOS_E_BUSYとなります。
PPAD_LANG_UNRESTRICTED	アプリケーションは、いつでも予め定義されるプロンプトメッセージの言語を変更することができます。現在表示されているメッセージは、即時に変わります。

このプロパティは**Open**メソッドによって初期化されます。

CapMACCalculation プロパティ

形式 BOOL CapMACCalculation;

説明 TRUE: PINパッドは、MAC計算をサポートします。

このプロパティは**Open**メソッドによって初期化されます。

CapKeyboard プロパティ

形式 BOOL CapKeyboard;

説明 アプリケーションがPINパッドのキーボードからの入力を得られるかどうかを定義します。

TRUE: アプリケーションは入力を得るためにPINパッドを使用することができます。アプリケーションは、PINパッドのキーボードへのインターフェースとして関連付けられたPOSキーボードコントロールを使用します。**PINEntryEnabled**がTRUEである間は、関連するPOS Keyboardコントロールは実際上無効となります。
FALSE: アプリケーションは、PINパッドのキーボードから入力を直接得ることができません。

このプロパティは**Open**メソッドによって初期化されます。

CapTone プロパティ

形式 BOOL CapTone;

説明 TRUE: PINパッドはトーン装置をもっています。このトーン装置は、関連付けられたトーンインジケーターコントロールを使ってアクセスされます。
FALSE: トーン装置はありません。

このプロパティは**Open**メソッドによって初期化されます。

EncryptedPIN プロパティ

形式 BSTR EncryptedPIN;

説明 このプロパティは、 **DataEvent**イベント後の暗号化されたPINの値となっています。このプロパティは、16進ASCII文字列となっています。おのおのの文字は、'0'から'9'または'A'から'F'の範囲です。おのおのの文字のペアは、1バイトの16進です。
例：例えば最初の4文字が"12FA"ならば、PINの最初の2バイトは16進の12（18）、次は16進のFA（250）です。
PIN入力がキャンセルされた場合には、このプロパティは空の文字列となります。

MaximumPINLength プロパティ R/W

形式 LONG MaximumPINLength;

説明 アプリケーションは、このプロパティをPINにおいて受け入れられることができる最大デジット数に設定すべきです。このプロパティは、**EnablePINEntry**メソッドが実行される前にアプリケーションから設定されなければなりません。このプロパティは、**Open**メソッドによりデフォルト値に設定されます。実装によっては、この値はアプリケーションによって変更されることができません。

MerchantID プロパティ　R/W

形式　BSTR MerchantID;

説明　取り引きID。EFTトランザクションホストで識別されるIDです。アプリケーションは、**BeginEFTTransaction**を呼び出す前に、このプロパティを設定しなければなりません。 **BeginEFTTransaction**メソッドが呼び出された以降の、このプロパティ値を変更するすべての試みは、 OPOS_E_ILLEGALの値が**ResultCode**に格納されることになります。

戻り値　このプロパティが設定されると、次のいずれかの値が**ResultCode**プロパティへ設定されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	このプロパティが設定される前に**BeginEFTTransaction**メソッドが呼び出されています。
その他	**ResultCode**を参照のこと。

MinimumPINLength プロパティ　R/W

形式　LONG MinimumPINLength;

説明　アプリケーションは、このプロパティをPINにおいて受け入れられることができる最小デジット数に設定すべきです。このプロパティは、**EnablePINEntry**メソッドが実行される前にアプリケーションから設定されなければなりません。このプロパティは、**Open**メソッドによりデフォルト値に設定されます。実装によっては、この値はアプリケーションによって変更されることができません。

PINEntryEnabled プロパティ

形式　　BOOL PINEntryEnabled;

説明　　PINパッドコントロールオブジェクトは、**EnablePINEntry**メソッドが実行されたときにこのプロパティをTRUEにします。ユーザーがPIN入力操作を完了したとき、あるいは、**EndEFTTransaction**が実行されたときにFALSEに設定されます。

Prompt プロパティ　R/W

形式　LONG Prompt;

説明　このプロパティは、PINパッド上に表示される予め定義されたメッセージを意味します。このプロパティは、**CapDisplay**がPPAD_DISP_RESTRICTED_LISTかPPAD_DISP_RESTRICTED_ORDERである場合に使用されます。**CapDisplay**がPPAD_DISP_PINRESTRICTEDである場合には、PIN入力中にも使用されます。サービスオブジェクトが受け入れられるこのプロパティへの値は、**AvailablePromptsList**プロパティに一覧されます。

このプロパティは**Open**メソッドによって初期化されます。

戻り値　このプロパティが設定されると、次のいずれかの値が**ResultCode**プロパティへ設定されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	次のいずれかが起こりました。 PINパッドサービスオブジェクトにより提供されない値がこのプロパティへ設定されようとしました。 認められない順序(**CapDisplay** = PPAD_DISP_RESTRICTED_ORDER)でプロンプトメッセージが選択されようとしました。
その他	**ResultCode**を参照のこと。

参照　**PromptLanguage;**

PromptLanguage プロパティ　R/W

形式　LONG PromptLanguage;

説明　このプロパティは、(**Prompt**プロパティで指定された)表示されるメッセージの言語を指定します。**Prompt**プロパティが使用されていれば、このプロパティは使用されます。このプロパティを変更するときの厳密な影響は、 **CapLanguage**プロパティの値に依存しています。

このプロパティの値は、MS Windows Language IDです。このプロパティは、 **Open**メソッドによってデフォルト値に初期化されます。

戻り値　このプロパティが設定されると、次のいずれかの値が**ResultCode**プロパティへ設定されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	PINパッドサービスオブジェクトにより提供されない値がこのプロパティへ設定されようとしました。
その他	**ResultCode**を参照のこと。

参照　CapLanguage, AvailableLanguagesList

TerminalID プロパティ　R/W

形式　BSTR TerminalID;

説明　端末ID。 EFTトランザクションホストで識別されるIDです。アプリケーションは、**BeginEFTTransaction**を呼び出す前に、このプロパティを設定しなければなりません。 **BeginEFTTransaction**メソッドが呼び出された以降の、このプロパティ値を変更するすべての試みは、 OPOS_E_ILLEGALの値が**ResultCode**に格納されることになります。

戻り値　このプロパティが設定されると、次のいずれかの値が**ResultCode**プロパティへ設定されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	このプロパティが設定される前に**BeginEFTTransaction**メソッドが呼び出されています。
その他	**ResultCode**を参照のこと。

Track1Data プロパティ　R/W　*第1.14.1版で更新*

形式　BSTR Track1Data;

説明　直前に読まれたカードのコード化されたトラック1データ、または、空の文字列となります。空の文字列は、そのトラックが物理的に読まれなかったことを意味します。アプリケーションは、**BeginEFTTransaction**を呼び出す前に、このプロパティを設定しなければなりません。 **BeginEFTTransaction**メソッドが呼び出された以降の、このプロパティ値を変更するすべての試みは、OPOS_E_ILLEGALの値が**ResultCode**内に格納されることになります。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

戻り値　このプロパティが設定されると、次のいずれかの値が**ResultCode**プロパティへ設定されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	このプロパティが設定される前に**BeginEFTTransaction**メソッドが呼び出されています。
その他	**ResultCode**を参照のこと。

Track2Data プロパティ　R/W　*第1.14.1版で更新*

形式　BSTR Track2Data;

説明　直前に読まれたカードのコード化されたトラック2データ、または、空の文字列となります。空の文字列は、そのトラックが物理的に読まれなかったことを意味します。アプリケーションは、**BeginEFTTransaction**を呼び出す前に、このプロパティを設定しなければなりません。 **BeginEFTTransaction**メソッドが呼び出された以降の、このプロパティ値を変更するすべての試みは、OPOS_E_ILLEGALの値が**ResultCode**内に格納されることになります。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

戻り値　このプロパティが設定されると、次のいずれかの値が**ResultCode**プロパティへ設定されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	このプロパティが設定される前に**BeginEFTTransaction**メソッドが呼び出されています。
その他	**ResultCode**を参照のこと。

Track3Data プロパティ　R/W　*第1.14.1版で更新*

形式　BSTR Track3Data;

説明　直前に読まれたカードのコード化されたトラック3データ、または、空の文字列となります。空の文字列は、そのトラックが物理的に読まれなかったことを意味します。アプリケーションは、**BeginEFTTransaction**を呼び出す前に、このプロパティを設定しなければなりません。 **BeginEFTTransaction**メソッドが呼び出された以降の、このプロパティ値を変更するすべての試みは、OPOS_E_ILLEGALの値が**ResultCode**内に格納されることになります。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

戻り値　このプロパティが設定されると、次のいずれかの値が**ResultCode**プロパティへ設定されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	このプロパティが設定される前に**BeginEFTTransaction**メソッドが呼び出されています。
その他	**ResultCode**を参照のこと。

Track4Data プロパティ　R/W　*第1.14.1版で更新*

形式　BSTR Track4Data;

説明　直前に読まれたカードのコード化されたトラック4（JIS-II）データ、または、空の文字列となります。空の文字列は、そのトラックが物理的に読まれなかったことを意味します。アプリケーションは、**BeginEFTTransaction**を呼び出す前に、このプロパティを設定しなければなりません。 **BeginEFTTransaction**メソッドが呼び出された以降の、このプロパティ値を変更するすべての試みは、OPOS_E_ILLEGAL値が**ResultCode**内に格納されることになります。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

以前のバージョンとの整合性を保つために、JIS-IIデータは、他のTrack*n*Dataにも格納されている可能性がありますが、アプリケーションの互換性を保つために、できるだけこのプロパティを使用することを推奨します。

戻り値　このプロパティが設定されると、次のいずれかの値が**ResultCode**プロパティへ設定されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	このプロパティが設定される前に**BeginEFTTransaction**メソッドが呼び出されています。
その他	**ResultCode**を参照のこと。

TransactionType プロパティ　R/W

形式　LONG TransactionType;

説明　現在のEFT取り引きのタイプ。アプリケーションは、**BeginEFTTransaction**を呼び出す前に、このプロパティを設定しなければなりません。 **BeginEFTTransaction**メソッドが呼び出された以降の、このプロパティ値を変更するすべての試みは、OPOS_E_ILLEGALの値が**ResultCode**内に格納されることになります。

TransactionTypeは、以下の値のいずれかです。

値	説明
PPAD_TRANS_DEBIT	指定された口座からデビット（減少）します。
PPAD_TRANS_CREDIT	指定された口座からクレジット（増加）します。
PPAD_TRANS_INQ	残高照会をします。
PPAD_TRANS_RECONCILE	リコンシリエーション/セトルメント（清算）
PPAD_TRANS_ADMIN	管理上の取り引き

戻り値　このプロパティが設定されると、次のいずれかの値が**ResultCode**プロパティへ設定されます。

値	説明
OPOS_SUCCESS	正常に設定されました。
OPOS_E_ILLEGAL	このプロパティが設定される前に**BeginEFTTransaction**メソッドが呼び出されています。
その他	**ResultCode**を参照のこと。

メソッド

BeginEFTTransaction メソッド　*第1. 14版にて更新*

形式　LONG BeginEFTTransaction (BSTR *PINPadSystem*,LONG *TransactionHost*);

パラメータ	説明
PINPadSystem	要求されるPINパッドマネージメントシステムの名称です。この仕様によって定義される、以下のPINパッドマネジメントシステム名を参照してください。 *注意:サービスオブジェクトの実装者により、以下に定義されていないPINパッドマネジメントシステム名がサポートされる場合があります。アプリケーションがそのPINパッドマネジメントシステム名を使う場合には、それに関する知識を持っている必要があります。*
TransactionHost	このトランザクションに使用される特定のEFTトランザクションホストを意味します。

*PINPadSystem*値には、以下のものがあります。

値	説明
“M/S”	Master/Session (USA, Latin America)
“DUKPT”	Derived Unique Key Per Transaction (USA, Latin America)
“APACS40”	Standard 40 (UK及び諸国)
“AS2805”	Australian Standard 2805
“HGEPOS”	(Italian)
“JDEBIT2”	Japan Debit 2

説明　このメソッドは、PINパッドコントロールへEFT取り引きの始まりを知らせるために、アプリケーションによって呼ばれなければなりません。これにより、PINパッドコントロールは、初期化処理（例えばセッションキーの計算）を実行します。
このメソッドが呼び出されるまで、他のどのようなPINパッド関数も実行することはできません。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明

OPOS_SUCCESS	メソッドは正常に終了しました。
OPOS_E_ ILLEGAL	下記エラーの内の一つが発生しました。 • 要求されたPINパッドマネジメントシステムは当サービスオブジェクトによりサポートされていません。 • 要求されてEFTトランザクションホストは、選択されたPINパッドマネジメントシステムにおいて無効です。
OPOS_E_BUSY	PINパッドは既にEFT取り引きを実行しています。
その他	**ResultCode**を参照のこと。

ComputeMAC メソッド

形式 LONG ComputeMAC (BSTR *InMsg*, BSTR* *pOutMsg*);

パラメータ	説明
InMsg	アプリケーションがEFTトランザクションホストへ送信するメッセージです。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
pOutMsg	*InMsg*に対してMAC計算を施した結果へのポインタです。 出力される値は、実際にEFTトランザクションホストへ送信されるリフォーマットされたメッセージとなります。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

説明 このメソッドは、PINパッドにMAC値を計算させて、指示されたメッセージに付加させるためにアプリケーションにより呼び出されます。選択されたPINパッドマネジメントシステムによっては、PINパッドがメッセージ中に他のフィールドを挿入する場合もあります。
ただし、**ComputeMAC**メソッドは、PIN入力が有効となっている間には使用することができません。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは成功しました。
OPOS_E_DISABLED	**BeginEFTTransaction**メソッドが実行されていません。
OPOS_E_BUSY	**PINEntryEnabled**はTRUEです。PINパッドは、PIN入力中MAC計算をできません。
その他	**ResultCode**を参照のこと。

EnablePINEntry メソッド

形式　LONG EnablePINEntry();

説明　このメソッドは、PINパッド装置でPIN入力を可能にするために、アプリケーションから呼ばれます。このメソッドが呼ばれると、**PINEntryEnabled**プロパティはTRUEとなります。
PINパッドが、PIN入力用にあらかじめ定義されたプロンプトを使う場合、**Prompt**プロパティの値は PPAD_MSG_ENTERPIN となります。

ユーザーが（PIN入力かキャンセルを押したことにより） PIN入力操作を完了したとき、**PINEntryEnabled**プロパティはFALSEとなります。
DataEventEnabledプロパティがTRUEに設定されたとき、暗号化されたPINをアプリケーションへ提供するために**DataEvent**イベントが発行されます。**PINEntryEnabled**がTRUEである間にPINパッドで入力されたすべてのデータは、このコントロールオブジェクトに暗号化された形で供給され、関連するPOS Keyboardコントロールオブジェクトには提供されません。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは成功しました。
OPOS_E_DISABLED	**BeginEFTTransaction**メソッドが実行されていません。
その他	**ResultCode**を参照のこと。

EndEFTTransaction メソッド

形式 LONG EndEFTTransaction (LONG *CompletionCode*);

*CompletionCode*は以下の値のいずれかです。

値	説明
PPAD_EFT_NORMAL	EFT取り引きは正常に完了しました。ただし、EFT取り引きが承認されたことを意味するものではありません。単にメッセージの適切なシーケンスが送受信されたことを意味します。
PPAD_EFT_ABNORMAL	メッセージの適切なシーケンスが送受信されませんでした。

説明 このメソッドは、PINパッドコントロールへEFT取り引きの終了を通知するために、アプリケーションにより呼び出さなければなりません。これによりPINパッドコントロールは、終了処理（例えば、次のトランザクションキーの計算）を実行します。

戻り値 次の値のいずれかが戻され、ResultCodeプロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは成功しました。
その他	ResultCodeを参照のこと。

UpdateKey メソッド

形式　LONG UpdateKey (LONG *KeyNum*, BSTR *Key*);

パラメータ	説明
KeyNum	暗号化キーの番号です。
Key	新しい暗号化キーの16進のASCII型の値です。

説明　このメソッドは、PINパッドへ新しい暗号化キーを設定するために使用されます。これは、新しい暗号化キー値が、EFTトランザクションホストからの標準メッセージ中の一つのフィールドとして、端末へ送信されるPINパッドマネジメントシステムにおいてのみ使用されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	指定された暗号化キーが正常に更新されました。
OPOS_E_BUSY	PINパッドは現時点において新しい暗号化キーを受け付けることができません。
OPOS_E_ILLEGAL	以下のいずれかの状態が起こりました。選択されたPINパッドマネジメントシステムは、この機能をサポートしていません。*KeyNum*は、受け付けることができない暗号化キーの番号です。*Key*は正しくない暗号化キーです（16進のASCIIでない、誤ったデータ長、あるいは、パリティが誤っています）。
その他	**ResultCode**を参照のこと。

VerifyMAC メソッド

形式 **LONG VerifyMAC (BSTR *Message*);**

*Message*は、EFTトランザクションホストから受信したメッセージです。

説明 このメソッドは、PINパッドにEFTトランザクションホストから受信したメッセージ中のMAC値を検証させるためにアプリケーションにより呼び出されます。このメソッドは、当該メッセージが正しいものと判断できる場合にはTRUEを返し、それ以外の場合には、FALSEを返します。 **VerifyMAC**メソッドはPIN入力が有効となっている間には使用することができません。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは成功しました。
OPOS_E_DISABLED	**BeginEFTTransaction**メソッドが実行されていません。
OPOS_E_BUSY	**PINEntryEnabled**がTRUEです。PINパッドは、PIN 入力中には、MAC検証をできません。
OPOS_E_FAILURE	サービスオブジェクトは、メッセージのMAC値の比較に失敗しました。
その他	**ResultCode**を参照のこと。

イベント

DataEvent イベント

形式　　VOID DataEvent (LONG *Status*);

*Status*引数は、以下の値のいずれかとなります。

値	説明
PPAD_SUCCESS	PIN入力があり、値がEncryptedPINとAdditionalSecurityInformationプロパティへ格納されました。
PPAD_CANCEL	ユーザーはPINパッドのキャンセルボタンを押下しました。
PPAD_TIMEOUT	PINパッドにてタイムアウトが発生しました（すべてのPINパッドがこの状態を報告するとは限りません）。

説明　　PIN入力のオペレーションが完了後に通知されます。

ErrorEvent イベント

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

パラメータ	説明
ResultCode	エラーイベントを引き起こす元となった結果コード。以下の値を参照のこと。
ResultCodeExtended	エラーイベントを引き起こす元となった拡張結果コード。以下の値を参照のこと。
ErrorLocus	エラーの位置。以下の値を参照のこと。
pErrorResponse	エラーイベントへの応答へのポインター。以下の値を参照のこと。

*ResultCode*引数は以下の値のいずれかとなっています。

値	説明
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_EPPAD_BAD_KEY: 暗号化キーが壊れているか見当たりません。
その他	**ResultCode**を参照のこと。

*ErrorLocus*引数は以下の値のいずれかとなっています。

値	説明
OPOS_EL_INPUT	イベント駆動型の入力を処理している最中にエラーが発生しました。入力データは有効でありません。

*pErrorResponse*パラメーターが示す位置にある内容は、*ErrorLocus*に基づいてデフォルト値が予め設定されています。アプリケーションはその値を次のいずれかに変更できます。

値	説明
OPOS_ER_CLEAR	バッファリングされている入力データをクリアし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。

説明　PIN暗号化機能を行っているときにエラーを発見すると通知されます。PINパッドサービスオブジェクトは、**CheckHealth**、あるいは、**DirectIO**機能を通じてより詳細な自己診断情報をオプションで用意している場合があります。

第　１９章
リモートオーダディスプレイ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.3	Boolean	R/W	*適用外*
BinaryConversion	1.3	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.3	String	R	Open
Claimed	1.3	Boolean	R	Open
DataCount	1.3	Long	R	Open
DataEventEnabled	1.3	Boolean	R/W	Open
DeviceEnabled	1.3	Boolean	R/W	Open,Claim
FreezeEvents	1.3	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.3	Long	R	Open
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.3	Long	R	無し
ResultCodeExtended	1.3	Long	R	Open
State	1.3	Long	R	無し
ControlObjectDescription	1.3	String	R	無し
ControlObjectVersion	1.3	Long	R	無し
ServiceObjectDescription	1.3	String	R	Open
ServiceObjectVersion	1.3	Long	R	Open
DeviceDescription	1.3	String	R	Open
DeviceName	1.3	String	R	Open

プロパティ (続き)

専用	*版数*	*型*	*アクセス*	*使用可能条件*
CapTransaction	1.3	Boolean	R	Open
AsyncMode	1.3	Boolean	R/W	Open, Claim, & Enable
EventType	1.3	Long	R/W	Open
SystemClocks	1.3	Long	R	Open. Claim, & Enable
SystemVideoSaveBuffers	1.3	Long	R	Open. Claim, & Enable
Timeout	1.3	Long	R/W	Open
UnitsOnline	1.3	Long	R	Open, Claim, & Enable
CurrentUnitID	1.3	Long	R/W	Open, Claim, & Enable
CapSelectCharacterSet	1.3	Boolean	R	Open, Claim, & Enable
CapTone	1.3	Boolean	R	Open, Claim, & Enable
CapTouch	1.3	Boolean	R	Open, Claim, & Enable
AutoToneDuration	1.3	Long	R/W	Open, Claim, & Enable
AutoToneFrequency	1.3	Long	R/W	Open, Claim, & Enable
CharacterSet	1.3	Long	R	Open, Claim, & Enable
CharacterSetList	1.3	String	R	Open, Claim, & Enable
Clocks	1.3	Long	R	Open, Claim, & Enable
VideoDataCount	1.3	Long	R	Open, Claim, & Enable
VideoMode	1.3	Long	R/W	Open, Claim, & Enable
VideoModeList	1.3	String	R	Open, Claim, & Enable
VideoSaveBuffers	1.3	Long	R	Open, Claim, & Enable
ErrorUnits	1.3	Long	R	Open
ErrorString	1.3	String	R	Open
EventUnitID	1.3	Long	R	Open, Claim
EventUnits	1.3	Long	R	Open, Claim
EventString	1.3	String	R	Open, Claim

メソッド

共通	*版数*	*使用可能条件*
Open	1.3	無し
Close	1.3	Open
ClaimDevice	1.3	Open
ReleaseDevice	1.3	Open, Claim
CheckHealth	1.3	Open, Claim, & Enable
ClearInput	1.3	Open, Claim
ClearInputProperties	1.10	*適用外*
ClearOutput	1.3	Open, Claim
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.3	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用		
ControlClock	1.3	Open, Claim, & Enable
ControlCursor	1.3	Open, Claim, & Enable
FreeVideoRegion	1.3	Open, Claim, & Enable
ResetVideo	1.3	Open, Claim, & Enable
SelectCharacterSet	1.3	Open, Claim, & Enable
SetCursor	1.3	Open, Claim, & Enable
ClearVideo	1.3	Open, Claim, & Enable
ClearVideoRegion	1.3	Open, Claim, & Enable
CopyVideoRegion	1.3	Open, Claim, & Enable
DisplayData	1.3	Open, Claim,& Enable
DrawBox	1.3	Open, Claim, & Enable
RestoreVideoRegion	1.3	Open, Claim, & Enable
SaveVideoRegion	1.3	Open, Claim, & Enable
UpdateVideoRegion Attribute	1.3	Open, Claim, & Enable
VideoSound	1.3	Open, Claim, & Enable
TransactionDisplay	1.3	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.3	Open, Claim, & Enable
DirectIOEvent	1.3	Open, Claim
ErrorEvent	1.3	Open, Claim, & Enable
OutputCompleteEvent	1.3	Open, Claim, & Enable
StatusUpdateEvent	1.3	Open, Claim, & Enable

概説

リモートオーダディスプレイコントロールのOLEプログラマティクIDは、“OPOS.RemoteOrderDisplay”です。

このデバイスは、第1.3版にて追加されました。

詳細はUPOS仕様書Version1.14を参照ください。

また次の機能はオプション機能として実装されます。

各々が異なるビデオモードを有する複数のビデオディスプレイ装置のサポート。

- タッチ入力のためのタッチスクリーンディスプレイのサポート。
- 周波数と鳴動時間によるトーン出力のサポート。
- ビデオディスプレイ装置がタッチされた事を確認できる自動トーン出力のサポート(タッチスクリーンディスプレイ時のみ)。
- 一つもしくは複数のビデオ装置への文字セットのダウンローディング機能のサポート。
- 一つもしくは複数のビデオ装置への一括モードでの表示出力機能のサポート。

なお、下記の機能は本バージョンの仕様には含まれていません。

ピクセルまたはドット位置設定によるビデオ表示される画像ディスプレイのサポート。本サポートについては、次期バージョンに向け検討中です。

第 20 章
バンプバー

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.3	Boolean	R/W	*適用外*
BinaryConversion	1.3	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.3	String	R	Open
Claimed	1.3	Boolean	R	Open
DataCount	1.3	Long	R	Open
DataEventEnabled	1.3	Boolean	R/W	Open
DeviceEnabled	1.3	Boolean	R/W	Open,Claim
FreezeEvents	1.3	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.3	Long	R	Open
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.3	Long	R	無し
ResultCodeExtended	1.3	Long	R	Open
State	1.3	Long	R	無し
ControlObjectDescription	1.3	String	R	無し
ControlObjectVersion	1.3	Long	R	無し
ServiceObjectDescription	1.3	String	R	Open
ServiceObjectVersion	1.3	Long	R	Open
DeviceDescription	1.3	String	R	Open
DeviceName	1.3	String	R	Open

プロパティ (続き)

専用	*版数*	*型*	*アクセスs*	*使用可能条件*
AsyncMode	1.3	Boolean	R/W	Open, Claim, & Enable
Timeout	1.3	Long	R/W	Open
UnitsOnline	1.3	Long	R	Open, Claim, & Enable
CurrentUnitID	1.3	Long	R/W	Open, Claim, & Enable
CapTone	1.3	Boolean	R	Open, Claim, & Enable
AutoToneDuration	1.3	Long	R/W	Open, Claim, & Enable
AutoToneFrequency	1.3	Long	R/W	Open, Claim, & Enable
BumpBarDataCount	1.3	Long	R	Open, Claim, & Enable
Keys	1.3	Long	R	Open, Claim, & Enable
ErrorUnits	1.3	Long	R	Open
ErrorString	1.3	String	R	Open
EventUnitID	1.3	Long	R	Open, Claim
EventUnits	1.3	Long	R	Open, Claim
EventString	1.3	String	R	Open, Claim

メソッド

共通	*版数*	*使用可能条件*
Open	1.3	無し
Close	1.3	Open
ClaimDevice	1.3	Open
ReleaseDevice	1.3	Open, Claim
CheckHealth	1.3	Open, Claim, & Enable
ClearInput	1.3	Open, Claim
ClearInputProperties	1.10	*適用外*
ClearOutput	1.3	Open, Claim
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.3	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用		
BumpBarSound	1.3	Open, Claim, & Enable
SetKeyTranslation	1.3	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.3	Open, Claim, & Enable
DirectIOEvent	1.3	Open, Claim
ErrorEvent	1.3	Open, Claim, & Enable
OutputCompleteEvent	1.3	Open, Claim, & Enable
StatusUpdateEvent	1.3	Open, Claim, & Enable

概説

バンプバーコントロールのOLEプログラマティックIDは“OPOS.BumpBar”です。

このデバイスは、第1.3 版で追加されました。

詳細はUPOS仕様書Version1.14を参照ください。

第　２１章

CAT (Credit Authorization Terminal)

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.4	Boolean	R/W	*適用外*
BinaryConversion	1.4	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.4	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.4	String	R	Open
Claimed	1.4	Boolean	R	Open
DataCount	1.4	Long	R	*適用外*
DataEventEnabled	1.4	Boolean	R/W	*適用外*
DeviceEnabled	1.4	Boolean	R/W	Open & Claim
FreezeEvents	1.4	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.4	Long	R	Open
PowerNotify	1.4	Long	R/W	Open
PowerState	1.4	Long	R	Open
ResultCode	1.4	Long	R	無し
ResultCodeExtended	1.4	Long	R	Open
State	1.4	Long	R	無し
ControlObjectDescription	1.4	String	R	無し
ControlObjectVersion	1.4	Long	R	無し
ServiceObjectDescription	1.4	String	R	Open
ServiceObjectVersion	1.4	Long	R	Open
DeviceDescription	1.4	String	R	Open
DeviceName	1.4	String	R	Open

専用				
AccountNumber	1.4	String	R	Open
AdditionalSecurityInformation	1.4	String	R/W	Open

ApprovalCode	1.4	String	R	Open
AsyncMode	1.4	Boolean	R/W	Open
Balance	1.9	CURRENCY	R/W	Open
CapAdditionalSecurityInformation	1.4	Boolean	R	Open
CapAuthorizeCompletion	1.4	Boolean	R	Open
CapAuthorizePreSales	1.4	Boolean	R	Open
CapAuthorizeRefund	1.4	Boolean	R	Open
CapAuthorizeVoid	1.4	Boolean	R	Open
CapAuthorizeVoidPreSales	1.4	Boolean	R	Open
CapCashDeposit	1.9	Boolean	R	Open
CapCenterResultCode	1.4	Boolean	R	Open
CapCheckCard	1.4	Boolean	R	Open
CapDailyLog	1.4	Long	R	Open
CapInstallments	1.4	Boolean	R	Open
CapLockTerminal	1.9	Boolean	R	Open
CapLogStatus	1.9	Boolean	R	Open
CapPaymentDetail	1.4	Boolean	R	Open
CapTaxOthers	1.4	Boolean	R	Open
CapTransactionNumber	1.4	Boolean	R	Open
CapTrainingMode	1.4	Boolean	R	Open
CapUnlockTerminal	1.9	Boolean	R	Open
CardCompanyID	1.4	String	R	Open
CenterResultCode	1.4	String	R	Open
DailyLog	1.4	String	R	Open
LogStatus	1.9	Long	R	Open
PaymentCondition	1.4	Long	R	Open
PaymentDetail	1.4	String	R	Open
PaymentMedia	1.5	Long	R/W	Open
SequenceNumber	1.4	Long	R	Open
SettledAmount	1.9	CURRENCY	R	Open
SlipNumber	1.4	String	R	Open
TrainingMode	1.4	Boolean	R/W	Open
TransactionNumber	1.4	String	R	Open
TransactionType	1.4	Long	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.4	無し
Close	1.4	Open
ClaimDevice	1.4	Open
ReleaseDevice	1.4	Open & Claim
CheckHealth	1.4	Open, Claim, & Enable
ClearInput	1.4	*適用外*
ClearInputProperties	1.10	*適用外*
ClearOutput	1.4	Open & Claim
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.4	Open & Claim
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用		
AccessDailyLog	1.4	Open, Claim, & Enable
AuthorizeCompletion	1.4	Open, Claim, & Enable
AuthorizePreSales	1.4	Open, Claim, & Enable
AuthorizeRefund	1.4	Open, Claim, & Enable
AuthorizeSales	1.4	Open, Claim, & Enable
AuthorizeVoid	1.4	Open, Claim, & Enable
AuthorizeVoidPreSales	1.4	Open, Claim, & Enable
CashDeposit	1.9	Open, Claim, & Enable
CheckCard	1.4	Open, Claim, & Enable
LockTerminal	1.9	Open, Claim, & Enable
UnlockTerminal	1.9	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.4	*適用外*
DirectIOEvent	1.4	Open & Claim
ErrorEvent	1.4	Open, Claim, & Enable
OutputCompleteEvent	1.4	Open, Claim, & Enable
StatusUpdateEvent	1.4	Open, Claim, & Enable

概説

CATコントロールのOLEプログラマティクIDは、“OPOS.CAT”です。

このデバイスは、第1.4版で追加されました。

用語説明

売上

POSでの代金決済において、クレジットカードまたはデビットカードによって決済を可能とする業務です。 支払方法（例：一括、分割、リボルビング払い等）には依存しません。

取消

売上当日、売上処理後に顧客からの購入中止要求あるいは、操作ミス等による売上処理の取消を行う業務です。

返品

売上当日以降、売上処理後に顧客からの返品要求あるいは、何らかの理由により以前の売上処理の取消を行う業務です。

承認後売上

売上承認において、承認機関からのレスポンスが「保留」となったとき、電話連絡等でクレジット会社から承認を得た後に、売上を行う業務です。

事前予約

顧客からのカード提示、CATでのカード入力により、売上承認に先立って利用予定金額分の承認を得て、予信限度額を減額することにより、枠取りを行う業務です。

事前予約の取消

事前予約当日、事前予約処理後に顧客からの中止要求あるいは、操作ミス等による事前予約処理の取消を行う業務です。

カードチェック

顧客からのカード提示、CATでのカード入力により、カードの有効性をクレジット会社に問い合わせる業務です。

認証ログ

CATで承認を得た取引のログです。

支払種別

一括払い、ボーナス払い、分割払い、リボルビング払い、さらに、それらの組合せによる併用支払の種別です。 またデビットでの支払いも可能です。　詳細は**PaymentCondition**、**PaymentMedia**と**PaymentDetail**のプロパティを参照してください。

承認機関

カード情報、ならびに購入金額、支払種別によって、その売上を承認するか決定する機関です。 一般にはカード会社等がこれにあたります。 その他の代行機関が代行する場合もあります。

- 機能　　第1.9版で更新

CATコントロールが対象としているCATデバイスは以下のモデル構成となっています。

CATデバイスはディスプレイ装置、キーボード、磁気カードリーダ、承認用通信装置、伝票印字装置要素から構成されています。

APGではCATコントロールとアプリケーションとのインタフェースを規定しており、CATデバイスの実装には依存しません。　しかし現在CATデバイスの実装としては以下のようなものが存在します。

- 分離型（POS連動）

  専用のCATデバイスをPOSと何らかの方法（例：RS232C）で接続しCATデバイスを実現します。

- 内蔵型

  分離型と同じハード構成でありますがハウジングとしてPOS内部に実装しCATデバイスを実現します。

CATデバイスはCATコントロールからの購入金額、税金等の金額と共に各承認要求を受信し、その種別によって動作を決定します。

CATデバイスはCATコントロールから各承認要求がきたときに、一般的に磁気カードを要求します。

CATデバイスで磁気カードを入力しますと購入金額、税金等のデータを承認通信装置を用いて承認機関と通信します。

CATデバイスは承認機関からの結果をCATコントロールへ返します。この返したデータはCATコントロールによって各プロパティに格納され、アプリケーションに通知されます。

以上のようにアプリケーションから金額と共に承認要求がCATデバイスに送られた後、全てのオペレーションはCATデバイス側で行われます。その後、最終結果としての承認の有無がアプリケーションに返されます。

CATコントロールが対象としている電子マネー決済デバイスは以下のモデル構成となっています。

電子マネー決済デバイスはカードリーダ（接触／非接触）、締め処理用通信装置などから構成されています。

電子マネー決済デバイスはCATコントロールからの金額、税金等の情報と共に決済要求を受信し、決済処理を開始します。

電子マネー決済デバイスはCATコントロールから決済要求があると、一般的にカード（接触／非接触）を要求します。

電子マネー決済デバイスでカード(接触／非接触)が入力されると金額、税金等のデータによりカードからの決済を試みます

電子マネー決済デバイスはカードからの決済結果をCATコントロールへ返します。決済結果はCATコントロールによって各プロパティに格納され、アプリケーションに通知されます。

以上のようにアプリケーションから金額と共に決済要求が電子マネー決済デバイスに送られた後、全てのオペレーションは電子マネー決済デバイス側で行われます。 その後、最終結果としての決済の結果がアプリケーションに返されます。

電子マネー決済デバイスは、決済の結果を取引ログとしてデバイス内部のメモリに保持するかもしれません。そのようなデバイスは、締め処理により取引ログをセンターに送信するかもしれません。

モデル

第1.9版で更新

以下にCATコントロールの一般的なモデルを示します。

CATコントロールは基本的に出力モデルデバイスに従います。 ただし非同期出力時に多重にメソッドは発行できません。

CATコントロールは以下のメソッドによりCATデバイスに各承認要求、および認証ログ取得要求を行います。

機能	メソッド名	該当**Cap**プロパティ
売上	**AuthorizeSales**	なし
取消	**AuthorizeVoid**	**CapAuthorizeVoid**
返品	**AuthorizeRefund**	**CapAuthorizeRefund**
承認後売上	**AuthorizeCompletion**	**CapAuthorizeCompletion**
事前予約	**AuthorizePreSales**	**CapAuthorizePreSales**
事前予約の取消	**AuthorizeVoidPreSales**	**CapAuthorizeVoidPreSales**
カードチェック	**CheckCard**	**CapCheckCard**
認証ログ取得	**AccessDailyLog**	**CapDailyLog**

CATコントロールはCATデバイスから承認結果を受け取ると以下のプロパティに承認結果を格納します。

内容	名称	該当**Cap**プロパティ
会員番号/口座番号	AccountNumber	なし
追加情報	AdditionalSecurityInformation	CapAdditionalSecurityInformation
承認番号	ApprovalCode	なし
カード会社ID/ 金融機関コード（支店コード含む）	CardCompanyID	なし

承認機関からのコード	CenterResultCode	CapCenterResultCode
支払種別	PaymentCondition	なし
支払種別詳細	PaymentDetail	CapPaymentDetail
シーケンス番号	SequenceNumber	なし
伝票番号	SlipNumber	なし
センター処理通番	TransactionNumber	CapTransactionNumber
取引種別	TransactionType	なし

AccessDailyLogメソッドの場合

内容	名称	該当**Cap**プロパティ
取引ログ	DailyLog	CapDailyLog

CATコントロールは以下のメソッドにより電子マネー決済デバイスに各決済要求、および取引ログ締め要求を行います。

機能	メソッド名	該当**Cap**プロパティ
決済	**AuthorizeSales**	なし
チャージ	**CashDeposit**	**CapCashDeposit**
残高照会	**CheckCard**	**CapCheckCard**
取引ログ締め	**AccessDailyLog**	**CapDailyLog**
セキュリティロック設定	**LockTerminal**	**CapLockTerminal**
セキュリティロック解除	**UnlockTerminal**	**CapUnlockTerminal**

CATコントロールは電子マネー決済デバイスから結果を受け取ると以下のプロパティに結果を格納します。

内容	名称	該当**Cap**プロパティ
カード番号	AccountNumber	なし
追加情報	AdditionalSecurityInformation	CapAdditionalSecurityInformation
承認番号	ApprovalCode	なし
決済額	SettledAmount	なし
残高	Balance	なし
シーケンス番号	SequenceNumber	なし
取引種別	TransactionType	なし

AccessDailyLogメソッドの場合

内容	名称	該当**Cap**プロパティ
取引ログ	DailyLog	CapDailyLog

各メソッドの引数にはアプリケーションで*SequenceNumber*を指定し、その結果として格納されたプロパティを参照する場合は、初めにSequenceNumberのプロパティを確認し、対応するレスポンスであるかをアプリケーションで判断する必要があります。この番号が一致しなかった場合、あるいはアプリケーションでこの番号の一致を確認しなかった場合は、各プロパティの値が正しいかどうかは保証できません。

全ての専用メソッドはAsyncModeプロパティの値に従い、同期的、もしくは非同期的に実行されます。 AsyncModeがFALSEなら、各承認のメソッドは同期的に実行され、メソッド終了時にはプロパティは確定します。 AsyncModeがTRUEなら、各承認メソッドは即時に終了しアプリケーションに制御が移ります。 承認結果確定すれば各プロパティを設定し、その後OutputCompleteEventで通知します。 AsyncModeがTRUEの場合でもメソッド発行後直ちに次のメソッドを発行するような多重発行はできません。ただしClearOutputは先行して発行された非同期メソッドを取消目的で実行されますので例外です。

各承認メソッド、および結果のプロパティは各メーカーのCATデバイスとCATコントロールにより異なるため、Capのプロパティを必ず確認の上メソッドの実行、あるいはプロパティの参照を行ってください。

一般的なOLEの実行結果は同期の場合はResultCodeに、非同期の場合は終了通知がOutputCompleteEventで、あるいはErrorEventの引数*ResultCode*に返されます。 プロパティのResultCodeまたは引数の*ResultCode*がOPOS_E_EXTENDEDである場合、CATデバイス固有のエラー情報は同期の場合はResultCodeExtendedに、非同期の場合はErrorEventの引数*ResultCodeExtended*に詳細なコードが格納されます。 またセンターからの結果コードは同期、非同期いずれの場合でもCenterResultCodeに格納されます。

各メソッドを発行する前にTrainingModeをTRUEにすることによりトレーニングモードでの実行となります。

非同期メソッド発行後に、その非同期メソッドを取消すのはClearOutputメソッドを発行することにより可能です。

認証ログ取得はAccessDailyLogメソッドにより取得できます。 取得はAsyncModeプロパティの値に従い同期的、もしくは非同期的に実行されます。 AsyncModeがFALSEなら、各承認のメソッドは同期的に実行され、メソッド終了直前にDailyLogプロパティに格納されます。AsyncModeがTRUEなら、認証ログ取得メソッドは即時に終了しアプリケーションに制御が移ります。 認証ログをDailyLogプロパティに格納し、認証ログ取得終了をOutputCompleteEventで通知します。 非同期でAccessDailyLogを実行中に他のメソッドは発行できません。ただしClearOutputメソッドは例外的に発行することができます。

一般的な使用例は以下の通りです。

同期動作

Open

ClaimDevice

DeviceEnabled=TRUE

引数の*SequenceNumber*の設定

PaymentMediaの設定　　第１.５版で追加

AuthorizeSales()

ResultCodeのチェック

プロパティの**SequenceNumber**との一致確認

AuthorizeSales()によりセットされたプロパティの参照

DeviceEnabled=FALSE

ReleaseDevice

Close

非同期動作

Open

ClaimDevice

DeviceEnabled=TRUE

AsyncMode=TRUE

引数の*SequenceNumber*の設定

PaymentMediaの設定　　第１.５版で追加

AuthorizeSales()

ResultCodeのチェック

OutputCompleteEvent待ち

引数の*ResultCode*のチェック

プロパティの**SequenceNumber**との一致確認

AuthorizeSales()によりセットされたプロパティの参照

DeviceEnabled=FALSE

ReleaseDevice

Close

デバイスの共有

CATは、以下に示すような排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に、排他アクセス権を獲得しなければなりません。

アプリケーションは、プロパティを参照、あるいは、デバイスを更新するメソッドを呼び出す前には、デバイスの排他アクセス権を獲得し、イネーブルにしなければなりません。

使用上の必要条件に関しては、「**一覧**」を参照してください。

プロパティ

AccountNumber プロパティ　　第1.9版で更新

形式　BSTR AccountNumber;

説明　承認結果としての「会員番号」または「口座番号」を通知します。

電子マネー決済デバイスでは、決済されたカード番号を通知します。

このプロパティは、Openメソッドにより空の文字列に初期化されます。

AdditionalSecurityInformationプロパティ　R/W

形式　BSTR AdditionalSecurityInformation;

説明　承認要求メソッド発行前に、このプロパティに任意のデータをセットすることにより、任意のデータをCATデバイスへ送信できます。 また承認結果として、CATデバイスから送られたデータで他のプロパティに格納されないデータ（例：ポイント用の会員コード）をこのプロパティに格納することにより、アプリケーションへ渡すことが可能です。　ここに格納されるデータはCATデバイス、CATコントロールやカードに依存するため、このプロパティは移植性を考慮した開発においては極力使用を避けるべきです。このデータのフォーマットはBinaryConversionプロパティの値に依存します。詳細はBinaryConversionプロパティを参照してください。

このプロパティは、Openメソッドにより空の文字列に初期化されます。

参照　CapAdditionalSecurityInformationプロパティ、 BinaryConversionプロパティ

ApprovalCode プロパティ 第1.9版で更新

形式 BSTR ApprovalCode;

説明 承認結果としての「承認番号」を通知します。

電子マネー決済デバイスでは、決済結果としての「決済番号」を通知します。

このプロパティは、**Open**メソッドにより空の文字列に初期化されます。

AsyncMode プロパティ R/W

形式 BOOL AsyncMode;

説明 TRUE: 全専用メソッドが非同期的に実行されます。

FALSE: 同期的に実行されます。

このプロパティは、**Open**メソッドによりFALSEに初期化されます。

戻り値 次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティが正常にセットされました。

参照 **AuthorizeXxx**メソッド、**CardCheck**メソッド、**AccessDailyLog**メソッド

Ballance プロパティ 第1.9版で追加

形式 CURRENCY Balance;

説明 電子マネー決済デバイスでは、残高が設定されます。

このプロパティは、**Open**メソッドにより初期化されます。

CapAdditionalSecurityInformationプロパティ

形式　BOOL CapAdditionalSecurityInformation;

説明　TRUE: AdditionalSecurityInformationプロパティは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　AdditionalSecurityInformationプロパティ

CapAuthorizeCompletionプロパティ

形式　BOOL CapAuthorizeCompletion;

説明　TRUE: AuthorizeCompletionメソッドは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　AuthorizeCompletionメソッド

CapAuthorizePreSalesプロパティ

形式　BOOL CapAuthorizePreSales;

説明　TRUE: AuthorizePreSalesメソッドは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　AuthorizePreSalesメソッド

CapAuthorizeRefundプロパティ

形式　　BOOL CapAuthorizeRefund;

説明　　TRUE: **AuthorizeRefund**メソッドは実装されています。

FALSE:サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**AuthorizeRefund**メソッド

CapAuthorizeVoidプロパティ

形式　　BOOL CapAuthorizeVoid;

説明　　TRUE: **AuthorizeVoid**メソッドは実装されています。

FALSE:サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**AuthorizeVoid**メソッド

CapAuthorizeVoidPreSalesプロパティ

形式　BOOL CapAuthorizeVoidPreSales;

説明　TRUE: AuthorizeVoidPreSalesメソッドは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　AuthorizeVoidPreSalesメソッド

CapCashDepositプロパティ　第1.9版で追加

形式　BOOL CapCashDeposit;

説明　デバイスがCashDepositメソッドによるチャージ機能を有するかどうかを示します。
TRUE: CashDepositメソッドは実装されています。
FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　CashDepositメソッド

CapCenterResultCodeプロパティ

形式　BOOL CapCenterResultCode;

説明　TRUE:CenterResultCodeプロパティは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　CenterResultCodeプロパティ

CapCheckCardプロパティ

形式　BOOL CapCheckCard;

説明　TRUE:**CheckCard**メソッドは実装されています。

FALSE:サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CheckCard**メソッド

CapDailyLogプロパティ

形式　LONG CapDailyLog;

説明　デバイスの認証ログ能力を示します

値	意味
CAT_DL_NONE	このデバイスは認証ログ機能を一切持ちません。
CAT_DL_REPORTING	このデバイスは中間計の機能のみ持ちます。中間計とは認証ログの読み出しは行いますが、認証ログの消去（精算）は行いません。
CAT_DL_SETTLEMENT	このデバイスは精算機能のみ持ちます。 精算とは認証ログの読み出しと、認証ログの消去を行います。
CAT_DL_ REPORTING_ SETTLEMENT	このデバイスは中間計と精算機能の両方を持ちます。

このプロパティは**Open**メソッドにより初期化されます。

参照　**DailyLog**プロパティ、**AccessDailyLog**メソッド

CapInstallmentsプロパティ

形式　BOOL CapInstallments;

説明　TRUE: AccessDailyLogの結果としてDailyLogプロパティに格納される項目のInstallmentsは実装されています。

FALSE: サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　**DailyLog**プロパティ

CapLockTerminalプロパティ　第1.9版で追加

形式　BOOL CapLockTerminal;

説明　デバイスがセキュリティロック機能を持ち、LockTerminalメソッドによりロックできるか否かを示します。

このプロパティは、Openメソッドにより初期化されます。

参照　**LockTerminal**プロパティ

CapLogStatusプロパティ　第1.9版で追加

形式　BOOL CapLogStatus;

説明　デバイスがLogStatusプロパティにより、ログの状態を通知できるか否かを示します。

このプロパティは、Openメソッドにより初期化されます。

参照　**LogStatus**プロパティ

CapPaymentDetailプロパティ

形式　BOOL CapPaymentDetail;

説明　TRUE: **PaymentDetail**プロパティは実装されています。

FALSE:サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**PaymentDetail**プロパティ

CapTaxOthersプロパティ

形式　BOOL CapTaxOthers;

説明　TRUE: **AccessDailyLog**の結果として**DailyLog**プロパティに格納される項目のTaxOthersは実装されています。 このプロパティがFALSEの場合でも承認系メソッドの引数"*TaxOthers*"にはダミーで“０”をセットする必要があります。

FALSE: サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**DailyLog**プロパティ

CapTransactionNumberプロパティ

形式　BOOL CapTransactionNumber;

説明　TRUE: **TransactionNumber**プロパティは実装されています。

FALSE:サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**TransactionNumber**プロパティ

CapTrainingModeプロパティ

形式　BOOL CapTrainingMode;

説明　TRUE: TrainingModeプロパティは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照 TrainingModeプロパティ

CapUnlockTerminalプロパティ　第1.9版で追加

形式　BOOL CapUnlockTerminal;

説明　デバイスがセキュリティロック機能を持ち、UnlockTerminalメソッドによりロックを解除できるか否かを示します。

このプロパティは、Openメソッドにより初期化されます。

参照 UnlockTerminalプロパティ

CardCompanyID プロパティ

形式　BSTR CardCompanyID;

説明　承認結果としての「カード会社ID」または「金融機関コード（支店コード含む）」を通知します。 桁数はCATデバイスに依存します。

このプロパティは、Openメソッドにより空の文字列に初期化されます。

CenterResultCodeプロパティ

形式　　BSTR CenterResultCode;

説明　　各承認メソッドの結果として「承認機関からのコード」を格納します。 実際格納されるコードは承認機関に確認してください。

このプロパティは、**Open**メソッドにより空の文字列に初期化されます。

DailyLogプロパティ　　第1.16.1版で更新

形式　　BSTR DailyLog;

説明　　**AccessDailyLog**メソッドの結果を格納します。データは１取引ごとにCR（13）＋LF（１０）で区切られて、各値はASCIIコードで格納されます。 各取引の詳細なデータは“,”（４４）で区切られています。

このプロパティは、**Open**メソッドにより空の文字列に初期化されます。

１取引データの詳細は以下順番の通りです。 また対応するプロパティと**Cap**プロパティも併記します。

順番	項目	対応するプロパティ	対応する**Cap**プロパティ
1	カード会社ID/ 金融機関コード（支店コード含む）	**CardCompanyID**	なし
２	取引種別	**TransactionType**	なし
３	センター処理日時 注1）	なし	なし
４	センター処理通番 注３）	**TransactionNumber**	**CapTransactionNumber**
５	支払種別	**PaymentCondition**	なし
６	伝票番号	**SlipNumber**	なし
７	センター承認番号	**ApprovalCode**	なし
８	利用日 注１、注５）	なし	なし
９	会員番号/口座番号	**AccountNumber**	なし
１０	金額 注４)	**Authorize**系メソッドの引数の*Amount*または実際に承認された金額	なし
１１	税･その他 注３）	**Authorize**系メソッドの引数の*TaxOthers*	**CapTaxOthers**
１２	分割回数 注３）	なし	**CapInstallments**
１３	追加データ 注２）	**AdditionalSecurityInformation**	**CapAdditionalSecurityInformation**

注

1）フォーマット

項目	フォーマット
センター処理日時	YYYYMMDDHHMMSS
利用日	MMDD

CATデバイスによっては秒まで内部時計で持っていない場合があります。 その場合、センター処理日時の秒には“００”が書き込まれます。

２）追加データ

CATデバイスがベンダー固有に持っているデータを格納する領域です。 これらを使用することにより項目にないデータをアプリケーションが受取ることが可能になります。 ここに格納されるデータはCATやカードに依存するため、このデータは移植性を考慮した開発においては極力使用を避けるべきです。

３）その該当する**Cap**プロパティがFALSEの場合

CATデバイスによって該当するデータが無く、**Cap**プロパティがFALSEになります。 その場合、その項目が無く次の区切りのコンマがすぐにきます。 例えば「金額」が１２３４円で、「税・その他」が無く、「分割回数」が２の場合「1234,,２」となります。 これにより、**Cap**プロパティに依存せずに、各データの位置が一義的に決まります。

４）金額

項目１１の「税・その他」がある場合(**CapTaxOthers**プロパティがTRUEの場合)でも、ない場合(FALSEの場合)でも常に「税・その他」を含みます。

もし通貨がJPY（日本円）で、実際の金額が12,345円である場合、64ビット整数値は「123450000」となり、文字列表現は「12345.00」または「12345.0000」あるいは「12345」または「123450000」となります。
もし通貨がUSD（米ドル）で、実際の金額が1,234.56ドルである場合、64ビット整数値は「12345600」となり、文字列表現は「1234.56」または「1234.5600」あるいは「12345600」となります。

５）利用日

マニュアル入力による承認後売上の利用日です。

例　　以下に例を示します。

項目	内容	意味
カード会社ID/ 金融機関コード（支店コード含む）	102	JCB
取引種別	**CAT_TRANSACTION_SALES**	売上
センター処理日時	19980116134530	1998年1月16日 13時45分30秒
センター処理通番	123456	123456
支払種別	**CAT_PAYMENT_INSTALLMENT_1**	分割1
伝票番号	12345	12345
センター承認番号	0123456	0123456
利用日	なし	なし
会員番号/口座番号	1234123412341234	1234-1234-1234-1234
金額	JPY（日本円） 12345.00 or 12345.0000 or 12345 or 123450000	12345円
	USD（米ドル） 12345.00 or 12345.0000 or 123450000	12345.00 USD
税・その他	なし	なし
分割回数	2	2回
追加データ	12345678	固有情報

上の例で実際に**DailyLog**に格納されるデータは以下の通りです。

```
102,10,19980116134530,123456,61,12345,0123456,,1234123412341234,12345,,2,12345678[CR][LF]
```

電子マネー決済デバイスでは、締め処理用通信装置を備えていない電子マネー決済デバイスが締め処理を行った結果の取引ログが設定されます。

取引ログの内容がセンター以外に対して暗号化されているデバイスがあるかもしれません。その場合、プロパティにはバイナリデータが設定されるため、**BinaryConversion**プロパティをOPOS_BC_NONE以外の値に設定してアクセスする必要があります。

参照　**CapDailyLog**プロパティ、**BinaryConversion**プロパティ、**AccessDailyLog**メソッド

LogStatusプロパティ　第1.9版で追加

形式　LONG LogStatus;

説明　このプロパティは、デバイスの取引ログの状態を示します。

値	意味
CAT_LOGSTATUS_OK	取引ログは充分な空き容量があります。
CAT_LOGSTATUS_NEARFULL	取引ログはフルに近い状態です。
CAT_LOGSTATUS_FULL	取引ログは一杯です。

このプロパティは初期化された後、デバイスがイネーブルの間は最新の状態となります。

取引ログが一杯になると、デバイスによっては決済処理を行うことが出来なくなるかもしれません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**StatusUpdateEvent**イベント

PaymentConditionプロパティ　　第1.9版で更新

形式　LONG PaymentCondition;

説明　各承認メソッドの結果として「支払種別」を格納します。

このプロパティには、次に示す値の内の一つが設定されます。詳細な支払方法は**PaymentDetail**を参照してください。

値	意味
CAT_PAYMENT_LUMP	一括
CAT_PAYMENT_BONUS_1	ボーナス１
CAT_PAYMENT_BONUS_2	ボーナス２
CAT_PAYMENT_BONUS_3	ボーナス３
CAT_PAYMENT_BONUS_4	ボーナス４
CAT_PAYMENT_BONUS_5	ボーナス５
CAT_PAYMENT_INSTALLMENT_1	分割１
CAT_PAYMENT_INSTALLMENT_2	分割２
CAT_PAYMENT_INSTALLMENT_3	分割３
CAT_PAYMENT_BONUS_COMBINATION_1	ボーナス併用１
CAT_PAYMENT_BONUS_COMBINATION_2	ボーナス併用２
CAT_PAYMENT_BONUS_COMBINATION_3	ボーナス併用３
CAT_PAYMENT_BONUS_COMBINATION_4	ボーナス併用４
CAT_PAYMENT_REVOLVING	リボルビング
CAT_PAYMENT_DEBIT	デビットカード
CAT_PAYMENT_ELECTRONIC_MONEY	電子マネー

このプロパティは、**Open**メソッドによりゼロに初期化されます。

参照　**PaymentDetail**プロパティ

PaymentDetailプロパティ　　第1.16.1版で追加

形式　BSTR PaymentDetail;

説明　各承認メソッドの結果として「支払種別の詳細」を格納します。支払の詳細は各支払種別（ PaymentCondition ）により異なります。 それぞれのデータはASCIIコードで格納され、値は“,”（コンマ）で区切られます。 「ボーナス４」、「ボーナス５」、「分割２」、「ボーナス併用３」および「ボーナス併用４」は最大で６回までで、もし６回以下ならばそれまでのデータのみが格納されます。 “NULL”とはデータが何も格納されないという意味であり、長さゼロの**BSTR**となります。

一括

NULL

ボーナス１

NULL

ボーナス２

ボーナス回数

ボーナス３

ボーナス月

ボーナス４

ボーナス回数、ボーナス１回目月、ボーナス２回目月、ボーナス３回目月、ボーナス4回目月、ボーナス５回目月、ボーナス６回目月　　(最大６回)

ボーナス５

ボーナス回数、ボーナス１回目月、１回目金額、ボーナス２回目月、２回目金額、ボーナス３回目月、３回目金額、ボーナス４回目月、４回目金額、ボーナス５回目月、５回目金額、ボーナス６回目月、６回目金額、　(最大６回)

分割１

開始月、分割回数

分割２

開始月、分割回数、１回目金額、２回目金額、３回目金額、４回目金額、５回目金額、６回目金額(最大６回)

分割３

開始月、分割回数、初回金額

ボーナス併用１

開始月、分割回数

ボーナス併用２

開始月、分割回数、ボーナス金額

ボーナス併用３

開始月、分割回数、ボーナス回数、ボーナス１回目月、ボーナス２回目月、ボーナス３回目月、ボーナス４回目月、ボーナス５回目月、ボーナス６回目月 (最大６回)

ボーナス併用４

開始月、分割回数、ボーナス回数、ボーナス１回目月、ボーナス１回目金額、ボーナス２回目月、ボーナス２回目金額、ボーナス３回目月、ボーナス３回目金額、ボーナス４回目月、ボーナス４回目金額、ボーナス５回目月、ボーナス５回目金額、ボーナス６回目月、ボーナス６回目金額(最大６回)

リボルビング払い

NULL

デビットカード払い

NULL

電子マネー

NULL

これらの支払区分、名称は各CATデバイスによって異なります。 次ページからそのCATデバイス別可能支払区分と用語の一覧を示します。 一部にはOPOS用語と各CATでの用語が異なるため注意が必要です。

このプロパティは、**Open**メソッドにより空の文字列に初期化されます。

CAT別支払区分名称、方法一覧

支払方法	入力項目	コード	CAT名称	CAT(旧CAT)	G-CAT	JET-S	SG-CAT	Master-T
			カード	不特定	不特定	ＪＣＢ	ＶＩＳＡ	ＭＡＳＴＥＲ
			OPOS用語	各社の用語				
一括	（なし）	10	一括	同左	同左	同左	同左	同左
ボーナス	（なし）	21	ボーナス１	同左	同左	同左	同左	同左
	ボーナス回数	22	ボーナス２	同左	同左	同左	同左	同左
	ボーナス月	23	ボーナス３	同左	－	－	同左	同左
	ボーナス回数 ボーナス月(1) ボーナス月(2) ボーナス月(3) ボーナス月(4) ボーナス月(5) ボーナス月(6)	24	ボーナス４	同左	ボーナス３	ボーナス３	同左 （但し、ボーナス月入力は２回まで）	同左
	ボーナス回数 ボーナス月(1) ボーナス金額(1) ボーナス月(2) ボーナス金額(2) ボーナス月(3) ボーナス金額(3) ボーナス月(4) ボーナス金額(4) ボーナス月(5) ボーナス金額(5) ボーナス月(6) ボーナス金額(6)	25	ボーナス５	同左	－	－	－	同左
分割	支払開始月 分割回数	61	分割１	同左	同左	同左	同左	同左

	支払開始月 分割回数 分割金額(1) 分割金額(2) 分割金額(3) 分割金額(4) 分割金額(5) 分割金額(6)	62	分割２	同左	－	－	－	－
	支払開始月 分割回数 初回金額	63	分割３	同左	分割２	分割２	－	分割２
ボ併	支払開始月 分割回数	31	ボーナス 併用１	同左	同左	同左	同左	同左
	支払開始月 分割回数 ボーナス金額	32	ボーナス 併用２	同左	－	－	同左	同左
	支払開始月 分割回数 ボーナス回数 ボーナス月(1) ボーナス月(2) ボーナス月(3) ボーナス月(4) ボーナス月(5) ボーナス月(6)	33	ボーナス 併用３	同左	－	－	同左 （但し、ボーナス月入力は２回まで）	同左

	支払開始月 分割回数 ボーナス回数 ボーナス月(1) ボーナス金額(1) ボーナス月(2) ボーナス金額(2) ボーナス月(3) ボーナス金額(3) ボーナス月(4) ボーナス金額(4) ボーナス月(5) ボーナス金額(5) ボーナス月(6) ボーナス金額(6)	34	ボーナス 併用４	同左	ボーナス 併用２	ボーナス 併用２	同左 （但し、ボーナス月、金額入力は２回まで）	同左
リボ	（なし）	80	リボルビング	同左	同左	同左	同左	同左
デビット	（なし）	110	デビットカード	対応機種のみ	対応機種のみ	対応機種のみ	対応機種のみ	対応機種のみ

凡例：　同左：ＯＰＯＳ用語に同じ

―　：その支払区分が存在しない

参照　　**CapPaymentDetail**プロパティ、**DailyLog**プロパティ

PaymentMediaプロパティR/W　　第１．５版で追加、第1.9版で更新

形式　LONG PaymentMedia;

説明　各承認メソッドの承認すべきメディアを指定します。

このプロパティには、各承認メソッドを発行する前に次に示す値の内の一つを設定します。指定無しとはCATデバイス側で指定することを意味します。

値	意味
CAT_MEDIA_UNSPECIFIED	指定無し
CAT_MEDIA_NONDEFINE	同上（第1.4版以前との互換性確保のため残すがCAT_MEDIA_UNSPECIFIEDを使用すること）
CAT_MEDIA_CREDIT	クレジット
CAT_MEDIA_DEBIT	デビット
CAT_MEDIA_ELECTRONIC_MONEY	電子マネー

このプロパティは、**Open**メソッドによりCAT_MEDIA_NONDEFINEに初期化されます。

SequenceNumberプロパティ

形式　LONG SequenceNumber;

説明　各承認メソッドの結果として「受信シーケンス番号」を格納します。この番号はアプリケーションで承認系メソッドの引数*SequenceNumber*との一致を確認する必要があります。もしCATデバイスからの「受信シーケンス番号」が数値でない場合はゼロが格納されます。

このプロパティは、Openメソッドでゼロに初期化されます。

SettledAmountプロパティ　第1.9版で追加

形式　CURRENCY SettledAmount;

説明　実決済額が設定されます。

このプロパティは、Openメソッドで初期化されます。

参照　AuthorizeSalesメソッド、CashDepositメソッド

SlipNumberプロパティ

形式　BSTR SlipNumber;

説明　各承認メソッドの結果として「伝票番号」を格納します。

このプロパティは、Openメソッドで空の文字列に初期化されます。

TrainingModeプロパティR/W

形式 BOOL TrainingMode;

説明 TRUE: 各承認メソッドが練習モードで実行されます。

FALSE: 各承認メソッドが通常モードで実行されます。

練習モードで実行する場合は各承認メソッドの発行に先立ち、**TrainingMode**プロパティをTRUEにする必要があります。メソッドの実行後も自動的にはFALSEには戻りませんので、練習モードを抜ける場合はアプリケーションで明示的にFALSEに戻す必要があります。

このプロパティは、**Open**メソッドによりFALSEに初期化されます。

戻り値

本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	このプログラムは正常に設定されました。
OPOS_E_ILLEGAL	正常に設定されていません。練習モードはサポートされていません。(**CapTrainingMode**プロパティを参照)

TransactionNumberプロパティ

形式 BSTR TransactionNumber;

説明 各承認メソッドの結果として「センター処理通番」を格納します。

このプロパティは、**Open**メソッドで空の文字列に初期化されます。

TransactionTypeプロパティ　　第1.9版で更新

形式　LONG TransactionType;

説明　各承認メソッドの結果として「取引種別」を格納します。

このプロパティには、次に示す値の内の一つが設定されます。

値	意味
CAT_TRANSACTION_SALES	売上
CAT_TRANSACTION_VOID	取消
CAT_TRANSACTION_REFUND	返品
CAT_TRANSACTION_COMPLETION	承認後売上
CAT_TRANSACTION_PRESALES	事前予約
CAT_TRANSACTION_CHECKCARD	カードチェック
CAT_TRANSACTION_VOIDPRESALES	事前予約の取消
CAT_TRANSACTION_CASHDEPOSIT	チャージ

このプロパティは、**Open**メソッドでゼロに初期化されます。

メソッド

AccessDailyLog メソッド　　第1.9版で追加

形式　LONG AccessDailyLog (LONG *SequenceNumber*, LONG *Type*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	認証ログ取得のためのシーケンス番号
Type	認証ログが中間計か精算かを指定します。
Timeout	CATデバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　認証ログをCATから取得します。

*SequenceNumber*によって指定された処理として認証ログを取得し**DailyLog**に格納します。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、CATから応答があるまで無限に待ちます。

*type*は次に示す値のいずれかを設定し、認証ログのタイプ（中間計か精算のどちらか）を指定します。 指定できるタイプは**CapDailyLog**に従います。

電子マネー決済デバイスでは、認証ログをCATから取得します。

電子マネー決済デバイスの取引ログを締めます。電子マネー決済デバイスが締め処理用通信装置をもっている場合には、取引ログは電子マネー決済端末から直接センターに送信されるでしょう。電子マネー決済デバイスが締め処理用通信装置をもたず、取引ログをPOSアプリケーションによってセンターに送信する必要がある場合には、取引ログは**DailyLog**プロパティに格納されます。

値	意味
CAT_DL_REPORTING	認証ログとして中間計を実行します。
CAT_DL_SETTLEMENT	認証ログとして精算を実行します。 電子マネー決済デバイスの取引ログを締めます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に認証ログ取得処理が行われました。
OPOS_E_ILLEGAL	無効な*Type*あるいは*Timeout*パラメータが指定されました。又は認証ログ取得はサポートされていません。(**CapDailyLog**プロパティを参照)
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、CATからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	CATデバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**CapDailyLog**プロパティ、**DailyLog**プロパティ、**ErrorEvent**イベント

AuthorizeCompletion メソッド

形式　LONG AuthorizeCompletion (LONG *SequenceNumber*, CURRENCY *Amount*, CURRENCY *TaxOthers*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	承認のためのシーケンス番号
Amount	承認のための売上金額（税、その他を含まない金額）
TaxOthers	承認のための税、その他金額
Timeout	CATデバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　承認後売上を行います。

*SequenceNumber*によって指定された承認として、 *Amount*、*TaxOthers*、の承認後売上を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、CATから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に承認処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又は承認後売上はサポートされていません。(**CapAuthorizeCompletion**プロパティ参照)
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、CATからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	CATデバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**CapAuthorizeCompletion**プロパティ、**ErrorEvent**イベント

AuthorizePreSales メソッド

形式　LONG AuthorizePreSales (LONG *SequenceNumber*, CURRENCY *Amount*, CURRENCY *TaxOthers*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	承認のためのシーケンス番号
Amount	承認のための売上金額（税、その他を含まない金額）
TaxOthers	承認のための税、その他金額
Timeout	CATデバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　事前予約を行います。

*SequenceNumber*によって指定された承認として、 *Amount*、*TaxOthers*、の事前予約を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、CATから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に承認処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又は事前予約はサポートされていません。(**CapAuthorizePreSales**プロパティ参照)
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、CATからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	CATデバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**CapAuthorizePreSales**プロパティ、**ErrorEvent**イベント

AuthorizeRefund メソッド

形式　LONG AuthorizeRefund (LONG *SequenceNumber*, CURRENCY *Amount*, CURRENCY *TaxOthers*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	承認のためのシーケンス番号
Amount	承認のための売上金額（税、その他を含まない金額）
TaxOthers	承認のための税、その他金額
Timeout	CATデバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　返品承認を行います。

*SequenceNumber*によって指定された承認として、 *Amount*、*TaxOthers*、の返品承認を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、CATから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に承認処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又は返品承認はサポートされていません。(**CapAuthorizeRefund**プロパティ参照)
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、CATからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	CATデバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**CapAuthorize**Refundプロパティ、**ErrorEvent**イベント

AuthorizeSales メソッド

形式　LONG AuthorizeSales (LONG *SequenceNumber*, CURRENCY *Amount*, CURRENCY *TaxOthers*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	承認のためのシーケンス番号
Amount	承認のための売上金額（税、その他を含まない金額）
TaxOthers	承認のための税、その他金額
Timeout	CATデバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　通常売上承認を行います。

*SequenceNumber*によって指定された承認として、 *Amount*、*TaxOthers*、の通常売上承認を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、CATから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に承認処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、CATからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	CATデバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**ErrorEvent**イベント

AuthorizeVoid メソッド

形式　LONG AuthorizeVoid (LONG *SequenceNumber*, CURRENCY *Amount*, CURRENCY *TaxOthers*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	承認のためのシーケンス番号
Amount	承認のための売上金額（税、その他を含まない金額）
TaxOthers	承認のための税、その他金額
Timeout	CATデバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　売上取消の承認を行います。

*SequenceNumber*によって指定された承認として、 *Amount*、*TaxOthers*、の取消承認を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、CATから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に承認処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又は売上取消の承認はサポートされていません。(**CapAuthorizeVoid**プロパティ参照)
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、CATからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	CATデバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**CapAuthorize**Voidプロパティ、ErrorEventイベント

AuthorizeVoidPreSales メソッド

形式　LONG AuthorizeVoidPreSales (LONG *SequenceNumber*, CURRENCY *Amount*, CURRENCY *TaxOthers*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	承認のためのシーケンス番号
Amount	承認のための売上金額（税、その他を含まない金額）
TaxOthers	承認のための税、その他金額
Timeout	CATデバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　事前予約の取消承認を行います。

*SequenceNumber*によって指定された承認として、 *Amount*、*TaxOthers*、の事前予約の取消承認を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、CATから応答があるまで無限に待ちます。

事前予約の取消承認が実装されていないCATコントロール／CATデバイスでは通常の取消で事前予約の取消承認ができる場合があります。各CATコントロール／CATデバイスのマニュアルを参照してください。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に承認処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又は事前予約の取消承認はサポートされていません。(**CapAuthorizeVoidPreSales**プロパティ参照)
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、CATからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。

OPOS_E_BUSY	CATデバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**CapAuthorizeVoidPreSales**プロパティ、**ErrorEvent**イベント

CashDeposit メソッド　　第1.9版で追加

形式　LONG CashDeposit (LONG *SequenceNumber*, CURRENCY *Amount*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	チャージのためのシーケンス番号
Amount	チャージ金額
Timeout	デバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　チャージを行います。

*SequenceNumber*によって指定された処理として、 *Amount* 額をチャージします。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、CATから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

CheckCard メソッド　　第1.9版で更新

形式　LONG CheckCard (LONG *SequenceNumber*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	承認のためのシーケンス番号
Timeout	CATデバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　カードチェックを行います。

*SequenceNumber*によって指定された処理としてカードチェックを行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、CATから応答があるまで無限に待ちます。

電子マネー決済デバイスでは、電子マネー決済デバイスに対して、残高照会を要求します。

残高は、**Balance**プロパティに設定されます。

残高照会の結果、残高以外の情報をデバイスが返すことができる場合には、決済処理の結果と同様に、それぞれのプロパティに情報が設定されるでしょう。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に承認処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又はカードチェックはサポートされていません。(**CapCheckCard**プロパティ参照)
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、CATからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。

OPOS_E_BUSY	CATデバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**CapCheckCard**プロパティ、ErrorEventイベント

LockTerminal メソッド　　第1.9版で追加

形式　LONG LockTerminal ();

説明　電子マネー決済デバイスをロックし、使用できないようにします。

電子マネー決済デバイスがセキュリティロック機能を持つ場合にサポートされます。

ロックするために鍵となる情報が必要な場合には、**AdditionalSecurityInformation**プロパティに設定します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	電子マネー決済デバイスはセキュリティロック機能を持ちません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**CapLockTerminal**プロパティ

UnlockTerminal メソッド　　第1.9版で追加

形式　LONG UnlockTerminal ();

説明　ロックを解除し、電子マネー決済デバイスを使用できるようにします。

電子マネー決済デバイスがセキュリティロック機能を持つ場合にサポートされます。

ロックを解除するために鍵となる情報が必要な場合には、**AdditionalSecurityInformation**プロパティに設定します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	電子マネー決済デバイスはセキュリティロック機能を持ちません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**CapUnlockTerminal**プロパティ

イベント

ErrorEvent イベント　　第1.9版で更新

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

パラメータ	説明
ResultCode	エラーイベントを起こした要因のコード。 値はResultCodeの項目を参照。
ResultCodeExtended	エラーイベントを起こした要因の拡張コード。値は下記の値を参照。
ErrorLocus	OPOS_EL_OUTPUTに設定します。 非同期動作中に、エラーが起りました。
pErrorResponse	エラーイベント応答へのポインタです。 下記の値を参照。

もし,*ResultCode*がOPOS_E_EXTENDEDであれば、*ResultCodeExtended*は次に示す値のいずれかに設定されます。

値	意味
OPOS_ECAT_CENTERERROR	承認機関（センター）からエラーが返されました。 詳細なエラーコードはCenterResultCodeに設定されています。
OPOS_ECAT_COMMANDERROR	CATに送信されたコマンドが間違っています。CATコントロールが正常に動作している限りは、このエラーが返されることはありません。
OPOS_ECAT_RESET	CATが処理中にCATのリセットキー（中断キー）等で処理が中断されました。
OPOS_ECAT_COMMUNICATIONERROR	CATと承認機関（センター）間で通信障害が発生しました。
OPOS_ECAT_DAILYLOGOVERFLOW	認証ログが大きすぎて格納できませんでした。認証ログ取得は中断され、DailyLogプロパティの値は不定です。

OPOS_ECAT_DAILYLOGOVERFLOW
電子マネー決済を試みましたが、デバイスの取引ログが一杯なため、決済できませんでした。

OPOS_ECAT_DEFICIENT
残高が不足しているため、決済できませんでした。

OPOS_ECAT_OVERDEPOSIT
チャージ可能金額を超えるため、チャージできませんでした。

*pErrorResponse*によって指定される位置の内容は、OPOS_ER_RETRYのデフォルト値にプリセットされます。アプリケーションは、次に示す値の内の一つを設定します。

値	意味
OPOS_ER_RETRY	その非同期処理を再度試みます。 エラー状態は抜けます。
OPOS_ER_CLEAR	その他非同期処理を削除します。 エラー状態は抜けます。

説明　非同期で**Authorize**系メソッドおよび**AccessDailyLog**メソッド実行時にエラーを察知し、CATコントロールの状態がエラー状態に移った時に通知します。

参照　**「ステータス、結果コード、状態モデル」**

StatusUpdateEvent イベント　　*第1.9 版で更新*

形式　　void StatusUpdateEvent (LONG *Status*);

*Status*パラメータは、デバイスの状態が設定されます。

値	意味
CAT_LOGSTATUS_OK	取引ログは充分な空き容量があります。
CAT_LOGSTATUS_NEARFULL	取引ログはフルに近い状態です。
CAT_LOGSTATUS_FULL	取引ログは一杯です。
*電源通知の**StatusUpdateEvent**の値*	「第2章 共通プロパティ、メソッド、イベント」の**StatusUpdateEvent**の記述を参照。

説明　　デバイスの状態が変化した時に通知されます。

CapLogStatusプロパティがTRUEの場合、取引ログの状態が通知されます。

参照　　**CapLogStatus**プロパティ、**LogStatus**プロパティ

第 22 章
ポイントカード機

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.5	Boolean	R/W	適用外
BinaryConversion	1.5	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.5	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.5	String	R	Open
Claimed	1.5	Boolean	R	Open
DataCount	1.5	Long	R	Open
DataEventEnabled	1.5	Boolean	R/W	Open
DeviceEnabled	1.5	Boolean	R/W	Open & Claim
FreezeEvents	1.5	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.5	Long	R	Open
PowerNotify	1.5	Long	R/W	Open
PowerState	1.5	Long	R	Open
ResultCode	1.5	Long	R	無し
ResultCodeExtended	1.5	Long	R	Open
State	1.5	Long	R	無し
ControlObjectDescription	1.5	String	R	無し
ControlObjectVersion	1.5	Long	R	無し
ServiceObjectDescription	1.5	String	R	Open
ServiceObjectVersion	1.5	Long	R	Open
DeviceDescription	1.5	String	R	Open
DeviceName	1.5	String	R	Open

専用

CapBold	1.5	Boolean	R	Open
CapCardEntranceSensor	1.5	Boolean	R	Open
CapCharacterSet	1.5	Long	R	Open
CapCleanCard	1.5	Boolean	R	Open
CapClearPrint	1.5	Boolean	R	Open
CapDhigh	1.5	Boolean	R	Open
CapDwide	1.5	Boolean	R	Open
CapDwideDhigh	1.5	Boolean	R	Open
CapItalic	1.5	Boolean	R	Open
CapLeft90	1.5	Boolean	R	Open
CapMapCaracterSet	1.7	Boolean	R	Open
CapPrint	1.5	Boolean	R	Open
CapPrintMode	1.5	Boolean	R	Open
CapRight90	1.5	Boolean	R	Open
CapRotate180	1.5	Boolean	R	Open
CapTracksToRead	1.5	Long	R	Open
CapTracksToWrite	1.5	Long	R	Open
CardState	1.5	Long	R	Open
CharacterSet	1.5	Long	R/W	Open,Claim,& Enable
CharacterSetList	1.5	String	R	Open
FontTypeFaceList	1.5	String	R	Open
LineChars	1.5	Long	R/W	Open,Claim,& Enable
LineCharsList	1.5	Long	R	Open
LineHeight	1.5	Long	R/W	Open,Claim,& Enable
LineSpacing	1.5	Long	R/W	Open,Claim,& Enable
LineWidth	1.5	Long	R	Open,Claim,& Enable
MapCharacterSet	1.7	Boolean	R/W	Open
MapMode	1.5	Long	R/W	Open,Claim,& Enable
MaxLine	1.5	Long	R	Open,Claim,& Enable
PrintHeight	1.5	Long	R	Open,Claim,& Enable
ReadState1	1.5	Long	R	Open
ReadState2	1.5	Long	R	Open
RecvLength1	1.5	Long	R	Open,Claim,& Enable
RecvLength2	1.5	Long	R	Open,Claim,& Enable
SidewaysMaxChars	1.5	Long	R	Open
SidewaysMaxLines	1.5	Long	R	Open
TracksToRead	1.5	Long	R/W	Open,Claim,& Enable
TracksToWrite	1.5	Long	R/W	Open,Claim,& Enable

Track1Data	1.5	String	R	Open
Track2Data	1.5	String	R	Open
Track3Data	1.5	String	R	Open
Track4Data	1.5	String	R	Open
Track5Data	1.5	String	R	Open
Track6Data	1.5	String	R	Open
WriteState1	1.5	Long	R	Open
WriteState2	1.5	Long	R	Open
Write1Data	1.5	String	R/W	Open
Write2Data	1.5	String	R/W	Open
Write3Data	1.5	String	R/W	Open
Write4Data	1.5	String	R/W	Open
Write5Data	1.5	String	R/W	Open
Write6Data	1.5	String	R/W	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.5	無し
Close	1.5	Open
ClaimDevice	1.5	Open
ReleaseDevice	1.5	Open & Claim
CheckHealth	1.5	Open, Claim, & Enable
ClearInput	1.5	Open & Claim
ClearInputProperties	1.10	Open & Claim
ClearOutput	1.5	Open & Claim
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.5	Open & Claim
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用		
BeginInsertion	1.5	Open, Claim,＆Enable
BeginRemoval	1.5	Open, Claime,＆Enable
CleanCard	1.5	Open, Claim,＆Enable
ClearPrintWrite	1.5	Open, Claim,＆Enable
EndInsertion	1.5	Open, Claim,＆Enable
EndRemoval	1.5	Open, Claim,＆Enable
PrintWrite	1.5	Open, Claim,＆Enable
RotatePrint	1.5	Open, Claim,＆Enable
ValidateData	1.5	Open, Claim,＆Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.5	Open,Claim,&Enable
DirectIOEvent	1.5	Open & Claim
ErrorEvent	1.5	Open,Claim,&Enable
OutputCompleteEvent	1.5	Open,Claim,&Enable
StatusUpdateEvent	1.5	Open,Claim,&Enable

概説

ポイントカード機コントロールのOLEプログラマティックIDは“OPOS.PointCardRW”です。

このデバイスは、第1.５版で追加されました。

機能

ポイントカード機コントロールは次の機能を持っています。

磁気ストライプからコード化されたデータの読取と書込みが可能です。データはデバイスの機能に対応してトラック1から６までの読取と書込みができます。

磁気データは、デバイスにより指定されたデータフォーマットが読み込まれます。通常は、アスキーコードが使用されますが、フォーマットについては、各デバイスのマニュアルを参照してください。さらに、読込まれたデータは、各トラックに対応したプロパティに格納されます。

印字エリアを持たないタイプと、サーマル印字／消去機能を持つタイプをサポートします。印字方法は、デバイスの機能に依存します。

ポイントカード機は、カードの挿入と排出の機能を有します。

特別なセキュリティ機能は有りません。

モデル

ポイントカード機の一般的なモデルを示します。

磁気ストライプ面の読み込みは、カードが持つ全てのトラックに対して行います。読込まれたのデータの長さと結果をトラックに対応したプロパティにセットします。

ポイントカード機は、カード挿入処理において、イベント駆動入力の一般的な入力モデルに従います。また、印字エリアと磁気ストライプ面への書込み時には、一般的な出力モデルに従います。

入力モデル

ポイントカード機コントロールは、Openメソッド、ClaimDeviceメソッドを実行し、DeviceEnabledをTRUEにすることで使用します。

アプリケーションがポイントカード機に、カードを挿入させたい場合、アプリケーションは、タイムアウト値と共にBeginInsertionメソッドを呼び出します。このメソッドにより、デバイスにポイントカードが挿入可能となります。タイムアウト値以内に、ポイントカードが挿入されるとコントロールは、正常終了を返します。

タイムアウトの設定値を過ぎても、ポイントカードがポイントカード機に挿入されない時、OPOSコントロールは、タイムアウトエラーをアプリケーションに返します。

タイムアウトエラー発生後もポイントカード機は、ポイントカードの挿入可能な状態を維持します。ポイントカード機の機能によって、カードの挿入を通知するものもありますが、アプリケーションは、ポイントカードの挿入を確認するため、再度BeginInsertionを実行します。

ポイントカードがポイントカード機に挿入されるか、挿入を中止する場合、アプリケーションはEndInsertionメソッドを呼び出します。

これによりポイントカード機は、ポイントカード機の読み込み状態を終了します。

ポイントカードがポイントカード機に存在していればカードデータの読込みが可能になります。アプリケーションがDataEventEnabledプロパティをTRUEに設定すると、DataEventがアプリケーションに通知されます。

入力データの読込み中または処理中に、コントロールにおいてエラーが発生した場合、ErrorEventが発行され、DataEventEnabledプロパティがTRUEならばアプリケーションに通知されます。

DataCountプロパティを参照することでコントロールによってキューイングされたデータイベント数が得られます。

キューイングされたすべての入力データは、ClearInputメソッドを呼び出す事で削除できます。

出力モデル

カードにデータを書き込む場合はPrintWriteメソッドを呼び出します。データの書き込み処理は、デバイスの機能に依存します。PrintWriteメソッドの完了はOutputCompleteEventが通知されることで確認できます。

ポイントカードの処理が終了すると、アプリケーションはポイントカードを排出するため、BeginRemovalメソッドを、タイムアウトの設定値とともに呼び出します。ポイントカードが正常に排出され、排出口から抜き取られると、OPOSコントロールはOPOS_SUCCESSを戻します。ポイントカードが、排出口に残されている場合は、OPOS_E_TIMEOUTを戻します。OPOS_E_TIMEOUTの時、アプリケーションは再度BeginRemovalを実行しポイントカードの抜き取りを確認する必要があります。ポイントカードが排出されると、アプリケーションはEndRemovalメソッドを呼び出し、ポイントカード機の排出モードを終了させます。

ポイントカードへの書き込みは、非同期出力で行います。

PrintWrite：カードにデータの書き込みと印字を一括して行います。

アプリケーションはPrintWriteメソッドを呼び出して書込み処理を開始します。コントロールはこの書込み処理のリクエストをバッファリングし、リクエストの識別子をOutputIDプロパティに設定して、できる限り迅速にアプリケーションへ戻ります。デバイスがリクエストを正常に終了すると、コントロールは、OutputCompleteEventを通知します。このイベントのパラメータは完了したリクエストのOutputIDです。

PrintWriteメソッドはライトエラーなどのハードの問題によるエラーステータスは返しません。書込み処理のエラーは、ErrorEventによってのみ通知されます。次の場合にPrintWriteメソッドの戻り値にエラーステータスが返されます。ポイントカード機が排他アクセス権を獲得していない場合、デバイスが利用可能な状態でない場合、パラメータが不正である場合、要求がキューに追加できない場合です。最初の二つのケースはアプリケーションのエラーです。一方、最後のケースは重大なシステムリソースの例外です。

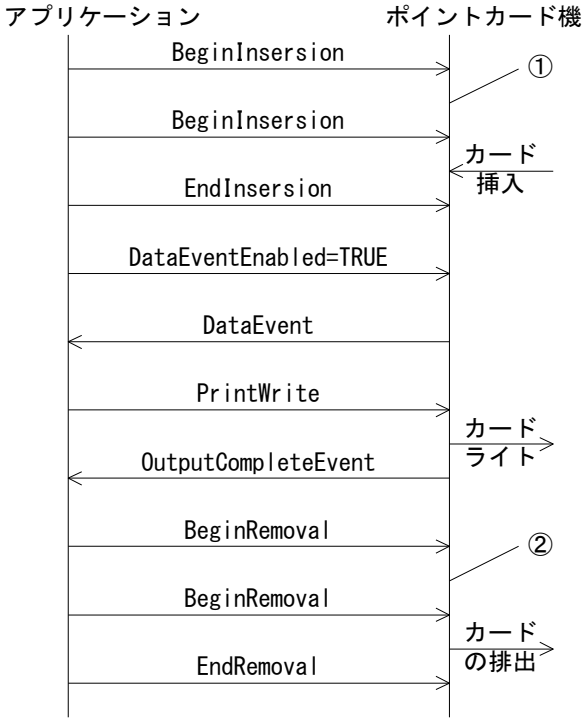
ポイントカード機コントロールは、非同期出力がFIFOで実行されることを保証しています。

ClearOutputメソッドを呼ぶことによって、ポイントカード機コントロールによってバッファリングされたすべての出力は削除されます。クリアした出力に対してOutputCompleteEventは通知されません。このメソッドは（可能ならば）処理中の出力も停止します。

ポイントカードシーケンス図

カードの挿入から排出までの処理を以下に示します。

PrintWriteメソッド以外の全てのメソッドが同期的に動作します。



アプリケーションが指定した時間内にカードが挿入されないと、**Begininsertion**メソッドに対してOPOS_E_TIMEOUTが返されます。アプリケーションは、カードの挿入を確認するため、再度**Begininsertion**メソッドを実行する必要が有ります。この時点で、挿入モードを中断するために、**Endinsertion**メソッドを実行する事も可能です。但し、ポイントカード機にカードが挿入されていないので、コントロールは常にOPOS_E_FAILUREを返します。

挿入口にセンサーを備える場合、排出処理が正常に終了した後に、アプリケーションが指定した時間内にカードが抜き取られないと、**BeginRemoval**メソッドに対してOPOS_E_TIMEOUTが返されます。アプリケーションは、挿入口からカードを取り出すメッセージを表示すると共に、**BeginRemoval**メソッドを再度実行し、カードの抜き取りを確認します。挿入口にセンサーがない場合は、排出処理が正常に終了するとOPOS_SUCCESSを返します。

印字機能について

ポイントカード機の印字機能は、リライト式の印字機能を有するタイプをサポートします。

印字タイプはドットタイプとラインタイプをサポートします。
CapPrintModeがTRUEの時、ドットタイプとして機能します。

印字データは、**PrintWrite**メソッドの*Data*にセットします。

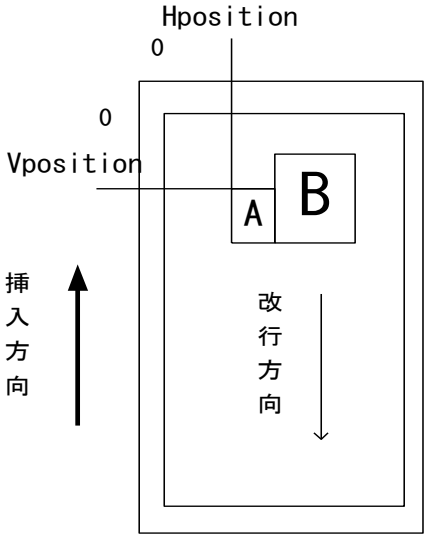
倍角印字等の印字文字の修飾はポイントカード機の機能に依存します。

ドットタイプの印字方法は、印字開始位置を水平方向の*Hposition*と、垂直方向の*Vposition*に**MapMode**の単位でセットします。

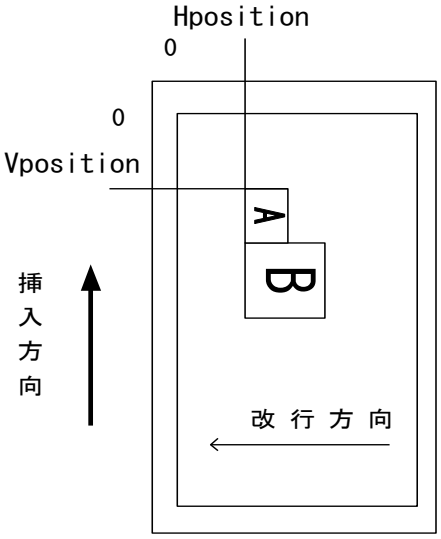
ラインタイプは、ポイントカード機の機能によって印字します。

ドットタイプの印字開始位置の指定は、倍角文字を含む文字列の印字を行う場合に、通常のプリンタのようにプリンタのヘッドが移動する方向を考慮し、印字開始位置を倍角文字に合せて指定するものと、下図に示すように、印字する文字列に倍角文字が含まれることを考慮して、印字開始位置を指定するタイプや、**MapMode**単位の座標位置が挿入方向に対して常左角を基準値に固定しているタイプのものがあります。ポイントカードの印字機能は、各メーカーのデバイスの機能に依存します。

横書き（カード縦長方向のデザイン）



縦書き（カード横長方向のデザイン）



クリーニング機能

ポイントカード機のクリーニングは、ポイントカード機内の汚れによる読み込みエラーや書き込みエラーを防ぐなど、ポイントカード機を安定して稼動させるために必要な機能です。

クリーニングには、クリーニング専用のカードを使用します。クリーニングは、ポイントカード機内に挿入されたクリーニングカードが磁気ヘッドなどを数回往復することで行われます。

使用するクリーニングカードには、エタノールなどを使用する湿式タイプのものと乾式タイプのものが有ります。

ポイントカード機のクリーニング機能は、ポイントカード機で操作する機能として備えられるものと、上位機からのコマンドによりクリーニング操作を実行するものが有ります。

磁気ストライプデータの初期化

ポイントカードの不正使用を防止するため、磁気ストライプ面の初期化を行う機能を有するポイントカード機が有ります。磁気ストライプ面の初期化には、次の方法があります。

スタートマーク、エンドマーク、ＬＲＣ等のすべてのデータを初期化する。

データエリアにアプリケーションが認識できるコードを書き込む。

スタートマークの直後にエンドマークを書き込むことでデータレングスを０にする。

ポイントカードの磁気ストライプ面の初期化は、デバイスの機能に依存します。

デバイスの共有

ポイントカード機は、下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、デバイスの読み込みを始める前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の詳細な必要条件は、「一覧」を参照してください。

印字データとエスケープシーケンス

全てのポイントカード機における印字機能のデフォルト文字設定は、少なくとも0x20から0x7FのASCII文字をサポートすると仮定されます。これは、スペース、数字、大文字、小文字、いくつかの特種な文字を含んでいます。もし、ポイントカード機の印字機能が小文字をサポートしていなければ、サービスオブジェクトは小文字を大文字に変換する事があります。

あらゆるエスケープシーケンスは、後ろに垂直バー（'|'）が続くエスケープ文字ESC（この値は10進法で27）で始まります。この後ろには、0またはより大きな数、小文字のアルファベットが続きます。エスケープシーケンスは、大文字のアルファベットで終わります。

ESC "|" で始まらないエスケープシーケンスは、ポイントカード機にパススルーされます。また、 ESC "|" で始まるがOPOSエスケープシーケンスに当てはまらないエスケープシーケンスも同様にポイントカード機にパススルーされます。

アプリケーションは**ValidateData**メソッドを呼び出す事によって、エスケープシーケンスやデータが、ポイントカード機の印字機能で実行されることができるか判断する事ができます。（いくつかのエスケープシーケンスは、同等なプロパティを使用することによって同じように判断できます。）

以下のエスケープシーケンスが定義されています。もし、エスケープシーケンスがポイントカード機の印字機能によってサポートされないオペレーションを指定すれば、そのエスケープシーケンスは無視されます。

（1）指定時のみ、動作するエスケープシーケンス　第1.7版で更新

名称	データ	内容
埋め込みデータの送信（欄外の備考を参照）	ESC |#E	"#E"に続く文字列は、一切変更されずにデバイスへ渡されます。'#'は、ASCII10進数の文字列で、このエスケープシーケンスに続く、そのままデバイスへ渡されるべき文字列のバイト数を指定します。

備考：このエスケープシーケンスは、第1.7以降でのみ利用可能です。

(2) 印字中に、動作するエスケープシーケンス

明確に変更されるまで、その状態を保つ特徴があります。

名称	データ	内容
フォントタイプ指定	ESC |#fT	以後のデータに対する新しいフォントタイプを選択します。文字'#'の値は、 0 = デフォルトフォントタイプ 1 = **FontTypefaceList**の最初のタイプを選択します。 2 = **FontTypefaceList**の2番目のタイプを選択します。 3以下も同様です。

(3) 印字時に、動作するエスケープシーケンス

各印刷メソッドの終わりや、あるいはノーマルシーケンスによってリセットされる特徴があります。

名称	データ	内容
ボールド	ESC |bC	ボールド体または重ね打ちで印刷します。
アンダーライン	ESC |#uC	アンダーラインと共に印刷します。文字'#'は、アンダーラインの太さ(**MapMode**の単位)を示すASCII10進数の文字列です。もし、'#'が省略されていれば、デフォルトの太さのアンダーラインが使用されます。
イタリック	ESC |iC	イタリック体で印刷します。
反転文字	ESC |rvC	明暗を反対にして印刷します。
縦横1倍角	ESC |1C	通常の大きさで印刷します。
横倍角	ESC |2C	横倍角文字で印刷します。
縦倍角	ESC |3C	縦倍角文字で印刷します。
縦横倍角	ESC |4C	縦横倍角文字で印刷します。
横倍率	ESC |#hC	文字を横方向に拡大して印刷します。文字'#'は、横方向の拡大倍率を示すASCII10進数の文字列です。
縦倍率	ESC |#vC	文字を縦方向に拡大して印刷します。文字'#'は、縦方向の拡大倍率を示すASCII10進数の文字列です。
中央揃え	ESC |cA	中心に以下のテキストを整列させます。
右寄せ	ESC |rA	右に以下のテキストを整列させます。
ノーマル	ESC |N	ポイントカード機の特徴を通常の状態に回復させます。

プロパティ

CapBold プロパティ

形式　　BOOL CapBold;

説明　　TRUE：ポイントカード機はボールド印字属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapCardEntranceSensor プロパティ

形式　　BOOL CapCardEntranceSensor;

説明　　ポイントカード機の挿入口にセンサーが有り無しを示します。

TRUE：ポイントカード機は、挿入口センサーを有します。

FALSE：ポイントカード機はセンサーを有しません。

このプロパティは、Openメソッドにより初期化されます。

CapCharacterSet プロパティ

形式　LONG CapCharacterSet;

説明　ポイントカードに印刷可能な文字設定を示します。

このプロパティには、次に示す値の内の一つが入ります。

値	意味
PCRW_CCS_ALPHA	文字設定は、大文字のアルファベットと、数値、空白、マイナス、ピリオドをサポートします。
PCRW_CCS_ASCII	文字設定は、0x20と0x7F間の全てのASCII文字をサポートします。
PCRW_CCS_KANA	文字設定は、コードページ932の一部をサポートします。それは、0xA1と0xDFの間の半角カタカナと、0x20と0x7F間の全てのASCII文字を含んでいますが、漢字は含まれていません。
PCRW_CCS_KANJI	文字設定は、コードページ932をサポートします。それは、0xA1と0xDFの間の半角カタカナ、0x20と0x7F間の全てのASCII文字をサポートします。また、JIS第一水準、JIS第二水準レベルで定義される文字のシフトJISコード文字をサポートします。
PCRW_CCS_UNICODE	文字設定は、UNICODEをサポートします。

デフォルトの文字設定は、これらの範囲の上位集合を含みます。初めの**CharacterSet**プロパティは、付加的な情報のために調べられます。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CharacterSet**プロパティ

CapCleanCard プロパティ

形式　BOOL CapCleanCard;

説明　TRUE：上位機からのコマンドで磁気ヘッドなどのクリーニングを行う機能をサポートします。
FALSE：サポートしていません。

このプロパティは、Openメソッドにより初期化されます。

CapClearPrint プロパティ

形式　BOOL CapClearPrint;

説明　TRUE：範囲指定して印字エリアをクリアできる機能を有します。
FALSE：一括して印字エリアをクリアします。

このプロパティは、Openメソッドにより初期化されます。

CapDhigh プロパティ

形式　BOOL CapDhigh;

説明　TRUE：ポイントカード機は縦倍角印字属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapDwide プロパティ

形式　BOOL CapDwide;

説明　TRUE：ポイントカード機は横倍角印字属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapDwideDhigh プロパティ

形式　　BOOL CapDwideDhigh;

説明　　TRUE：ポイントカード機は縦横倍角印字属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapItalic プロパティ

形式　　BOOL CapItalic;

説明　　TRUE：ポイントカード機はイタリック印字属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapLeft90 プロパティ

形式　　BOOL CapLeft90;

説明　　TRUE：ポイントカード機は左90度回転印字属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapMapCaracterSet プロパティ　　*第1.7版にて追加*

形式　　BOOL CapMapCharacterSet;

説明　　印字の際にアプリケーションの文字を選択した文字セットにマップするサービスオブジェクトの機能を示します。
TRUE：サービスオブジェクトは文字をCharacterSetListプロパティに定義された文字セットにマッピングすることが可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

参照　　CharacterSetプロパティ、MapCharacterSetプロパティ、CharacterSetListプロパティ

CapPrint プロパティ

形式　　BOOL CapPrint;

説明　　TRUE：印字機能を有します。
FALSE：サポートしません。

このプロパティは、Openメソッドにより初期化されます。

CapPrintMode プロパティ

形式　　BOOL CapPrintMode;

説明　　TRUE：印字開始位置をMapModeの単位で指定可能です。
FALSE：指定できません。

このプロパティは、Openメソッドにより初期化されます。

CapRight90 プロパティ

形式　　BOOL CapRight90;

説明　　TRUE：ポイントカード機は右90度回転印字属性が可能です。
FALSE：不可能です。

このプロパティは、Openメソッドにより初期化されます。

CapRotate180 プロパティ

形式　BOOL CapRotate180;

説明　TRUE：ポイントカード機は180度回転印字属性が可能です。
FALSE：不可能です。

このプロパティは、**Open**メソッドにより初期化されます。

CapTracksToRead プロパティ

形式　LONG CapTracksToRead;

説明　挿入されたポイントカードのアクセス可能な磁気トラックの番号を示します。アクセス可能なトラックはPCRW_TRACK1からPCRW_TRACK6で表します。

例えば、プロパティの値がPCRW_TRACK1のとき、トラック１へのアクセスが可能です。

このプロパティは、**Open**メソッドにより初期化されます。

値	意味
PCRW_TRACK1	Track1
PCRW_TRACK2	Track2
PCRW_TRACK3	Track3
PCRW_TRACK4	Track4
PCRW_TRACK5	Track5
PCRW_TRACK6	Track6

CapTracksToWrite プロパティ

形式　　LONG CapTracksToWrite;

説明　　PrintWriteメソッドでアクセス可能な磁気トラックの番号を示します。アクセス可能なトラックはPCRW_TRACK1からPCRW_TRACK6で表します。

例えば、プロパティーの値がPCRW_TRACK1のとき、トラック１へのアクセスが可能です。

このプロパティは、Openメソッドにより初期化されます。

値	意味
PCRW_TRACK1	Track1
PCRW_TRACK2	Track2
PCRW_TRACK3	Track3
PCRW_TRACK4	Track4
PCRW_TRACK5	Track5
PCRW_TRACK6	Track6

CardState プロパティ

形式　LONG CardState;

説明　このプロパティは、CapCardEntranceSensorプロパティがTrueの時、ポイントカード機の状態が設定されます。

値	意味
PCRW_STATE_NOCARD	カードなしあるいはセンサー無しのため位置不明
PCRW_STATE_REMAINING	挿入口カード残留
PCRW_STATE_INRW	装置内カードあり

CapCardEntranceSensorプロパティがFALSEの場合は常にPCRW_STATE_NOCARDがセットされます。

参照　CapCardEntranceSensorプロパティ

CharacterSet プロパティ　R/W

形式　LONG CharacterSet;

説明　文字の印字に用いるキャラクタを設定します。

このプロパティは、Openメソッド後最初にデバイスをイネーブルにした時に初期化されます。

このプロパティには、次に示す値の内の一つが設定されます。

値	意味
Range 101 - 199	デバイス特有のキャラクタを設定します。コードページにもASCII、Windows ANSIにも当てはまりません。
Range 400 - 990	コードページを設定します。標準の内の一つと適合します。
PCRW_CS_UNICODE	UNICODEキャラクタを設定します。この定数値は、997です。
PCRW_CS_ASCII	ASCIIキャラクタを設定します。0x20と0x7Fの間のASCIIキャラクタをサポートします。この定数値は998です。

PCRW_CS_ANSI　PCRW_CS_WINDOWSと同じ値を持っています。

PCRW_CS_WINDOWS　Windows ANSIキャラクタを設定します。この定数値は999です。これは正確には、Windowsのコードページ1252と等しい。

Range 1000 and higher　Windowsのコードページ；標準値の一つと適合します。

このプロパティを使うための実装に依存する追加情報として「キャラクターセットのマッピング」を参照してください。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	不正な値が使用されました。
その他	**ResultCode**の項目を参照してください。

参照　**CharacterSetList**プロパティ

CharacterSetList プロパティ

形式　BSTR CharacterSetList;

説明　キャラクタ設定番号の文字列。

このプロパティは、Openメソッドにより初期化されます。ASCII数字からなる番号設定の文字列はカンマによって区切られます。たとえば文字列が"101,850,999"で設定されれば、デバイス特有のキャラクタとコードページ850とWindows ANSI キャラクタがサポートされます。

参照　CharacterSetプロパティ

FontTypefaceList プロパティ

形式　BSTR FontTypefaceList;

説明　ポイントカード機によりサポートされるフォントやフォントタイプを指定する文字列です。

このプロパティは、Openメソッドにより初期化されます。フォントまたはフォントタイプからなる文字列はカンマによって区切られます。アプリケーションは、フォントタイプ指定のエスケープシーケンス(ESC|#fT)を使うことにより、ポイントカード機に対するフォントまたはフォントタイプを選択します。文字"#"はリストの中のフォントまたはフォントタイプの番号(1、2など)に置き換わります。

日本では、このプロパティは「明朝」フォントと「ゴシック」フォントをしばしば含みます。その他のフォントやフォントタイプは、他の国でサポートされるかもしれません。

空文字列は、デフォルトのフォントタイプだけがサポートされていることを示します。

参照　「印字データとエスケープシーケンス」

LineChars プロパティ　R/W

形式　　LONG LineChars;

説明　　ポイントカードの一行に印字される半角文字の文字数（行文字幅）です。

この値が、サポートされている行文字幅に変われば、文字幅はその指定された値に設定されます。正確な幅がサポートできない場合、サポートされている行文字幅の中で、指定された値より大きく、かつ最も近い値に設定されます。（例えば、文字数36をセットしてポイントカード機がライン毎に30あるいは40のいずれかの文字幅を印刷することができるならば、サービスオブジェクトは文字のサイズ「40」を選んで、それぞれのラインの上に最高36文字をプリントするべきです。）

もし、文字幅がサポートできなければ、エラーが返されます。（例えば、42をセットしてポイントカード機がライン毎に30あるいは40のいずれかのサイズの文字を印刷することができないならば、サービスオブジェクトはリクエストをサポートすることができません。）

LineCharsを設定することにより、**LineWidth**、**LineHeight**、**LineSpacing**も更新されるかもしれません。なぜなら、文字のピッチやフォントが変わるかもしれないからです。

LineCharsの値は、**Open**メソッド後最初にデバイスをイネーブルにした時に、ポイントカード機のデフォルトの行文字幅に初期化されます。

戻り値　　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	不正な行文字幅が指定されました。

参照　　**LineCharsListプロパティ**

LineCharsList プロパティ

形式　BSTR LineCharsList;

説明　ポイントカードによりサポートされる行文字幅を含む文字列です。

このプロパティは、**Open**メソッドにより初期化されます。

文字列は、カンマによって分けられるASCII数字の設定から成ります。たとえば、文字列が"32,36,40"ならば、ポイントカード機は、32文字、36文字、40文字の行幅をサポートします。

参照　LineCharsプロパティ

LineHeight プロパティ　R/W

形式　LONG LineHeight;

説明　ポイントカード印刷の行の高さです。**MapMode**で定義された単位で表記します。

現在の文字の幅と一緒にサポートできる高さに変更されれば、その行の高さがこの値に設定されます。もし、正確な高さがサポートされていなければ、その高さは最も近似のサポートされている値に設定されます。

同一ラインに倍角文字がある場合は、デバイスの機能に依存します。

LineCharsが変わると、**LineHeight**は、選択された幅に対するデフォルトの行高さに更新されます。

LineHeightの値は、**Open**メソッド後最初にデバイスをイネーブルにした時に、ポイントカード機のデフォルトの行高さに初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

LineSpacing プロパティ　R/W

形式　　LONG LineSpacing;

説明　　半角文字の印刷行の高さを示します。すなわち、印字行の高さと行間スペースの高さの両方を加えたものです。本プロパティは**MapMode**で定義された単位で表記します。

ポイントカード機によってサポートされる行間に変更されれば、その行間はこの値に設定されます。もし、その行間がサポートされなければ、その行間は最も近似的にサポートされる値に設定されます。

LineCharsまたは**LineHeight**が変わると、**LineSpacing**は、選択された幅あるいは高さに対するデフォルトの行間に更新されます。

LineSpacingの値は、**Open**メソッド後最初にデバイスをイネーブルにした時に、ポイントカード機のデフォルトの行間に初期化されます。

戻り値　　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

LineWidth プロパティ

形式　　LONG LineWidth;

説明　　**LineChars**の一行の幅です。**CapPrintMode**プロパティがTRUEならば、**MapMode**で定義された単位で表記します。

LineCharsを設定すると、**LineWidth**も更新されます。

LineWidthの値は、デバイスをイネーブルにした時に、ポイントカード機のデフォルトの行幅に初期化されます。

MapCaracterSet プロパティ R/W　*第1.7版にて追加*

形式　BOOL MapCharacterSet;

説明　TRUE：印字の際にサービスオブジェクトはアプリケーションから渡された文字を**CharacterSet**プロパティで選択された文字セットにマップします。
FALSE：マッピングはサポートされません。この場合、アプリケーションは**CharacterSet**プロパティで選択された文字セットへのマップを保証する必要があります。

CapMapCharacterSetプロパティがFALSEならば、このプロパティの値は常にFALSEです。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CharacterSet**プロパティ、**CapMapCharacterSet**プロパティ

MapMode プロパティ R/W

形式 LONG MapMode;

説明 ポイントカード機のマッピングモードを示します。マッピングモードは他のプロパティで使用されるラインの高さや行間を示すような尺度の単位を定義します。

以下のマップモードをサポートしています。

値	意味
PCRW_MM_DOTS	ポイントカード機のドット幅。この幅は、各ポイントカード機によって異なるかもしれません。
PCRW_MM_TWIPS	1インチの1/1440。
PCRW_MM_ENGLISH	0.001インチ。
PCRW_MM_METRIC	0.01ミリメータ。

MapModeを設定すると、**LineHeight**、**LineSpacing**、**LineWidth**、も変化するかもしれません。

MapModeの値は、**Open**メソッド後最初にデバイスをイネーブルにした時に、PCRW_MM_DOTSに初期化されます。

戻り値 本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	不正なマッピングモードが指定されました。

MaxLine プロパティ

形式　LONG MaxLine;

説明　CapPrintModeプロパティがFALSEの時、MaxLineは、印字可能な最大ライン数を示します。

同一ラインに倍角文字がある場合は、デバイスの機能に依存します。

LineHeight、LineSpacingが変わると、MaxLineは、更新される場合もあります。

参照　LineHeightプロパティ

PrintHeight プロパティ

形式　LONG PrintHeight;

説明　CapPrintModeプロパティがTRUEの時、デバイス固有の印字エリアを縦方向にMapMode単位の数で表します。

MapModeが変更されるとPrintHeightプロパティの値は変化します。

参照　MapModeプロパティ

ResultCodeExtended プロパティ

形式 LONG ResultCodeExtended;

説明 ResultCodeがOPOS_E_EXTENDEDの場合、このプロパティにデバイスクラスの説明で記述されているクラス固有のエラー情報値が設定されます。

ResultCodeが別の値の場合、サービスオブジェクトがSO固有の値をこのプロパティに設定できます。これらの値に意味があるのは、アプリケーションがSO固有の値を追加して処理する場合だけです。

値	意味
OPOS_EPCRW_READ	リードエラー
OPOS_EPCRW_WRITE	ライトエラー
OPOS_EPCRW_JAM	カード詰まり
OPOS_EPCRW_MOTOR	搬送モーターエラー
OPOS_EPCRW_COVER	搬送路カバーオープン
OPOS_EPCRW_PRINTER	プリンターエラー
OPOS_EPCRW_RELEASE	カードが挿入口に残っています
OPOS_EPCRW_DISPLAY	表示器エラー
OPOS_EPCRW_NOCARD	リーダ内にカードがありません

ReadState1 プロパティ

形式　LONG ReadState1;

説明　４つのトラックに関する読取情報のLONG値を4分割して持っています。下の図はどのように **ReadState1**が分割されているかを示します。このプロパティは、**DataEvent**、**ErrorEvent**を通知する直前に値がセットされます。このプロパティは、**Open**メソッドにより初期化されます。

上位ワード		下位ワード	
上位バイト	下位バイト	上位バイト	下位バイト
トラック 4	トラック 3	トラック 2	トラック 1

以下の値がセットされます。

値	意味

OPOS_SUCCESS　正常に読み込まれました。または読み込み対象トラックでは有りません。

OPOS_EPCRW_START　　スタートマークエラーです。

OPOS_EPCRW_END　　エンドマークエラーです。

OPOS_EPCRW_PARITY　パリティーエラーです。

OPOS_EPCRW_ENCODE　エンコードが有りません。

OPOS_EPCRW_LRC　　LRCエラーです。

OPOS_EPCRW_VERIFY　ベリファイエラーです。

OPOS_E_FAILURE　　その他エラー

ReadState2 プロパティ

形式 LONG ReadState2;

説明 ２つのトラックに関する読取情報のLONG値を4分割して持っています。下の図はどのように **ReadState2**が分割されているかを示します。このプロパティは、**DataEvent**、**ErrorEvent**を通知する直前に値がセットされます。このプロパティは、**Open**メソッドにより初期化されます。

上位ワード		下位ワード	
上位バイト	下位バイト	上位バイト	下位バイト
未使用	未使用	トラック 6	トラック 5

以下の値がセットされます。

値	意味

OPOS_SUCCESS　正常に読み込まれました。または読み込み対象トラックでは有りません。

OPOS_EPCRW_START　スタートマークエラーです。

OPOS_EPCRW_END　エンドマークエラーです。

OPOS_EPCRW_PARITY　パリティーエラーです。

OPOS_EPCRW_ENCODE　エンコードが有りません。

OPOS_EPCRW_LRC　LRCエラーです。

OPOS_EPCRW_VERIFY　ベリファイエラーです。

OPOS_E_FAILURE　その他エラー

RecvLength1 プロパティ

形式　　LONG RecvLength 1

説明　４つのトラックに関するバイト情報のLONG値を4分割して持っています。下の図はどのように **RecvLength 1** が分割されているかを示します。

上位ワード		下位ワード	
上位バイト	下位バイト	上位バイト	下位バイト
トラック　4	トラック　3	トラック　2	トラック　1

TrackstoReadで表される読み込み可能なトラックのデータが読み込まれます。読み込んだトラックの磁気データの長さが0なら、トラックのデータ読み込みはありません。これは、デバイスがそのトラックに対する読取りヘッドを単に持たない場合、磁気ストライプがSTX,ETX,LRCだけの場合、または、何らかのエラーによってデータの読み込みが出来なかった場合です。値が

1以上なら、対応する**TrackxData**プロパティのデータ長を示します。

DataEventイベントが通知される直前に、操作されたポイントカードの磁気データは **Track1Data**、**Track2Data**、**Track3Data**と**Track4Data**プロパティに設定されます。

RecvLength2 プロパティ

形式　　LONG RecvLength2

説明　トラック５とトラック６の２つのトラックに関するバイト情報のLONG値を4分割して持っています。下の図はどのように **RecvLength2** が分割されているかを示します。

上位ワード		下位ワード	
上位バイト	下位バイト	上位バイト	下位バイト
未使用	未使用	トラック 6	トラック 5

TrackstoReadで表される読み込み可能なトラックのデータが読み込まれます。読み込んだトラックの磁気データの長さが0なら、トラックのデータ読み込みはありません。これは、デバイスがそのトラックに対する読み取りヘッドを単に持たない場合、磁気ストライプがSTX,ETX,LRCだけの場合、または、何らかのエラーによってデータの読み込みが出来なかった場合です。値が

1以上なら、対応する**Track*x*Data**プロパティのデータ長を示します。

DataEventイベントが通知される直前に、操作されたポイントカードの磁気データは**Track5Data**と**Track6Data**プロパティに設定されます。

SidewaysMaxChars プロパティ

形式　LONG SidewaysMaxChars;

説明　サイドウェイモード（左90度または右90度回転印刷）において、各行に印刷される半角文字の最大文字数です。

CapLeft90とCapRight90の両方がFALSEならば、SidewaysMaxCharsは0です。

LineHeightとLineSpacing、LineCharsプロパティを変えることにより、このプロパティは変わるかもしれません。

このプロパティは、Openメソッドにより初期化されます。

参照　SidewaysMaxLinesプロパティ

SidewaysMaxLines プロパティ

形式　LONG SidewaysMaxLines;

説明　サイドウェイモード（左90度または右90度回転印刷）において、印刷される最大行数です。

CapLeft90とCapRight90の両方がFALSEならば、SidewaysMaxLinesは0です。

LineHeightとLineSpacing、LineCharsプロパティを変えることにより、このプロパティは変わるかもしれません。

このプロパティは、Openメソッドにより初期化されます。

参照　SidewaysMaxCharsプロパティ

TracksToRead プロパティ R/W

形式　LONG TracksToRead;

説明　**CapTrackToRead**で表される読込み可能なトラックへの読込み位置を指定します。このプロパティで指定された内容に沿って**RecvLengthX**プロパティに値がセットされます。読込みを行うトラックは、PCRW_TRACK1からPCRW_TRACK6で表します。**CapTrackToRead**の値が、PCRW_TRACK1と PCRW_TRACK2とPCRW_TRACK3の論理和で表されるとき、トラック１、トラック2、トラック3へ書込みが行われます。

このプロパティは、**Open**メソッド後最初にデバイスをイネーブルした時に初期化されます。

戻り値　本プロパティ設定時に、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_BUSY	出力中で実行できません。
OPOS_E_ILLEGAL	無効な値が指定されました。読込み出来ないトラックを指定しました。
その他	**ResultCode**の項目を参照してください。

参照　**CapTrackToRead**プロパティ

TracksToWrite プロパティ R/W

形式　LONG TracksToWrite;

説明　CapTrackToWriteで表される書込み可能なトラックへの書込み位置を指定します。このプロパティで指定された内容に沿ってPrintWriteメソッドで実際に書込みが行われます。書込みを行うトラックは、PCRW_TRACK1からPCRW_TRACK6で表します。CapTrackToWriteの値が、PCRW_TRACK1と PCRW_TRACK2とPCRW_TRACK3の論理和で表されるとき、トラック１、トラック2、トラック3へ書込みが行われます。

このプロパティは、Openメソッド後最初にデバイスをイネーブルした時に初期化されます。

戻り値　本プロパティ設定時に、次の値のいずれかがResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_BUSY	出力中で実行できません。
OPOS_E_ILLEGAL	無効な値が指定されました。書込み出来ないトラックを指定しました。
その他	ResultCodeの項目を参照してください。

参照　CapTrackToWriteプロパティ、PrintWriteメソッド

Track1Data プロパティ

形式　BSTR Track1Data;

説明　ポイントカードのトラック1の内容、あるいは空の文字列です。

このプロパティには、開始符号と終了符号を含まないトラックデータが入ります。

空の文字列は、トラックがアクセス可能でなかったことを示します。

Track2Data プロパティ

形式　　BSTR Track2Data;

説明　　ポイントカードのトラック2の内容、あるいは空の文字列です。

このプロパティには、開始符号と終了符号を含まないトラックデータが入ります。

空の文字列は、トラックがアクセス可能でなかったことを示します。

Track3Data プロパティ

形式　　BSTR Track3Data;

説明　　ポイントカードのトラック3の内容、あるいは空の文字列です。

このプロパティには、開始符号と終了符号を含まないトラックデータが入ります。

空の文字列は、トラックがアクセス可能でなかったことを示します。

Track4Data プロパティ

形式　　BSTR Track4Data;

説明　　ポイントカードのトラック4の内容、あるいは空の文字列です。

このプロパティには、開始符号と終了符号を含まないトラックデータが入ります。

空の文字列は、トラックがアクセス可能でなかったことを示します。

Track5Data プロパティ

形式　　BSTR Track5Data;

説明　　ポイントカードのトラック5の内容、あるいは空の文字列です。

このプロパティには、開始符号と終了符号を含まないトラックデータが入ります。

空の文字列は、トラックがアクセス可能でなかったことを示します。

Track6Data プロパティ

形式　　BSTR Track6Data;

説明　　ポイントカードのトラック6の内容、あるいは空の文字列です。

このプロパティには、開始符号と終了符号を含まないトラックデータが入ります。

空の文字列は、トラックがアクセス可能でなかったことを示します。

WriteState1 プロパティ

形式 LONG WriteState 1

説明 ４つのトラックに関する書込み情報のLONG値を4分割して持っています。下の図はどのように **WriteState 1** が分割されているかを示します。このプロパティは、**ErrorEvent**を通知する直前に値がセットされます。

上位ワード		下位ワード	
上位バイト	下位バイト	上位バイト	下位バイト
トラック 4	トラック 3	トラック 2	トラック 1

以下の値がセットされます。

値	意味
OPOS_SUCCESS	正常に書き込まれました。または、書込み対象トラックでは有りません。
OPOS_EPCRW_START	スタートマークエラーです。
OPOS_EPCRW_END	エンドマークエラーです。
OPOS_EPCRW_PARITY	パリティーエラーです。
OPOS_EPCRW_ENCODE	エンコードが有りません。
OPOS_EPCRW_LRC	LRCエラーです。
OPOS_EPCRW_VERIFY	ベリファイエラーです。
OPOS_E_FAILURE	その他エラー

WriteState2 プロパティ

形式　LONG WriteState2

説明　２つのトラックに関する読取情報のLONG値を4分割して持っています。下の図はどのように **WriteState2**が分割されているかを示します。このプロパティは、**ErrorEvent**を通知する直前に値がセットされます。

上位ワード		下位ワード	
上位バイト	下位バイト	上位バイト	下位バイト
未使用	未使用	トラック 6	トラック 5

以下の値がセットされます。

値	意味

OPOS_SUCCESS　正常に書き込まれました。または、書込み対象トラックでは有りません。

OPOS_EPCRW_START　スタートマークエラーです。

OPOS_EPCRW_END　エンドマークエラーです。

OPOS_EPCRW_PARITYパリティーエラーです。

OPOS_EPCRW_ENCODE　エンコードが有りません。

OPOS_EPCRW_LRC　LRCエラーです。

OPOS_EPCRW_VERIFYベリファイエラーです。

OPOS_E_FAILURE　その他エラー

Write1Data プロパティ

形式　　BSTR Write1Data;

説明　　PrintWriteメソッドにて書き込むポイントカード機のトラック１のライトデータを設定します。

このプロパティには、開始符号と終了符号を含まないデータをセットします。

Write2Data プロパティ

形式　　BSTR Write2Data;

説明　　PrintWriteメソッドにて書き込むポイントカード機のトラック２のライトデータを設定します。

このプロパティには、開始符号と終了符号を含まないデータをセットします。

Write3Data プロパティ

形式　　BSTR Write3Data;

説明　　PrintWriteメソッドにて書き込むポイントカード機のトラック３のライトデータを設定します。

このプロパティには、開始符号と終了符号を含まないデータをセットします。

Write4Data プロパティ

形式　　BSTR Write4Data;

説明　　PrintWriteメソッドにて書き込むポイントカード機のトラック４のライトデータを設定します。

このプロパティには、開始符号と終了符号を含まないデータをセットします。

Write5Data プロパティ

形式　　BSTR Write5Data;

説明　　PrintWriteメソッドにて書き込むポイントカード機のトラック５のライトデータを設定します。

このプロパティには、開始符号と終了符号を含まないデータをセットします。

Write6Data プロパティ

形式　　BSTR Write6Data;

説明　　PrintWriteメソッドにて書き込むポイントカード機のトラック６のライトデータを設定します。

このプロパティには、開始符号と終了符号を含まないデータをセットします。

メソッド

BeginInsertion メソッド

形式 LONG BeginInsertion (LONG *Timeout*);

このメソッドは、*Timeout*パラメータにミリ秒単位で設定された値の間、デバイスの処理を待ちます。パラメータ値が0の場合、このメソッドはデバイスをポイントカード挿入モードにし、即時に適切なステータスを戻します。OPOS_FOREVER（-1）の場合、このメソッドはデバイスをポイントカード挿入モードにし、ポイントカードが挿入されるかエラーが発生するまで待ちます。

説明 ポイントカードの取り込みを開始する時に呼び出します。

呼び出されると、デバイスをポイントカード取り込みモードにし、ポイントカード受け取りの準備をします。ポイントカード取り込み処理のため、このメソッドは、**EndInsertion**メソッドとセットで呼び出します。

もしポイントカード機デバイスを取り込みモードに設定できなければ、アプリケーションにエラーが返されます。それ以外の場合、コントロールは次のいずれかの状況まで取り込み状況をモニターします。

ポイントカードの取り込みに成功した場合：

この場合、ポイントカード機コントロールはOPOS_SUCCESSを返します。

Timeout パラメータに設定したミリ秒単位の値が経過するまでに、ポイントカードが取り込まれない場合やエラーがポイントカード機デバイスから報告された場合：

この場合、ポイントカード機コントロールは、OPOS_E_TIMEOUTか他のエラーを返します。ただし、ポイントカード機デバイスは、デバイスの動作確認メカニズムを変えずに、アプリケーションがユーザとの対話処理や**BeginInsertion**メソッドを再発行できるよう取り込みモードを維持します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	カードが正常に挿入されました。
OPOS_E_BUSY	出力中で実行できません。
OPOS_E_ILLEGAL	無効な*Timeout*値が指定されました。
OPOS_E_TIMEOUT	ポイントカードが適切に挿入されずに、設定されたタイムアウト値が経過しました。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

参照　**EndInsertion**メソッド、**BeginRemoval**メソッド、**EndRemoval**メソッド

BeginRemoval メソッド

形式　　LONG BeginRemoval (LONG *Timeout*);

このメソッドは、*Timeout*パラメータにミリ秒単位で設定された値の間、デバイスの処理を待ちます。パラメータ値が0の場合、このメソッドは、ポイントカードの排出を実行し、即時に適切なステータスを戻します。OPOS_FOREVER（-1）の場合、このメソッドは、ポイントカードが排出され、挿入口から抜き取られるかエラーが発生するまで待ちます。

説明　　ポイントカードの排出処理をする時に呼び出します。

このメソッドが呼び出されると、ポイントカード機をポイントカード排出モードにし、ポイントカードを排出します。

挿入口にセンサーを有するタイプは、カードがポイントカード機から抜き取られることで、正常終了し、センサーを有しないタイプは、排出処理が実行されることで、正常終了します。

このメソッドは、**EndRemoval**メソッドとセットで呼び出します。

もしポイントカード機デバイスを排出モードに設定できなければ、アプリケーションにエラーが返されます。それ以外の場合、コントロールは次のいずれかの状況まで排出状況をモニターします。

ポイントカードの排出成功した場合：

この場合、ポイントカード機コントロールは、OPOS_SUCCESSを返します。

Timeout パラメータに設定したミリ秒単位の値が経過するまでポイントカードが排出されない場合やエラーがポイントカード機デバイスから報告された場合：

ポイントカード機コントロールは、OPOS_E_TIMEOUTか他のエラーを返します。ただし、ポイントカード機デバイスは、デバイスの動作確認メカニズムを変えずに、アプリケーションがユーザとの対話処理や**BeginRemoval**メソッドを再発行できるよう排出モードを維持します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	カードが抜き取られました。
OPOS_E_BUSY	出力中で実行できません。
OPOS_E_ILLEGAL	無効な*Timeout*値が指定されました。
OPOS_E_TIMEOUT	ポイントカードが適切に排出されずに、設定されたタイムアウト値が経過しました。**CapCardEntranceSensor**がTRUEの場合は、**CardState**詳細情報がセットされます。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

参照　**CapCardEntranceSensor**ﾌﾟﾛﾊﾟﾃｨ、**CardState**ﾌﾟﾛﾊﾟﾃｨ、**BeginInsertion**メソッド、**EndInsertion**メソッド、**EndRemoval**メソッド

CleanCard メソッド

形式 LONG CleanCard;

説明 ポイントカード機のクリーニングを行います。このメソッドは、**CapCleanCard**プロパティがTRUEの時、ポイントカード機のクリーニングを実行します。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に実行されました。
OPOS_E_ILLEGAL	クリーニングの機能が有りません。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

参照 **CapClearning**プロパティ

ClearPrintWrite メソッド

形式　LONG ClearPrintWrite (LONG *Kind*, LONG *Hposition*, LONG *Vposition*, LONG *Width*、 LONG *Height*);

パラメータ	説明
Kind	消し込みを行う機能を指定します。　1：印字面　2：磁気面　3：印字面＋磁気面

Hposition　水平方向の消去開始位置を**MapMode**の単位で指定します。**CapClearPrint**プロパティがTRUEの時この値は、有効に機能します。

*Vposition*垂直方向の消去開始位置を**MapMode**の単位で指定します。**CapClearPrint**プロパティがTRUEの時この値は、有効に機能します。

Width 消し込みを行う幅を指定します。**CapClearPrint**プロパティがTRUEの時この値は、有効に機能します。

Height　消し込みを行う高さを指定します。**CapClearPrint**プロパティがTRUEの時この値は、有効に機能します。

説明　ポイントカード機の印字のクリアと磁気ストライプのデータをクリアします。

このメソッドは、**CapPrint**メソッドがTRUEで**CapClearPrint**プロパティがTRUEの時、クリアする印字面の範囲指定が有効に機能します。パラメータ*Hposition,Vposition,Width,Height*の値がそれぞれ０，０，－１、－１の時は、印字面の全面クリアを行います。

磁気面の初期化は、デバイスの機能に依存します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効なパラメータ値がセットされました。
OPOS_E_BUSY	他の処理が処理中です。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

参照　**CapClearPrint**プロパティ、**CapPrint**プロパティ

EndInsertion メソッド

形式　LONG EndInsertion ();

説明　ポイントカードの取り込み終了のため呼び出します。

呼び出されると、ポイントカード機はポイントカード取り込みモードを終了します。ポイントカードがデバイスに正しく感知されると、OPOS_SUCCESSがアプリケーションに返されます。ポイントカードが挿入されていない場合、常にOPOS_EPCRW_NOCARDが拡張エラーステータスとして戻ります。

このメソッドは、ポイントカード取り込み処理のため**BeginInsertion**メソッドとセットで呼び出します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	カードが正常に取り込まれました。
OPOS_E_ILLEGAL	ポイントカード機がポイントカード取り込みモードではありません。
OPOS_E_FAILURE	ポイントカード機にカードが入っていません。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

参照　**BeginInsertion**メソッド、**BeginRemoval**メソッド、**EndRemoval**メソッド

EndRemoval メソッド

形式 LONG EndRemoval ();

説明 ポイントカードの排出終了のため呼び出します。

呼び出されると、ポイントカード機はポイントカード排出モードを終了します。ポイントカードがデバイスに感知されないと、OPOS_SUCCESSがアプリケーションに返されます。ポイントカードが残っている場合、OPOS_E_EXTENDEDエラーが戻され、**ResultCodeExtended**プロパティにOPOS_EPCRW_RELEASEがセットされます。

このメソッドは、ポイントカード排出処理のため、**BeginRemoval**メソッドとセットで呼び出します。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	カードが正常に排出されました。
OPOS_E_ILLEGAL	ポイントカード機はポイントカード排出モードではありません。
OPOS_E_FAILURE	ポイントカード機にカードが残っています。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

参照 **BeginInsertion**メソッド、**EndInsertion**メソッド、**BeginRemoval**メソッド

PrintWrite メソッド

形式　LONG PrintWrite (LONG *Kind*, LONG *Hposition*, LONG *Vposition*, BSTR *Data*);

パラメータ	説明
Kind	カードに書き込むデータを指定します。　1：Print　2：Write　3：Print+Write

Hposition　水平方向の印字開始位置を**MapMode**の単位で指定します。**CapPrintMode**プロパティがTRUEの時この値は、有効に機能します。

*Vposition*垂直方向の印字開始位置を**MapMode**の単位で指定します。**CapPrintMode**プロパティがTRUEの時この値は、有効に機能します。

Data　印字データを設定します。印字データはポイントカード機の機能に依存します。

詳細は**BinaryConversion**プロパティを参照してください。

説明　**CapPrint**プロパティがTRUEのとき、セットされた印字データをポイントカードに印字します。

ポイントカードに**Write 1 Data**から**Write6Data**にセットされた値を**TracksToWrite**プロパティに従って磁気データの書込みと印字を行います。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効なパラメータ値がセットされました。
OPOS_E_BUSY	他の処理が処理中です。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

参照　**CapTracksToWrite, CapPrint,CapPrintMode, WriteXData,TracksToWrite**プロパティ

RotatePrint メソッド

形式 LONG RotatePrint (LONG *Rotation*);

パラメータ	説明
Rotation	回転方向。下記の値を参照。 デバイス固有の値を０度とします。

値	意味
PCRW_RP_RIGHT90	右に90度回転（時計回りに）。
PCRW_RP_LEFT90	左に90度回転（時計と反対回りに）。
PCRW_RP_ROTATE180	180度回転、つまり、倒立印字。
PCRW _RP_NORMAL	回転モードの終了

説明 回転印刷モードの開始と終了。

各デバイス固有の印字方向を０度として、回転する方向を指定します。

RotatePrintメソッドは、リライト式の印字エリアの回転方向を指定します。**PrintWrite**メソッドを実行すると、印字エリアに指定した方向に回転した印字データが印字されます。**RotatePrint**が*Rotation*パラメータをPCRW_RP_NORMALに設定して呼ぶ事で、回転モードは解除されます。

回転モードを変えることは、リライト式の印字エリアの行高さや行間、行幅、他の長さも変えるかもしれません。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	回転をサポートしていません。
OPOS_E_BUSY	出力中なので実行できません。
OPOS_E_EXTENDED	**ResultCodeExtended** の項目を参照してください。
その他	**ResultCode**の項目を参照してください。

参照 「印字データとエスケープシーケンス」

ValidateData メソッド

形式　LONG ValidateData(BSTR *Data*);

パラメータ	説明
Data	判定されるデータ。印刷可能データとエスケープシーケンスを含みます。 引数値の*Data*は、**BinaryConversion**の値によって定義されます。

説明　**PrintWrite**メソッドを呼び出す前に、リライト式の印字エリアにとって、おそらく一つ以上のエスケープシーケンスを含むデータシーケンスが、有効か否かを判定する時に呼び出します。

このメソッドは、全く印刷しませんが、リライト式の印字エリアの能力を判定するために使用されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	データは有効です。
OPOS_E_ILLEGAL	一つ以上のエスケープシーケンスが範囲外です。しかし、コントロールは有効な選択肢を選ぶことができます。
OPOS_E_FAILURE	一つ以上のエスケープシーケンスがサポートされていません。選ぶことのできる選択肢はありません。

OPOS_E_ILLEGALが返されるケースは、

エスケープシーケンス	状況
アンダーライン	太さ'#'は正確にサポートされていません。コントロールはサポートされている最も近い値を選択します。
網掛け文字	パーセンテージ'#'は正確にサポートされていません。コントロールはサポートされている最も近い値を選択します。
横倍率	倍率'#'はサポートされていません。コントロールはサポートされている最も近い値を選択します。
縦倍率	倍率'#'はサポートされていません。コントロールはサポートされている最も近い値を選択します。

OPOS_E_FAILUREが返されるケースは、

エスケープシーケンス	状況
(一般)	エスケープシーケンスの書式が無効です。
フォントタイプ指定	フォント番号'#'がサポートされていません。
ボールド	サポートされていません。
アンダーライン	サポートされていません。
イタリック	サポートされていません。
反転文字	サポートされていません。
縦横1倍角	サポートされていません。
横倍角	サポートされていません。
縦倍角	サポートされていません。
縦横倍角	サポートされていません。

参照　「**印字データとエスケープシーケンス**」

イベント

DataEvent イベント

形式　　void DataEvent (LONG *Status*);

説明　　デバイスからアプリケーションに、入力データを送るために通知されます。このイベントが通知される前に、ポイントカード機からの応答データが各プロパティに設定されます。

ErrorEventイベント

形式 **void ErrorEvent(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**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**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_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が、すでにバッファリングされたデータの一部は利用できます。

*pErrorResponse*パラメータが示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_CLEAR	バッファリングされている入力データをクリアし、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、**DataEventEnabled**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、**DataEventEnabled**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　磁気ストライプデータの読み込みと書込みにおいて、磁気ストライプデータにエラーがある場合、イベントが発行されます。

正しいアプリケーションシーケンシングが行われるよう、**DataEventEnabled**プロパティがTRUEになるまで、入力エラー及び出力のイベントは通知されません。

カード挿入において、**TracksToRead**プロパティで示される読み込可能な磁気データのいずれかのトラックに異常がある場合、**ErrorEvent**を通知する直前に、異常が有ったトラックの**ReadStateX**プロパティにエラー情報がセットされ、**TrackLengthX**プロパティには、読み込みデータ長０がセットされ、**TrackXData**プロパティには空の文字列がセットされます。

PrintWriteメソッドの実行時、書込みトラックのいずれかに異常がある場合、**WriteStateX**プロパティに異常が有ったトラックのエラー情報が、**ErrorEvent**を通知する直前にセットされます。

参照　「ステータス、結果コード、状態モデル」、**ReadStateX**プロパティ、**TrackLengthX**プロパティ、**TrackXData**プロパティ、**WriteStateX**プロパティ

StatusUpdateEventイベント

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

説明　　挿入口にセンサーを有するポイントカード機は、ポイントカードが挿入口にセットされると、**StatusUpdateEvent**が通知されます。**CapCardEntranceSensor**がFALSEならば、デバイスは、状態を通知する機能をサポートしないので、このイベントは、カードの挿入状態の変化をアプリケーションへ通知することはない。

値	意味
PCRW_SUE_NOCARD	カードがリーダ内に存在しないか、カードリーダにカードをセンスするセンサーがない
PCRW_SUE_REMAINING	挿入口にカードがあります。
PCRW_SUE_INRW	カードリーダ内にカードがあります。

*電源状態通知の**StatusUpdateEvent**の値*

「第2章 共通プロパティ、メソッド、イベント」の**StatusUpdateEvent**の記述を参照。

第 23 章

パワーマネージメント

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.5	Boolean	R/W	*適用外*
BinaryConversion	1.5	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.5	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.5	String	R	Open
Claimed	1.5	Boolean	R	Open
DataCount	1.5	Long	R	*適用外*
DataEventEnabled	1.5	Boolean	R/W	*適用外*
DeviceEnabled	1.5	Boolean	R/W	Open
FreezeEvents	1.5	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.5	Long	R	*適用外*
PowerNotify	1.5	Long	R/W	Open
PowerState	1.5	Long	R	Open
ResultCode	1.5	Long	R	無し
ResultCodeExtended	1.5	Long	R	Open
State	1.5	Long	R	無し
ControlObjectDescription	1.5	String	R	無し
ControlObjectVersion	1.5	Long	R	無し
ServiceObjectDescription	1.5	String	R	Open
ServiceObjectVersion	1.5	Long	R	Open
DeviceDescription	1.5	String	R	Open
DeviceName	1.5	String	R	Open

プロパティ (続き)

専用	*版数*	*型*	*アクセス*	*使用可能条件*
BatteryCapacityRemaining	1.9	Long	R	Open
BatteryCapacityRemainingInSeconds	1.16	Long	R	Open
BatteryCriticallyLowThreshold	1.9	Long	R/W	Open
BatteryCriticallyLowThresholdInSeconds	1.16	Long	R/W	Open
BatteryLowThreshold	1.9	Long	R/W	Open
BatteryLowThresholdInSeconds	1.16	Long	R/W	Open
ChargeTime	1.16	Long	R/W	Open
CapBatteryCapacityRemaining	1.9	Long	R/W	Open
CapBatteryCapacityRemainingInSeconds	1.16	Boolean	R	Open
CapChargeTime	1.16	Boolean	R	Open
CapFanAlarm	1.5	Boolean	R	Open
CapHeatAlarm	1.5	Boolean	R	Open
CapQuickCharge	1.5	Boolean	R	Open
CapRestartPOS	1.9	Boolean	R	Open
CapShutdownPOS	1.5	Boolean	R	Open
CapStandbyPOS	1.9	Boolean	R	Open
CapSuspendPOS	1.9	Boolean	R	Open
CapUPSChargeState	1.5	Long	R	Open
CapVariableBatteryCriticallyLowThreshold	1.9	Boolean	R	Open
CapVariableBatteryCriticallyLowThresholdInSeconds	1.16	Boolean	R	Open
CapVariableBatteryLowThreshold	1.9	Boolean	R	Open
CapVariableBatteryLowThresholdInSeconds	1.16	Boolean	R	Open
EnforcedShutdownDelayTime	1.5	Long	R/W	Open
PowerFailDelayTime	1.5	Long	R	Open
PowerSource	1.9	Long	R	Open
QuickChargeMode	1.5	Boolean	R	Open
QuickChargeTime	1.5	Long	R	Open
UPSChargeState	1.5	Long	R	Open & Enable

メソッド

共通	*版数*	*使用可能条件*
Open	1.5	無し
Close	1.5	Open
ClaimDevice	1.5	Open
ReleaseDevice	1.5	Open & Claim
CheckHealth	1.5	Open & Enable
ClearInput	1.5	*適用外*
ClearInputProperties	1.10	*適用外*
ClearOutput	1.5	*適用外*
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.5	Open
ResetStatistics	1.8	Open & Enable
RetrieveStatistics	1.8	Open & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open & Enable
専用		
RestartPOS	1.9	Open & Enable
ShutdownPOS	1.5	Open & Enable
StandbyPOS	1.9	Open & Enable
SuspendPOS	1.9	Open & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.5	*適用外*
DirectIOEvent	1.5	Open,
ErrorEvent	1.5	*適用外*
OutputCompleteEvent	1.5	*適用外*
StatusUpdateEvent	1.5	Open & Enable

概説

パワーマネージメントコントロールのOLEプログラマティクIDは“OPOS.POSPower”です。

このデバイスは、第1.5 版で追加され第1.9 版、第1.16 版で更新されています。

機能　　*第1.16版で更新*

パワーマネージメントコントロールは次の機能を持っています。

システムをシャットダウンするコマンドのサポート

システムを再起動するコマンドのサポート

システムをサスペンドするコマンドのサポート

システムがスタンバイに移行するためのコマンドのサポート

OS及びH/Wによって制御される電源管理機構の制御機能

電源断が発生したことをアプリケーションへ通知する機能

電池の状態をアプリケーションに通知する機能

UPSの充電状態の変化をアプリケーションへ通知する機能

CPUの温度警告をアプリケーションに通知する機能

CPUのファン停止をアプリケーションに通知する機能

OSによる強制シャットダウンが実行されたことを通知する機能

アプリケーションにてデータをセーブ（ローカルまたは、サーバへの転送）した後、POSターミナルをシャットダウンする機能

シャットダウンの開始をアプリケーションに通知する機能

バッテリがフル充電になるまでの残り時間を%単位または秒単位で取得する機能

モデル　*第1.16 版で更新*

一般的なモデルは、第１．３版で追加された、各デバイス毎の電源通知モデルをベースにしています。電源通知に関する共通プロパティー及び共通イベントが有効になります。各プロパティ及びイベントの値はPOSターミナル本体の電源状態が設定されます。

POSターミナル本体の電源状態として次の３つがあります。

- ONLINE：POSターミナルの電源が供給され、いつでも使用できる状態です 。「使用可能」状態を示します。
- OFF：POSターミナルが電源オフまたは、電源が供給されていない状態です POSターミナルはバッテリーで動作しています。「電源断」状態です。
- OFFLINE：POSターミナルは電源オン状態ではあるが、省電力モードで動作している状態です。「ONLINE」モードに切り替えるためには、ボタンやキーボード等の操作を行なう必要があります。

上記の電源状態を更新するためには、デバイスをオープンし**PowerNotify**プロパティをOPOS_PN_ENABLEDに設定する必要があります。

POSが「電源断」状態の時（すなわち［OFF］モードにある時）最後のアプリケーションがパワーマネージメントをクローズするかもしくは、**EnforcedShutdownDelayTime**プロパティに設定されている時間が経過すると、自動的にシャットダウン処理が実行されます。

ShutdownPOSメソッドを実行すると、電源状態に関係なくシャットダウンが行われます。

第1.16版より、デバイスの対象はPOSターミナルのほかに、RSCDデバイスも含みます。

第1.9版以降

バッテリー電源供給のデバイスが追加されました。バッテリーレベルと電源を報告するプロパティが追加されたことに伴い、ローとクリティカルローのバッテリーレベルを設定することが可能であるかを示すプロパティが追加されます。また、パワーマネージメントデバイスには、スタンバイやサスペンド状態に移行させる要求を発行できるか、もしくはその要求に応答できるかを示す機能が含まれます。このモデルは、APMやACPIのようなパワーマネージメントモデルの複製を試みているのではありませんが、デバイスの提供者にきめ細かな実装を可能にします。通常、サスペンド状態はスタンバイ状態よりも電力を消費せず、さらにスタンバイ状態は電源オン状態よりも少ない消費電力です。他の電源モデルの状態とどのように対応するかの目安を以下の表で示します。

State	ACPI	APM	Description
On	S0	ON	電源が入って動作状態
Standby	S1	SUSPEND	ディスプレイとHDがOFFでRAMとCPU,ファンは通電中
Suspend	S3	SUSPEND	RAMのみに通電
Off	S5	OFF	完全に電源オフ

デバイス共有

パワーマネージメントは共有可能なデバイスです。

デバイスを共有する際のルールは以下のとおりです。

デバイスをオープンしイネーブルにすると、アプリケーションはすべてのプロパティ、メソッド及びイベントを使用することができます。

複数のアプリケーションがデバイスをオープンしイネーブルにすると、それらすべてのアプリケーションはプロパティとメソッドを扱うことができます。またステータスアップデートイベントはそれら全てのアプリケーションに通知されます。

あるアプリケーションがデバイスの排他アクセス権を獲得すると、そのアプリケーションのみがShutDownPOSメソッドを呼び出すことができます。これによってある程度のセキュリティが保証されます。例えばメインとなるPOSアプリケーションが起動時にデバイスの排他アクセス権を獲得すると、ShutDownPOSメソッドの制御はこのアプリケーションのみに限られます。

使用上の必要条件については、「一覧」を参照してください。

プロパティ

BatteryCapacityRemaining プロパティ　*第1.9 版で追加*

形式　LONG BatteryCapacityRemaining；

説明　電池の残容量を０％から１００％の値で示します。

このプロパティはOpen メソッドにより初期化されます。

参照　CapBatteryCapacityRemainingプロパティ

BatteryCapacityRemainingInSeconds プロパティ　*第1.16 版で追加*

形式　LONG BatteryCapacityRemainingInSeconds;

説明　電池の残容量を秒数で示します。

このプロパティはOpen メソッドにより初期化されます。

参照　CapBatteryCapacityRemainingInSecoundsプロパティ

BatteryCriticallyLowThreshold プロパティ　*第1.9 版で追加*

形式　LONG BatteryCriticallyLowShereshold;

説明　ゼロ以外ならば、PWR_SUE_BAT_CRITICALのStatusUpdateEventが発生する閾値をこのプロパティに保持します。１から９９の値が残容量の比率で表されます。この値が０なら、バッテリークリティカルローの報告が未サポートであるかディセーブルであることを示します。

このプロパティはOpen メソッドにより初期化されます。

戻り値　本プロパティ設定時、次の値のいずれかが ResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	不正な値が使用されました。

参照　CapVariableBatteryCriticallyLowThresholdプロパティ, StatusUpdateEventイベント

BatteryCriticallyLowThresholdInSecounds プロパティ　*第1.16 版で追加*

形式　LONG BatteryCriticallyLowThresholdInSecounds;

説明　ゼロ以外ならば、PWR_SUE_BAT_CRITICALのStatusUpdateEventが発生する閾値をこのプロパティに保持します。値が残容量の秒数で表されます。この値が０なら、バッテリークリティカルローの報告が未サポートであるかディセーブルであることを示します。

このプロパティはOpen メソッドにより初期化されます。

戻り値　本プロパティ設定時、次の値のいずれかが ResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	不正な値が使用されました。

参照　CapVariableBatteryCriticallyLowThresholdInSecoundsプロパティ, StatusUpdateEventイベント

BatteryLowThreshold プロパティ　　*第1.9 版で追加*

形式　　LONG BatteryLowThreshold；

説明　　ゼロ以外ならば、PWR_SUE_BAT_LOWのStatusUpdateEventが発生する閾値をこのプロパティに保持します。１から９９の値の残容量が比率で表されます。この値が０なら、バッテリーローの報告が未サポートであるかディセーブルであることを示します。もしバッテリーローの閾値が変更可能なら、１から９９の範囲で値を設定すれば、その値が閾値として設定されます。０の値が設定されると、バッテリーローの報告はディセーブルされます。

このプロパティはOpen メソッドにより初期化されます。

戻り値　　本プロパティ設定時、次の値のいずれかが ResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	不正な値が使用されました。

参照　　CapVariableBatteryCriticallyLowThresholdプロパティ，StatusUpdateEventイベント

BatteryLowThresholdInSecounds プロパティ　　*第1.16 版で追加*

形式　　**LONG BatteryLowThresholdInSecounds;**

説明　　ゼロ以外ならば、PWR_SUE_BAT_LOWの**StatusUpdateEvent**が発生する閾値をこのプロパティに保持します。残容量が秒数で表されます。この値が０なら、バッテリーローの報告が未サポートであるかディセーブルであることを示します。もしバッテリーローの閾値が変更可能なら、秒数で値を設定すれば、その値が閾値として設定されます。０の値が設定されると、バッテリーローの報告はディセーブルされます。

このプロパティは**Open** メソッドにより初期化されます。

戻り値　　本プロパティ設定時、次の値のいずれかが **ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	不正な値が使用されました。

参照　　**CapVariableBatteryLowThresholdInSecounds**プロパティ, **StatusUpdateEvent**イベント

CapBatteryCapacityRemaining プロパティ　　*第1.9 版で追加*

形式　　**BOOL CapBatteryCapacityRemaining ;**

説明　　TRUE：デバイスは電池残容量を通知する機能を有します。

FALSE：上記機能をサポートしません。

このプロパティは**Open** メソッドにより初期化されます。

参照　　**BatteryCapacityRemaining**プロパティ

CapBatteryCapacityRemainingInSecounds プロパティ　*第1.16 版で追加*

形式　BOOL CapBatteryCapacityRemainingInSecounds；

説明　TRUE：デバイスは電池残容量(秒数)を通知する機能を有します。

FALSE：上記機能をサポートしません。

このプロパティはOpen メソッドにより初期化されます。

参照　BatteryCapacityRemainingInSecondsプロパティ

CapFanAlarm プロパティ

形式　BOOL CapFanAlarm；

説明　TRUE：CPUのファンが停止していることを検知することが可能です。

FALSE：上記機能をサポートしません。

このプロパティはOpen メソッドにより初期化されます。

CapHeatAlarm プロパティ

形式　BOOL CapHeatAlarm；

説明　TRUE：CPUが高温状態で動作していることを検知することが可能です。

FALSE：上記機能をサポートしません。

このプロパティはOpen メソッドにより初期化されます。

CapQuickCharg プロパティ

形式　　BOOL CapQuickCharge；

説明　　TRUE：急速充電モードでのバッテリ充電をサポートしています。

FALSE：上記機能をサポートしません。

このプロパティはOpen メソッドにより初期化されます。

参照　　QuickChargeModeプロパティ、QuickChargeTimeプロパティ

CapRestartPOS プロパティ　　*第1.9 版で追加*

形式　　BOOL CapRestartPOS；

説明　　TRUE：デバイスはPOSを再起動する機能を有します。

FALSE：上記機能をサポートしません。

このプロパティはOpen メソッドにより初期化されます。

参照　　BatteryCapacityRemainingプロパティ

CapShutdownPOSプロパティ

形式　　BOOL CapShutdownPOS；

説明　　TRUE：シャットダウン機能をサポートします。

FALSE：上記機能をサポートしません。

参照　　ShutdownPOS メソッド

CapStandbyPOS プロパティ　　*第1.9 版で追加*

形式　　BOOL CapStabdbyPOS；

説明　　TRUE：デバイスはPOSをスタンバイ状態に移行する事を要求することが可能です。

FALSE：上記機能をサポートしません。

このプロパティはOpen メソッドにより初期化されます。

参照　　BatteryCapacityRemainingプロパティ

CapSuspendPOS プロパティ　　*第1.9 版で追加*

形式　　BOOL CapSuspendPOS；

説明　　TRUE：デバイスはPOSをサスペンド状態に移行する事を要求することが可能です。

FALSE：上記機能をサポートしません。

このプロパティはOpen メソッドにより初期化されます。

参照　　BatteryCapacityRemainingプロパティ

CapUPSChargeStateプロパティ

形式　　**LONG CapUPSChargeState；**

説明　　UPSが通知可能な充電状態を示します。
以下の値の論理和が設定されます。

値	意味
PWR_UPS_FULL	ニアフル充電状態の通知が可能です。
PWR_UPS_WARNING	約５０％充電状態の通知が可能です。
PWR_UPS_LOW	ニアエンプティーの通知が可能です。
PWR_UPS_CRITICAL	エンプティ状態の通知が可能です。警告なく短時間で給電されなくなります。

このプロパティは**Open** メソッドにより初期化されます。

参照　　**UPSChargeState** プロパティ

CapVariableBatteryCriticallyLowThreshold プロパティ　　*第1.9 版で追加*

形式　　**BOOL CapVariableBatteryCriticallyLowThreshold；**

説明　　TRUE：デバイスは、バッテリークリティカルロー閾値の変更をサポートします。

FALSE：上記機能をサポートしません。

このプロパティは**Open** メソッドにより初期化されます。

参照　　**BatteryCapacityRemaining**プロパティ

CapVariableBatteryCriticallyLowThresholdInSecounds プロパティ

第1.16 版で追加

形式　　BOOL CapVariableBatteryCriticallyLowThresholdInSecounds；

説明　　TRUE：デバイスは、バッテリークリティカルロー閾値(秒)の変更をサポートします。

FALSE：上記機能をサポートしません。

このプロパティは**Open** メソッドにより初期化されます。

参照　　**BatteryCriticallyLowThresholdInSecounds**プロパティ

CapVariableBatteryLowThreshold プロパティ　　***第1.9 版で追加***

形式　　BOOL CapVariableBatteryLowThreshold；

説明　　TRUE：デバイスは、バッテリーロー閾値の変更をサポートします。

FALSE：上記機能をサポートしません。

このプロパティは**Open** メソッドにより初期化されます。

参照　　**BatteryCapacityRemaining**プロパティ

CapVariableBatteryLowThresholdInSecounds プロパティ　***第1.16 版で追加***

形式　　BOOL CapVariableBatteryLowThresholdInSecounds；

説明　　TRUE：デバイスは、バッテリーロー閾値(秒)の変更をサポートします。

FALSE：上記機能をサポートしません。

このプロパティは**Open** メソッドにより初期化されます。

参照　　**BatteryLowThresholdInSecounds**プロパティ

EnforcedShutdownDelayTimeプロパティ R/W

形式　　LONG EnforcedShutdownDelayTime；

説明　　ゼロでない場合、システムは電源断発生後、規定の時間経過後POSターミナルをシャットダウンする機能をもっています。電源断発生後、システムが自動的にシャットダウンするまでの待ち時間（ミリ秒）をセットします。

０をセットすると、自動的にはシャットダウンしないので、アプリケーションがShutdownPOSメソッドを呼び出す必要があります。

アプリケーションには、自動的にシャットダウンが開始されたことが通知されます。

このプロパティはOpen メソッドにより初期化されます。

戻り値　　本プロパティ設定時、次の値のいずれかが ResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	不正な値が使用されました。

参照　　ShutdownPOSメソッド

PowerFailDelayTimeプロパティ

形式　　LONG PowerFailDelayTime；

説明　　電源断が発生してから電源断状態を通知するまでの待ち時間を設定します。この間に復電した場合は、電源断状態は通知しません。
電源事情が悪い場合、数ミリ秒から数秒の電源瞬停が時々発生し、バッテリやその他の外部装置を使用して、電源断状態にはならないようにしているケースがあり、そのような場合に不必要な電源断通知をしないための設定です。

このプロパティはシステム生成時に設定されるものであり、アプリケーションからは読み出し専用となります。

このプロパティはOpen メソッドにより初期化されます。

CapChargeTimeプロパティ　　*第1.16 版で追加*

形式　　BOOL CapChargeTime;

説明　　TRUE：フル充電になるまでの残り時間を取得する機能を有します。
FALSE：上記機能をサポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　　ChargeTimeプロパティ

ChargeTimeプロパティ　　*第1.16 版で追加*

形式　　LONG ChargeTime;

説明　　バッテリがフル充電になるまでの残り時間を秒単位の値で示します。

この値が0なら、バッテリは充電中でないか、値の取得が未サポートであることを示します。

このプロパティは、Openメソッドにより初期化されます。

参照　　CapChargeTimeプロパティ

PowerSourceプロパティ　　*第1.9 版で追加*

形式　　LONG PowerSource；

説明　電源の通知が可能であれば、このプロパティには現在の電源を保持します。このプロパティが更新されるごとにStatusUpdateEventが通知されます。

値	意味
PWR_SOURCE_NA	電源の通知はできません。
PWR_ SOURCE_AC	現在の電源はACラインです。
PWR_ SOURCE_BATTERY	現在の電源はシステムバッテリーです。この値はバッテリーで正常な動作が可能なシステムにのみ用意されています。
PWR_ SOURCE_BACKUP	現在の電源は、UPSかバックアップバッテリーのようなバックアップ電源です。

このプロパティはOpen メソッドにより初期化されます。

参照　UPSChargeState プロパティ

QuickChargeModeプロパティ

形式　BOOL QuickChargeMode；

説明　TRUE；急速充電モードでバッテリーを充電中であることを示します。

FALSE；通常のモードで充電中であることを示します。

このプロパティは、CapQuickChargeプロパティがTRUEの時にセットされます。

参照　CapQuickChargeプロパティ、QuickChargeTimeプロパティ

QuickChargeTimeプロパティ

形式　　LONG QuickChargeTime；

説明　　このプロパティは、急速充電モードでバッテリの充電が完了するまでの残時間（ミリ秒）を示します。この時間経過後、バッテリは通常の充電モードに切り替わります。

このプロパティは、CapQuickChargeプロパティがTRUEの時にセットされます。

参照　　CapQuickChargeプロパティ、QuickChargeModeプロパティ

UPSChargeStateプロパティ

形式　　LONG UPSChargeState；

説明　　このプロパティは、現在のUPSの充電状態を示します。以下の値が設定されます

値	意味
PWR_UPS_FULL	ニアフル充電状態
PWR_UPS_WARNING	約５０％の充電状態
PWR_UPS_LOW	ニアエンプティ状態。アプリケーションはバッテリを使い果たす前に処理を終了するために、シャットダウン処理を開始する必要があります。電源状態が「OFF」（電源断）になった最初の通知がこの通知でなければ、最低２分間（マシンによりデペンドします）の動作は可能です。
PWR_UPS_CRITICAL	エンプティ状態。警告なく短時間で給電されなくなります。

参照　　CapUPSChargeState プロパティ

メソッド

RestartPOS メソッド　　*第1.9 版で追加*

形式　　LONG RestartPOS ();

説明　　POSターミナルを再起動します。このメソッドは、システム電源状態に依存せず、常にシステムを再起動します

メソッドを呼び出す前に排他アクセス権を獲得した場合、そのアプリケーションだけが本メソッドを実行することができます。アプリケーションに再起動が始まったことが通知されます。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。（再起動しました。）
OPOS_E_ILLEGAL	このメソッドはサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照　　**CapRestartPropaty** プロパティ

ShutdownPOS メソッド

形式　　LONG ShutdownPOS ();

説明　　POS端末をシャットダウンします。システムの電源状態には関係なく、このメソッドはシステムを常にシャットダウンします。

メソッドを呼び出す前に排他アクセス権を獲得した場合、そのアプリケーションだけが本メソッドを実行することができます。
アプリケーションにシャットダウンが始まったことが通知されます。

電源断が発生した場合、アプリケーションは、全てのデータをセーブしてアプリケーションを所定の状態に設定した後、本メソッドを呼び出すことを推奨します。

EnforcedShutdownDelayTimeプロパティに０以上の時間がセットされ、その時間内にアプリケーションが本メソッドを呼び出さなければ、システムは自動的にシャットダウン処理を行ないます。この処理は、システムがシャットダウンされる前にバッテリーが空にならないようにするために、OSが実行するでしょう。

本メソッドは、**CapShutdownPOS**プロパティがTRUEの場合に使用可能です。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。（シャットダウンを起動しました。）
OPOS_E_ILLEGAL	このメソッドはサポートしていません。
その他	**ResultCode**の項目を参照してください。

StandbyPOS メソッド　　*第1.9 版で追加*

形式　　LONG StandbyPOS (LONG *Reason*);

説明　　システムをスタンバイ状態にするために、または、システムをスタンバイ状態にするOSやその他アプリケーションなどのシステム要求に応答するために呼び出します。

引数の*Reason*は、POS端末がスタンバイ状態になるための理由を示します。

値	記述
PWR_Reason_Request	呼び出しは、システムがスタンバイ状態になることを要求します。
PWR_Reason_Allow	呼び出しは、スタンバイのStatusUpdateEventへの応答であり、リクエストが受理されたことを示します。
PWR_Reason_Deny	呼び出しは、スタンバイのStatusUpdateEventへの応答であり、リクエストが否定されたことを示します。

戻り値　　次の値のいずれかが戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。（スタンバイで起動しました。）
OPOS_E_ILLEGAL	このメソッドはサポートしていません。
その他	ResultCodeの項目を参照してください。

参照　　CapStandbyPOS プロパティ

SuspendPOS メソッド　　*第1.9 版で追加*

形式　　LONG SuspendPOS (LONG *Reason*);

説明　　システムをサスペンド状態にするために、または、システムをサスペンド状態にするOSやその他アプリケーションなどのシステム要求に応答するために呼び出します。

引数の*Reason*は、POS端末がサスペンド状態になるための理由を示します。

値	記述
PWR_Reason_Request	呼び出しは、システムがサスペンド状態になることを要求します。
PWR_Reason_Allow	呼び出しは、サスペンドの**StatusUpdateEvent**への応答であり、リクエストが受理されたことを示します。
PWR_Reason_Deny	呼び出しは、サスペンドの**StatusUpdateEvent**への応答であり、リクエストが否定されたことを示します。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。（再起動を起動しました。）
OPOS_E_ILLEGAL	このメソッドはサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照　　**CapSuspendPOS** プロパティ

イベント

StatusUpdateEvent イベント　　*第1.9 版で更新*

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

*Status*パラメータは電源状態または、警告状態変化を通知します。以下のいずれかの値がセットされます。

値	意味
PWR_SUE_UPS_FULL	UPSの充電状態がニアフル。**CapUPSChargeState**プロパティにPWR_UPS_FULLがセットされている場合通知されます。
PWR_SUE_UPS_WARNING	UPSの充電状態が約５０％。**CapUPSChargeState**プロパティにPWR_UPS_WARNINGがセットされている場合通知されます。
PWR_SUE_UPS_LOW	UPSの充電状態がニアエンプティ。アプリケーションはバッテリを使い果たす前に処理を終了するために、シャットダウン処理を開始する必要があります。電源状態が「OFF」（電源断）になった最初の通知がこの通知でなければ、最低２分間（マシンデペントによります）の動作は可能です。**CapUPSChargeState**プロパティにPWR_UPS_LOWがセットされている場合通知されます。
PWR_SUE_UPS_CRITICAL	UPSの充電状態がエンプティ。警告なく短時間で給電されなくなります。**CapUPSChargeState**プロパティにPWR_UPS_CRITICALがセットされている場合通知されます。
PWR_SUE_FAN_STOPPED	CPUファンが停止した。**CapFanAlarm**プロパティがTRUEの時通知されます。
PWR_SUE_FAN_RUNNING	CPUファンが動作中。**CapFanAlarm**プロパティがTRUEの時通知されます。

PWR_SUE_TEMPERATURE_HIGH
CPUが高温で動作中。**CapHeatAlarm**プロパティがTRUEの時通知されます。

PWR_SUE_TEMPERATURE_OK
CPUが通常温度で動作中。**CapHeatAlarm**プロパティがTRUEの時通知されます。

PWR_SUE_SHUTDOWN
システムがまもなくシャットダウンされます。

第1.9版で以下の値が追加されました。

PWR_SUE_BAT_LOW　システムの残り電池容量がローバッテリーの閾値を下回ました。システムは電池で操作されています。

PWR_SUE_BAT_CRITICAL
システムの残り電池容量が緊急のローバッテリーの閾値を下回ました。システムは電池で操作されています。

PWR_SUE_BAT_CAPACITY_REMAINING
BatteryCapacityRemainingプロパティが更新されました。

PWR_SUE_RESTARTシステムは直ちに再起動されます。

PWR_SUE_STANDBY　システムは、スタンバイ状態に移行することを要求しています。

PWR_SUE_USER_STANDBY
システムはユーザー入力の結果としてスタンバイ状態に移行することを要求しています。

PWR_SUE_SUSPEND　システムは、サスペンド状態に移行することを要求しています。

PWR_SUE_USER_SUSPEND
システムはユーザー入力の結果としてサスペンド状態に移行する要求を継続中です。

PWR_SUE_PWR_SOURCE
PowerSourceプロパティが更新されました。

第1.16版で以下の値が追加されました。

PWR_SUE_BAT_CAPACITY_REMAINING_IN_SECONDS
BatteryCapacityRemainingInSecondsプロパティが更新されました。

説明　　電源状態、警告状態の変化を検出した時に、イベントが通知されます
第１．３版で**StatusUpdateEvent**の値に電源通知を追加すると共に電源状態の通知を追加しています。また、ファームウェアアップデートの機能が１．９版で追加されています。非同期で実行されるファームウェアの更新状況を確認するためにさらにステータスの値が追加されてます。詳細は、共通項目の**StatusUpdateEvent**の項目を参照ください。

参照　　**CapFanAlarm**、**CapHeatAlarm**、**CapUPSChargeState**、
UPSChargeStateプロパティ

第　2 4 章

スマートカードリーダライタ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.8	Boolean	R/W	適用外
BinaryConversion	1.8	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.8	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.8	String	R	Open
Claimed	1.8	Boolean	R	Open
DataCount	1.8	Long	R	Open
DataEventEnabled	1.8	Boolean	R/W	Open
DeviceEnabled	1.8	Boolean	R/W	Open & Claim
FreezeEvents	1.8	Boolean	R/W	Open
OpenResult	1.8	Long	R	無し
OutputID	1.8	Long	R	Open
PowerNotify	1.8	Long	R/W	Open
PowerState	1.8	Long	R	Open
ResultCode	1.8	Long	R	無し
ResultCodeExtended	1.8	Long	R	Open
State	1.8	Long	R	無し
ControlObjectDescription	1.8	String	R	無し
ControlObjectVersion	1.8	Long	R	無し
ServiceObjectDescription	1.8	String	R	Open
ServiceObjectVersion	1.8	Long	R	Open
DeviceDescription	1.8	String	R	Open
DeviceName	1.8	String	R	Open

専用

CapCardErrorDetection	1.8	Boolean	R	Open
CapInterfaceMode	1.8	Long	R	Open
CapIsoEmvMode	1.8	Long	R	Open
CapSCPresentSensor	1.8	Long	R	Open
CapSCSlots	1.8	Long	R	Open
CapTransamissionProtocol	1.8	Long	R	Open
InterfaceMode	1.8	Long	R/W	Open,Claim,& Enable
IsoEmvMode	1.8	Long	R/W	Open,Claim,& Enable
SCPresentSensor	1.8	Long	R	Open,Claim,& Enable
SCSlots	1.8	Long	R/W	Open,Claim,& Enable
TransactionProgress	1.8	Boolean	R	Open,Claim,& Enable
TransamissionProtocol	1.8	Long	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.8	無し
Close	1.8	Open
ClaimDevice	1.8	Open
ReleaseDevice	1.8	Open & Claim
CheckHealth	1.8	Open, Claim, & Enable
ClearInput	1.8	Open & Claim
ClearInputProperties	1.10	Open & Claim
ClearOutput	1.8	Open & Claim
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.8	Open & Claim
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable

専用		
BeginInsertion	1.8	Open, Claim,＆Enable
BeginRemoval	1.8	Open, Claime,＆Enable
EndInsertion	1.8	Open, Claim,＆Enable
EndRemoval	1.8	Open, Claim,＆Enable
ReadData	1.8	Open, Claim,＆Enable
WriteData	1.8	Open, Claim,＆Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.5	Open,Claim,&Enable
DirectIOEvent	1.5	Open & Claim
ErrorEvent	1.5	Open,Claim,&Enable
OutputCompleteEvent	1.5	Open,Claim,&Enable
StatusUpdateEvent	1.5	Open,Claim,&Enable

概説

スマートカードリーダライタコントロールのOLEプログラマティックIDは“OPOS. SmartCardRW”です。

このデバイスは、第1.8版で追加されました。

機能

スマートカードリーダライタは次の機能を持っています。

ISO/IEC 7816標準(接触型)と ISO/IEC 14443（非接触型）スマートカードへの読取と書き込みが可能です。

インターフェイスは、メモリカード、プロテクトあるいはセグメント化メモリカード、価値保存型カード(SVC)、CPU/MPU対応カード等です。

スマートカードの読取と書き込みの機能を持ちます。"Payment Terminal”のような、十分な機能をもつデバイス（スマートカードリーダライタに、ピンパッド、ディスプレイ、シグニチャキャプチャ、MSR の機能を有するデバイス）は、本コントロールではサポートしていません。

本コントロールでサポートするスマートカードは、接触型カードと非接触型カードです。

スマートカードの有る、無しを検知することが可能です。

追加サポートとして、CPU/MPUカードのセキュリティアプリケーションモジュール(SAM)が提供されるかもしれません。

4つのスマートカードリーダ通信APIがサポートされています。

１．コマンドデータモード : コマンドとデータをASCIIフォーマットに変換したもの。

２．データブロックモード : コマンドとデータを含むストリングデータブロックはサービスオブジェクトに送られます。アプリケーションおよびサービスオブジェクトは、このモードを使用する前にコミュニケーションプロトコルおよびデータフォーマットを一致させておく必要があります。

３．APDUモード : サービスオブジェクトに送られた、コマンドおよびデータを含んでいるストリングデータブロックが、ISO/IEC 7816 APDU基準に一致しているデータブロックモードです。アプリケーションは、サービスオブジェクトがISO, EMVの機能を有している場合にサポートされます。

４. XML データブロックモード: コマンドとデータを含むストリングデータブロックはサービスオブジェクトに送られます。アプリケーションおよびサービスオブジェクトは、NRF-ARTS IXRetail XML標準の中で規定されるように、XMLデータディクショナリおよびXMLスキーマガイドラインと一致するよう定義された、コミュニケーションプロトコルおよびデータフォーマットを使用しなければなりません。

モデル

スマートカードリーダライタの一般的なモデルを以下に示します。

スマードカードリーダライタは、ISO 7816 に準拠するスマートカードの種類によって、様々な使用方法があります。これらは、メモリカード、プロセッサカード(T0 , T1 TransmissionProtocolプロパティ)、電子財布カード、セキュリティアプリケーションモジュール(SAM)プロセッサカード、セキュリティカードの機能を有する接触、非接触型のカードです。スマートカードリーダライタコントロールの適用範囲は、スマードカードのデータの読取と書き込み、スマートカード上でのプログラム実行に限られます。

スマートカードリーダライタがどのようなスマートカードの処理方法が可能かを、アプリケーションは認識しなければなりません。様々な用途を容易に可能とするため、4 つの異なる通信コマンドとデータ交換メソッド(InterfaceModeプロパティ)を使用することが可能です。初期化シーケンスの一部としてアプリケーションは、InterfaceModeプロパティに使用可能なモードを設定できるかを、CapInterfaceModeプロパティを取得することで、確認を行うべきです。

デバイスに操作を開始する前に、アプリケーションは、Openメソッドと ClaimDeviceメソッドを実行しなければなりません。アプリケーションは DeviceEnabledプロパティをTRUE に設定することで、スマートカードリーダライタコントロールを使用することが可能となります。また、スマートカードを検知した場合、StatusUpdateEventイベントがアプリケーションに通知されます。

BeginInsertionメソッドは、タイムアウト値を設定して実行されます。アプリケーションはその間にスマートカードが検知されるまで待ちます。タイムアウト時間が経過した場合、アプリケーションはスマートカードの検知を続行するため、BeginInsertionメソッドを実行しなくてはなりません。スマートカードが検知されると、アプリケーションはEndInsertionメソッドを実行しなければなりません。

入力モデル

アプリケーションが、スマートカードからデータを取得する場合、ReadDataメソッドを呼び出します。データを取得すると、DataEvent イベントがキューイングされます。アプリケーションが DataEventEnabled プロパティをTRUEに設定すると、DataEvent イベントがアプリケーションに通知されます。

スマートカードのデータを読取中にエラーが発生した場合は、ErrorEventイベントがDataEventイベントの代わりにキューに入れられます。アプリケーションが DataEventEnabled プロパティをTRUEに設定すると、ErrorEvent イベントがアプリケーションに通知されます。

アプリケーションは、DataCountプロパティを取得することにより、キューイングされたデータイベントの数を取得することができます。

キューイングされたすべての入力データは、ClearInputメソッドを実行することで削除できます。

出力モデル

WriteDataメソッドは、常に非同期に呼び出されます。出力データはFIFOにて処理されます。アプリケーションが WriteDataメソッドを呼び出すと、スマートカードリーダライタコントロールに要求としてキューイングされ、スマートカードへの処理が開始されます。

InterfaceModeプロパティによって、WriteDataメソッドのデータは、サービスオブジェクトによって解析されるか、あるいは直接スマートカードリーダライタを通して、スマートカードに送られます。リクエストの識別子が、WriteDataメソッド呼び出し時に割り当てられ、OutputIDプロパティに設定されます。キューイングされたデータは直ぐに処理が開始されます。

WriteDataメソッドの実行が完了すると、OutputCompleteEventイベントがアプリケーションに通知されます。この出力リクエストのOutputIDは、OutputCompleteEventイベントのパラメータに含まれています。

WrieDataメソッドがデータ書き込み中に失敗した場合、ErrorEventイベントがアプリケーションに通知されます。アプリケーションが多数の出力リクエストを実行している場合、失敗したリクエストのOutputIDは、最後に成功したOutputCompleteEventイベントのOutputIDプロパティに次の値が設定されます。失敗したリクエストは、最後に成功したリクエストの直後に発行されたものです。

キューイングされたすべての出力データは、ClearOutputメソッドを呼び出すことで削除できます。可能ならば、実行中の出力を削除することができます。この場合、OutputCompleteEventイベントはアプリケーションに通知されません。

スマートカードにアクセスし終えた時、アプリケーションはタイムアウト値を指定して、BeginRemovalメソッドを呼び出します。タイムアウト時間までにカードが排出されない場合、エラーが返ります。アプリケーションは、スマートカードが排出されるまで、BeginRemovalメソッドを呼び出します。

スマートカードが検知されると、StatusUpdateEventイベントが通知されます。

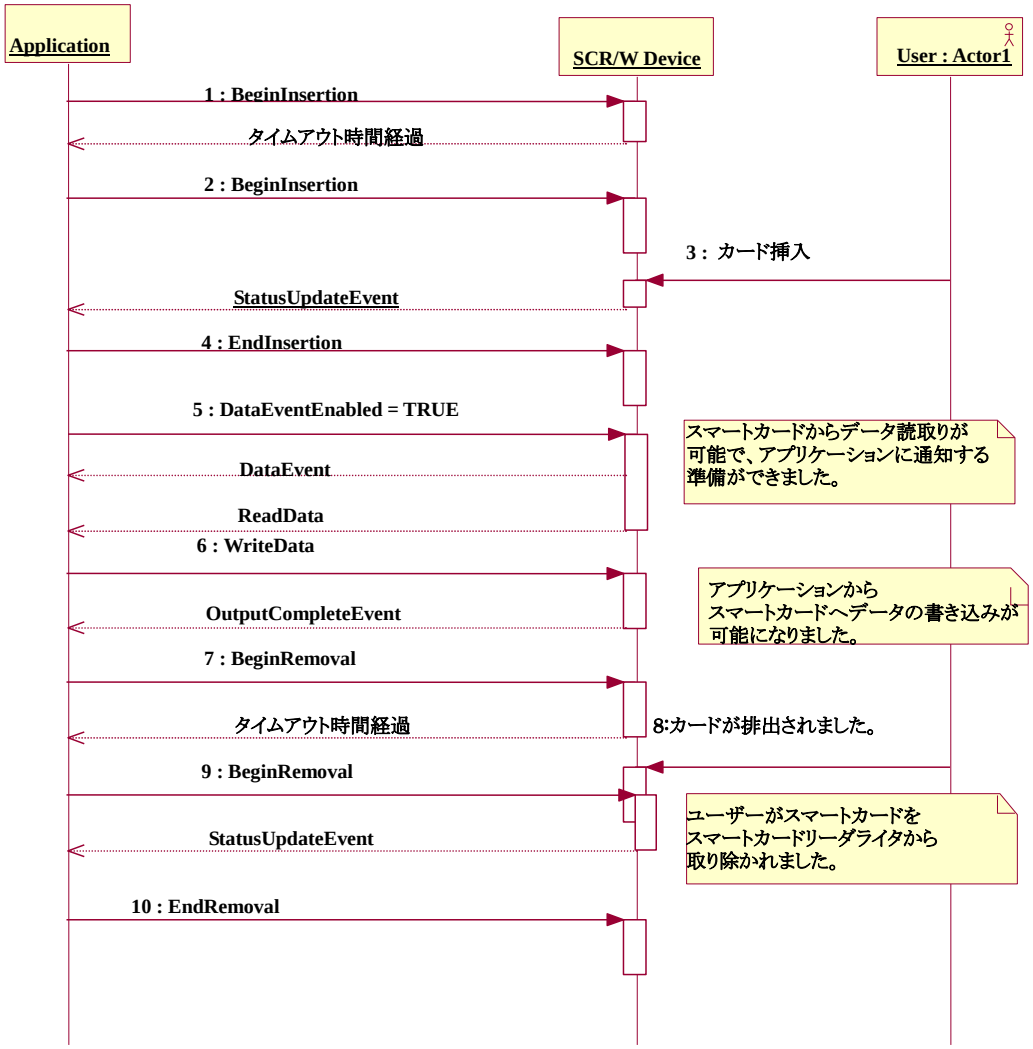
カードが物理的に排出されたか、アプリケーションがカード排出処理を中断させるかのいずれかの場合、カード排出モードを終了させるために、アプリケーションは、EndRemovalメソッドを呼び出さなければなりません。

アプリケーションがコントロールオブジェクトの使用を終了するときには、DeviceEnabledプロパティをFALSEに設定し、ReleaseDeviceメソッドを呼び出します。もしアプリケーションがデバイス制御をしなければ、Closeメソッドを呼び出します。

正常な処理が完了される前に、スマートカードが取り除かれることがあるかもしれません。これは、スマートカードリーダライタからスマートカードを「引き抜く」ような場合です。この場合、「カードが検知されくなった」として、StatusUpdateEventイベントがアプリケーションに通知されます。さらに、SCPresentSensorプロパティはFALSEに設定されます。

カード取り込みダイアグラム

カードの取り込みと排出の処理を以下に示します。WriteData以外のメソッドはすべて同期的に動作します。



(1)アプリケーションが設定したタイムアウト時間が経過する前に、スマートカードがスマートカードリーダライタに取り込まれない場合、エラーが返されます。アプリケーションはスマートカードが取り込まれたことを確認するため、**BeginInsertion**メソッドを呼び出すか、やめるために**EndInsertion**メソッドを呼び出す必要があります。**BeginInsertion**メソッドの呼び出しが成功した場合、アプリケーションはスマートカード取り込みモードを終了させ、**ReadData**メソッド、**WriteData**メソッドやその他のメソッドを呼び出し、スマートカードを使うために**EndInsertion**メソッドを呼び出さなければなりません。スマートカードを検知した場合、アプリケーションに**StatusUpdateEvent**イベントが通知されます。

(2)アプリケーションが設定したタイムアウト時間が経過する前に、スマートカードがスマートカードリーダライタに排出されない場合、エラーが返されます。アプリケーションは、スマートカードが排出されたことを確認するため、**BeginRemoval**メソッドを呼び出すか、あるいはスマートカード排出を取り消すために、**EndRemoval**メソッドを呼び出す必要があります。アプリケーションは**BeginRemoval**メソッドの呼び出し後に、スマートカードリーダライタの排出モードを終了させるために、**EndRemoval**メソッドを呼び出さなければなりません。スマートカードが検知されなくなった場合、アプリケーションに**StatusUpdateEvent**イベントが通知されます。

デバイスの共有

スマートカードリーダライタコントロールは、下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければいけません。

アプリケーションは、いくつかの専用プロパティにアクセスする前に、排他アクセス権を獲得し、イネーブルしなければなりません。

アプリケーションは、デバイスを操作するメソッドを呼び出す前に、イネーブルしなければなりません。

使用上の詳細な必要条件は、「一覧」を参照してください。

データ変換モード

スマートカードリーダライタコントロールは、スマートカードへデータの通信を可能にするいくつかのモードで機能することができる、柔軟性を持っています。スマートカードリーダライタコントロールが初期化されると、アプリケーションはスマートカードリーダライタの能力を問い合わせ、通信モードと操作モードを決定します。**InterfaceMode**プロパティは、現在の通信モードを設定するために使用されます

コマンドデータモードにおいて、単純な読取および書き込み機能は、アプリケーションとスマートカードリーダライタサービスの間で定義されます。コマンドは、データをスマートカードから取得したり、データを格納したり、プログラムを実行させたりします。このモードでは、スマートカードリーダライタの仕様についてのより多くの知識が必要となります。アプリケーションは、**DeviceName**プロパティまたは、**DeviceDescription**プロパティを取得し、システムに接続されるスマートカードリーダライタのタイプに基づき、書き込みデータや読取データを作成しなければなりません。

データブロックモードにおいて、データブロックがサービスオブジェクトから送信されたり、取得されたりします。サービスオブジェクトは、アプリケーションから渡されたコマンドとデータのブロックを解析し、現在デバイスの中にあるスマートカードの機能を呼び出します。**Open**メソッドが呼び出される前に、アプリケーションとスマートカードリーダライタの間で、メッセージ内容についての定義が確立されている必要があります。アプリケーションは、**DeviceName**プロパティと**DeviceDescription**プロパティを調べて、システムに接続されたスマートカードリーダライタのタイプに基づいたメッセージを認識すべきです。

APDUモードにおいては、データブロックが上に記述されたデータブロックモードと同様、サービスオブジェクトから送信されたり、取得されたりします。しかしながら、このモードにおいて、コマンドおよびデータは、スマートカード通信のISO/IEC 7816 APDU(Application Protocol Data Unit)に基づいたストリングデータ要素となります。この仕様はAPDUモードで動作する、ISO/EMVの要求をサポートするために用意されています。**CapIsoEmvMode**プロパティはどのモードがデバイスにサポートされているかが設定されています。その後、アプリケーションはデータの送受信に先立ち、希望する通信のメッセージスキームを**IsoEmvMode**プロパティに設定します。

XML データブロックモードにおいて、データブロックが上に記述されたデータブロックモードと同様、サービスオブジェクトから送信されたり、取得されたりします。しかしながら、このモードにおいてコマンドおよびデータは、XMLメッセージの形式になっています。データ要素とメッセージスキーマはスマートカードリーダライタデバイスに適合するIXRetail提供のXMLに従います。

プロパティ

CapCardErrorDetectionプロパティ

形式　　BOOL CapCardErrorDetection;

説明　　TRUEの場合、サービスオブジェクトはスマートカードがスマートカードリーダライタから"引き抜かれた"（データの設定がすべて終る前に引き抜かれた）ことを通知することができます。FALSE の場合、通知されません。このプロパティがTRUEの場合、スマートカードが”引き抜かれた”場合、**ErrorEvent** は **ResultCode**に、OPOS_E_ESC_TORNを設定して、通知されます。

このプロパティは、**Open**メソッドにより初期化されます。

参照　　**ErrorEvent** イベント

CapInterfaceModeプロパティ

形式 LONG CapInterfaceMode;

説明 このプロパティは、アプリケーションと通信するために、スマートカードリーダライタがサポートする、コマンドデータモード、ブロックデータモード、APDUモード、IXReatil標準を使用するXMLデータブロックモードのような通信インターフェースモードを示します。
CapInterfaceModeプロパティは、サービスオブジェクトがサポートする通信インターフェースモードが設定されています。**InterfaceMode**プロパティは、スマートカードリーダライタと通信するためにアプリケーションが使用するインターフェースモードが設定されています。

このプロパティは以下の値の論理和で設定されます。

値	意味
SC_CMODE_TRANS	コマンドデータモード
SC_CMODE_BLOCK	ブロックデータモード
SC_CMODE_APDU	APDUモード。APDU標準以外のブロックデータモードと同じコマンドおよびデータを定義します。
SC_CMODE_XML	XMLデータブロックモード

このプロパティは、**Open**メソッドにより初期化されます。

参照 **IsoEmvMode** プロパティ, **InterfaceMode** プロパティ

CapIsoEmvModeプロパティ

形式　LONG CapIsoEmvMode;

説明　このプロパティは、**InterfaceMode** プロパティが、SC_MODE_APDUに設定されている場合、アプリケーションと通信するために、スマートカードリーダライタが使用するメッセージモードを示します。APDUメッセージフォーマットは、ISOかEMVのいずれかが指定されています。

IsoEmvModeプロパティは、スマートカードリーダライタで使用するAPDUモードを選択するために使用されます。

このプロパティは以下の値の論理和で設定されます。

値	意味
SC_CMODE_ISO	現在使用中のフォーマットをメッセージとして送るAPDUは、ISOの標準に一致します。
SC_CMODE_EMV	現在使用中のフォーマットをメッセージとして送るAPDUは、EMVの標準に一致します。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**IsoEmvMode** プロパティ, **InterfceMode** プロパティ

CapSCPresentSensorプロパティ

形式　LONG CapSCPresentSensor;

説明　このプロパティは、スマートカードリーダライタが、スマートカード挿入を検知できるスロット(挿入口あるいはカード検知領域)を示します。スマートカードリーダライタは、LSBで指定される利用可能な(デフォルトのスロットとして指定された)最低1つのスロットを持ちます。

このプロパティは、ビット0(LSB)がスロット0(デフォルト), ビット1がスロット1、ビット2がスロット2 等の論理和で設定されます。ビットが1である場合、スマートカードを検知可能なセンサを持っていることを示します。センサが存在しない場合、ビットは0です。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**SCPresentSensor**プロパティ

CapSCSlotsプロパティ

形式　LONG CapSCSlots;

説明　このプロパティはスマートカード挿入可能なスロット(エントリー・ポイントあるいはカード検知領域)の数を示します。

アプリケーションは、**CapSCSlots**プロパティのうちの１つを、**SCSlot**プロパティに設定することにより、使用するスロットを選択することができます。スマートカードリーダライタはLSBで指定される利用可能な(デフォルトのスロットとして指定された)最低1つのスロットを持ちます。

このプロパティは、ビット0(LSB)がスロット0(デフォルト), ビット1がスロット1、ビット2がスロット2 等の論理和で設定されます。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**SCSlots**プロパティ

CapTransmissionProtocolプロパティ

形式　LONG CapTransmissionProtocol;

説明　このプロパティは、スマートカードリーダライタが使用可能なISO 7816-3送信プロトコル、T=0(非同期半二重の文字送信プロトコル)あるいはT=1(非同期半二重のブロック送信プロトコル)のタイプを示します。

TransmissionProtocolプロパティは、スマートカードリーダライタと通信するために使用されている送信プロトコルが設定されています。

このプロパティは以下の値の論理和で設定されます。

値	意味
SC_TRANS_PROTOCOL_T0	非同期半二重の文字送信プロトコルモード
SC_TRANS_PROTOCOL_T1	非同期半二重のブロック送信プロトコルモード

このプロパティは、Openメソッドにより初期化されます。

参照　TransmissionProtocolプロパティ

InterfaceModeプロパティ

形式 LONG InterfaceMode;

説明 このプロパティは、アプリケーションと通信するために、スマートカードリーダライタが使用する通信インターフェースモードを示します。**CapInterfaceMode**プロパティは、サービスオブジェクトがサポートする通信インターフェースモードが設定されています。

CapInterfaceMode でサポートされていない、**InterfaceMode**を選択した場合、エラーが返されます。

このプロパティは以下の値が設定されます。

値	意味
SC_MODE_TRANS	コマンドデータモード
SC_MODE_BLOCK	データブロックモード
SC_MODE_APDU	APDUモード。APDU標準以外のブロックデータモードと同じコマンドおよびデータを定義します。
SC_MODE_XML	XMLデータブロックモード

このプロパティは、**Open**メソッドにより SC_MODE_TRANS に初期化されます。

参照 **CapInterfaceMode**プロパティ

IsoEmvModeプロパティ

形式　LONG IsoEmvMode;

説明　このプロパティは、**InterfaceMode**プロパティが SC_MODE_APDU に設定された場合、スマートカードリーダライタが使用するスマートカードのメッセージモードを示します。APDUメッセージフォーマットは、ISOかEMVのいずれかが使用されます。**CapIsoEmvMode**プロパティは、スマートカードリーダライタがサポートするモードを示します。**IsoEmvMode**プロパティは、スマートカードリーダライタで使用するモードの値を示しています。

このプロパティは以下の値が設定されます。

値	意味
SC_MODE_ISO	現在使用中のフォーマットをメッセージとして送るAPDUは、ISOの標準に一致します。
SC_MODE_EMV	現在使用中のフォーマットをメッセージとして送るAPDUは、EMVの標準に一致します。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CapIsoEmvMode**プロパティ, **InterfaceMode**プロパティ,

SCPresentSensorプロパティ

形式　LONG SCPresentSensor;

説明　このプロパティは、スマートカードリーダライタに搭載されているスロットのうちの一つでスマードカードが検知されており、データの取得・設定が可能である状態であることを示します。このプロパティは、スマートカード検知のセンサがサポートされ、CapSCPresentSensor がそのスロットがセンサをサポートしていると示しているときに有効です。

スマートカードリーダライタは、スマートカード検知センサをサポートするかしないかに関わらず、LSBで指定される利用可能な(デフォルトのスロットとして指定された)最低1つのスロットを持ちます。

このプロパティは、ビット0(LSB)がスロット0(デフォルト), ビット1がスロット1、ビット2がスロット2 等の論理和で設定されます。ビットが1である場合、センサはスマートカードが検知されていることを示します。検知していない場合、ビットの値は 0 となります。

このプロパティは、Openメソッドにより初期化されます。

参照　CapSCPresentSensorプロパティ

SCSlotプロパティ

形式　LONG SCSlot;

説明　このプロパティはスマートカード取り込み可能なスロットを示します。アプリケーションは、CapSCSlotsプロパティの値の1つにSCSlotプロパティをセットすることにより使用するスロットを選択することができます。サービスオブジェクトは、常に1までLSBセットによって利用可能な(デフォルト(スロット0)として指定された)最低1つのスロットを示します。

このプロパティは、以下のように、CapSCSlotsプロパティで設定される値の一つが設定されます。

ビット0(LSB)は0(デフォルト);ビット1、スロット1;ビット2、スロット2;など

このプロパティは、Openメソッドにより初期化され、デフォルト値として、slot 0 が設定されます。

参照　CapSCSlotsプロパティ

TransactionInProgressプロパティ

形式　BOOL TransactionInProgress;

説明　TRUEの場合は、スマートカードを検知し、スマートカードとデータのやり取りをしています。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**SCPresentSensor**プロパティ

TransmissionProtocolプロパティ

形式　LONG TransmissionProtocol;

説明　このプロパティは、スマートカードリーダライタが現在サポートしている、ISO 7816-3送信プロトコル、T=0(非同期半二重の文字送信プロトコル)あるいはT=1(非同期半二重のブロック送信プロトコル)のタイプを示します。このプロパティは、スマートカードリーダライタと通信するために使用されている、現在選択された送信プロトコルが設定されています。

このプロパティは**CapTransmissionProtocol**プロパティで設定されるサポートするモードを指定します。

このプロパティは以下の値が設定されます。

値	意味
SC_TRANS_PROTOCOL_T0	非同期半二重の文字送信プロトコルモード
SC_TRANS_PROTOCOL_T1	非同期半二重のブロック送信プロトコルモード

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CapInterfaceMode**プロパティ

メソッド

BeginInsertion メソッド

形式 LONG BeginInsertion (LONG *Timeout*);

パラメータ	説明
Timeout	メソッドが失敗するまでのミリ秒単位

*Timeout*パラメータに 0 が指定された場合、メソッドは、スマートカードの取り込み処理を実行し、即時に適切なステータスを戻します。OPOS_FOREVER(-1)の場合、このメソッドはスマートカードが取り込まれるかエラーが発生するまで待ちます。

説明 接触型、非接触型スマートカードリーダライタで、スマートカードの取り込み処理をするために呼び出します。

このメソッドが呼ばれると、スマートカードリーダライタをスマートカード取り込み許可の状態にします。このメソッドは、スマートカード取り込みを制御するEndInsertionメソッドとセットで呼び出します。

スマートカードリーダライタを取り込みモードに設定できない場合、エラーが返されます。そうでなければ、設定されたタイムアウト値が経過してもスマートカードが取り込まれないか、エラーがスマートカードリーダライタから通知されるまで、サービスオブジェクトは、スマートカード取り込みをモニターします。この場合、サービスオブジェクトは、適切なエラーを返し、スマートカード取り込みモードのままとなります。これは、デバイスの動作確認メカニズムをかえずに、アプリケーションがユーザとの対話処理や BeginInsertionメソッドを再発行できるよう取り込みモードを維持します。

戻り値 次の値のいずれかが戻され、ResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	カードが正常に取り込まれました。
OPOS_E_BUSY	出力中で実行できません。
OPOS_E_ILLEGAL	スマートカードリーダライタが存在しないか、無効な *Timeout*値が設定されました。
OPOS_E_TIMEOUT	スマートカードが適切に取り込まれずに、設定された *Timeout*値が経過しました。
OPOS_E_EXTENDED	内容についてはResultCodeExtendedを参照してください。

その他　　**ResultCode**の項目を参照してください。

参照　　**EndInsertion**メソッド、**BeginRemoval**メソッド、**EndRemoval**メソッド

BeginRemoval メソッド

形式 LONG BeginRemoval (LONG *Timeout*);

パラメータ	説明
Timeout	メソッドが失敗するまでのミリ秒単位

0 が指定された場合、メソッドは、スマートカードの排出処理を実行し、即時に適切なステータスを戻します。OPOS_FOREVER(-1)の場合、このメソッドはスマートカードが排出されるかエラーが発生するまで待ちます。

説明 スマートカードの排出処理をするために呼びだします。

このメソッドが呼ばれると、接触型、非接触型スマートカードリーダライタのいずれかにおいてもスマートカード排出モードとなります。このメソッドは、スマートカード排出処理終了を制御する、**EndRemoval** メソッドとセットで呼び出します。

スマートカードリーダライタが接触型で、挿入口にセンサを有する場合、スマートカードが抜き取られることで正常終了します。スマートカードリーダライタが非接触タイプ(センサの無い)の場合、カード検知位置から離れた場合、正常終了します。

カード排出モードに設定できない場合、アプリケーションにエラーが返されます。さもなければ、設定されたタイムアウト値が経過してもスマートカードが排出されない場合、あるいはエラーがスマートカードリーダライタから通知されるまで、サービスオブジェクトはスマートカードをモニターします。この場合、サービスオブジェクトは、適切なエラーを返します。これは、デバイスの動作確認メカニズムをかえずに、アプリケーションがユーザとの対話処理や **BeginRemoval**メソッドを再発行できるよう排出モードを維持します。

もしスマートカードリーダライタを排出モードに設定できなければ、アプリケーションにエラーが返されます。それ以外の場合、コントロールは次のいずれかの状況まで排出状況をモニターします。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	カードが抜き取られました。
OPOS_E_BUSY	出力中で実行できません。
OPOS_E_ILLEGAL	スマートカードリーダライタが存在しないか、無効な *Timeout*値が設定されました。

OPOS_E_TIMEOUT	スマートカードが適切に排出されずに、設定された *Timeout*値が経過しました。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

参照　　**BeginInsertion**メソッド、**EndInsertion**メソッド、**EndRemoval**メソッド

EndInsertion メソッド

形式 LONG EndInsertion ();

説明 スマートカードの取り込み終了のために呼び出します。

このメソッドが呼び出されると、スマートカードリーダライタは、スマートカード取り込みモードを終了します。スマートカードが取り込まれていない場合、エラーが返されます。このメソッドは、接触型、非接触型スマートカードリーダライタのいずれかにおいても、スマートカード取り込み処理を制御するため、**BeginInsertion**メソッドとセットで呼び出します。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	カードが正常に取り込まれました。
OPOS_E_ILLEGAL	スマートカードリーダライタがスマートカード取り込みモードではありません。
OPOS_E_FAILURE	スマートカードリーダライタにスマートカードが入っていません。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

参照 **BeginInsertion**メソッド、**BeginRemoval**メソッド、**EndRemoval**メソッド

EndRemoval メソッド

形式　LONG EndRemoval ();

説明　スマートカード排出処理のために呼び出します。

デバイスは、接触型、非接触型いずれかの場合でも、スマードカード排出モードを終了します。スマートカードが残っている場合、エラーが返ります。このメソッドは、スマートカード排出処理のため、**BeginRemoval**メソッドとセットで呼び出します。

アプリケーションは、スマートカードがいつ排出されたかを特定できるようにスマートカードリーダライタのセンサを使用したい場合には、**BeginRemoval**メソッド成功後直ぐに、このメソッドを呼んだ方がよいでしょう。

代わりに、アプリケーションはこのメソッドを呼ぶ前に、ユーザに注意を促して、キーの押下を待っても良いでしょう

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	カードが正常に排出されました。
OPOS_E_ILLEGAL	スマートカードリーダライタがスマートカード排出モードではありません。
OPOS_E_FAILURE	スマートカードリーダライタにスマートカードが残っています。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

参照　**BeginInsertion**メソッド、**EndInsertion**メソッド、**BeginRemoval**メソッド

ReadDataメソッド

形式 LONG ReadData (LONG *Action*, LONG* *Count*, BSTR *Data*);

パラメータ	説明
Action	スマートカードに処理させる、データ処理方法。
Count	スマートカードから取得したデータ（バイト）数。
Data	スマートカードから取得したデータ。

詳細はBinaryConversionプロパティを参照してください。

説明 スマートカードからのデータを取得します。

*Action*は、以下のいずれかのデータを指定します。

値	意味
SC_READ_DATA	スマートカードのデータ領域から取得したデータが*Data*に設定されます。
SC_READ_PROGRAM	スマートカードのプログラム実行可能領域から取得した実行プログラムが *Data*に設定されます。
SC_EXECUTE_AND_READ_DATA	スマートカードの実行可能プログラムによって処理されたデータが *Data*に設定されます。この値が設定されたときに、スマートカード上の実行プログラムは実行を開始する場合があり、処理後のデータを*Data*に設定します。
SC_XML_READ_BLOCK_DATA	アプリケーションに送るXMLデータが *Data*に設定されます。取得したデータはアプリケーションによって解析されます。

戻り値 次の値のいずれかが戻され、ResultCodeプロパティにもその値が格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_CLAIMED	デバイスは他のプロセスにより排他アクセスされています。

OPOS_E_ILLEGAL	無効な *Action*, *Count*が指定されました。
参照	**WriteData**メソッド

WriteDataメソッド

形式　LONG WriteData (LONG *Action*, LONG *Count*, BSTR *Data*);

パラメータ	説明
Action	スマートカードに処理させる、データ処理方法。
Count	スマートカードに設定されるデータ（バイト）数。
Data	スマートカードに設定されるデータ。

詳細はBinaryConversionプロパティを参照してください。

説明　スマートカードにデータを書き込みます。

*Action*は、以下のいずれかのデータを指定します。

値	意味
SC_STORE_DATA	*Data*の値をスマートカードのデータ領域に設定します。
SC_STORE_PROGRAM	*Data*の値は、実行可能プログラムであり、スマートカードのプログラム実行可能領域に設定します。
SC_EXECUTE_DATA	*Data*の値は、スマートカードの実行可能プログラムによって処理可能なデータです。この値が設定されたときに、スマートカード上の実行プログラムは実行を開始し、*Data*の値を用いて処理を行います。
SC_XML_BLOCK_DATA	*Data*の値は XMLデータで、デバイスによって解析されます。
SC_SECURITY_FUSE	スマートカードに格納されているデータが上書きされるのを防ぐために、セキュリティヒューズを活性化するでしょう。
SC_RESET	スマートカードの状態を"電源ON"状態となるようリセットして、アプリケーションコマンドを実行可能な状態にします。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにもその値が格納されます。

値	意味

OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_CLAIMED	デバイスは他のプロセスにより排他アクセスされています。
OPOS_E_ILLEGAL	無効な *Action*, *Count*が指定されました。

参照　**ReadData**メソッド

イベント

DataEvent イベント

形式　　void DataEvent (LONG *Status*);

説明　　デバイスからデータが入力されたことをアプリケーションに通知します。このイベントが通知されるまえに、カードのデータは適切なプロパティに設定されます。

ErrorEventイベント

形式　void ErrorEvent(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**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**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。

ResultCode が OPOS_E_EXTENDED の場合、**RsultCodeExtended** は次のいずれかの値をとります。

値	意味
ESC_READ	読み込みエラー。
ESC_WRITE	書き込みエラー。
ESC_TORN	スマートカードが不意にスマートカードリーダライタから取り除かれました。 注:**CapCardErrorDetection**プロパテイが TRUE の場合、この値が設定されます。
ESC_NO_CARD	カードに対して処理を行おうとしましたが、カードがありませんでした。

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。

値	意味
OPOS_EL_OUTPUT	非同期出力を処理している間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。

アプリケーションはエラーイベントハンドラ内で*pErrorRespons*に次のいずれかの値を設定できます。

値	意味
OPOS_ER_RETRY	入力または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 以外の場合、使用できないことがあります。*ErrorLocus*が OPOS_EL_OUTPUTの場合のデフォルトです。
OPOS_ER_CLEAR	非同期出力、バッファリングされている出力データ（全ての非同期出力を含む)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 のときに限り設定されます。エラーを認識し、デバイスに処理の継続を指示します。デバイスはエラー状態のままですが、**DataEventEnabled** プロパティの設定にしたがって、別の**DataEvent** が引き続き通知されます。すべての入力データが通知されて、**DataEventEnabled** が再びTRUEに設定されたとき、*ErrorLocus*がOPOS_EL_INPUT に設定された新たな**ErrorEvent** 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　エラーが検知され、アプリケーションによる適切な応答がエラー状態の処理に必要なことをアプリケーションに通知します。

入力エラーは、新しく取り込まれたスマートカードからデータを読む間に発生します。**DataEventEnabled** がTRUEにならないかぎりこのイベントは通知されないので、アプリケーションは通常のシーケンスでイベントを処理できます。エラー状態は、イベントが通知される前に、適切なプロパティに設定されます。

出力エラーは、非同期で **WriteData**メソッドを実行中に発生します。エラー状態は、イベントが通知される前に、適切なプロパティに設定されます。

参照　CapCardErrorDetectionプロパティ、**SCPresentSensor**プロパティ、**ReadData**プロパティ、**WriteData**プロパティ

OutputCompleteEventイベント

形式　　void OutputCompleteEvent(LONG *OutputID*);

説明　　キューに登録された出力要求が正常に実行されたことをアプリケーションに通知します。出力要求のIDは*OutputID*パラメータに格納されます。このイベントは、リクエストのデータが送信され、サービスオブジェクトがデバイスによって正常に処理されたことを確認するとキューに入れられます。

StatusUpdateEventイベント

形式　　void StatusUpdateEvent(LONG *Status*);

説明　　スマートカードリーダライタの状態が変化したときに通知されます。具体的には、スマートカードリーダ内のスマートカード状態が変化したときに通知されます。

*Status*には、次の値のいずれかに設定されます。

値	意味
SC_SUE_NO_CARD	スマートカードリーダライタはスマートカードを検知していません。
SC_SUE_CARD_PRESENT	カードがスマートカードリーダライタ内にあります。

第　２５章
バイオメトリクス

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.10	Boolean	R/W	Open
BinaryConversion	1.10	Long	R/W	Open
CapCompareFirmwareVersion	1.10	Boolean	R	Open
CapPowerReporting	1.10	Long	R	Open
CapStatisticsReporting	1.10	Boolean	R	Open
CapUpdateFirmware	1.10	Boolean	R	Open
CapUpdateStatistics	1.10	Boolean	R	Open
CheckHealthText	1.10	String	R	Open
Claimed	1.10	Boolean	R	Open
DataCount	1.10	Long	R	Open
DataEventEnabled	1.10	Boolean	R/W	Open
DeviceEnabled	1.10	Boolean	R/W	Open & Claim
FreezeEvents	1.10	Boolean	R/W	Open
OpenResult	1.10	Long	R	無し
OutputID	1.10	Long	R	Open
PowerNotify	1.10	Long	R/W	Open
PowerState	1.10	Long	R	Open
ResultCode	1.10	Long	R	無し
ResultCodeExtended	1.10	Long	R	Open
State	1.10	Long	R	無し
ControlObjectDescription	1.10	String	R	無し
ControlObjectVersion	1.10	Long	R	無し
ServiceObjectDescription	1.10	String	R	Open
ServiceObjectVersion	1.10	Long	R	Open
DeviceDescription	1.10	String	R	Open
DeviceName	1.10	String	R	Open

専用

Algorithm	1.10	Boolean	R/W	Open & Claim
AlgorithmList	1.10	String	R	Open
BIR	1.10	String	R	Open & Claim
CapPrematchData	1.10	Boolean	R	Open
CapRawSensorData	1.10	Boolean	R	Open
CapCapRealTimeData	1.10	Boolean	R	Open
CapSensorColor	1.10	Long	R	Open
CapSensorOrientation	1.10	Long	R	Open
CapSensorType	1.10	Long	R	Open
CapTemplateAdaptation	1.10	Boolean	R	Open
RawSensorData	1.10	String	R	Open & Claim
RealTimeDataEnabled	1.10	Boolean	R/W	Open
SensorBPP	1.10	Long	R	Open
SensorColor	1.10	Long	R/W	Open
SensorHeight	1.10	Long	R	Open
SensorOrientation	1.10	Long	R/W	Open,Claim,& Enable
SensorType	1.10	Long	R/W	Open,Claim,& Enable
SensorWidth	1.10	Long	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.10	無し
Close	1.10	Open
ClaimDevice	1.10	Open
ReleaseDevice	1.10	Open & Claim
CheckHealth	1.10	Open, Claim, & Enable
ClearInput	1.10	Open & Claim
ClearInputProperty	1.10	Open & Claim
ClearOutput	1.10	適用外
CompareFirmwareVersion	1.10	Open, Claim, & Enable
DirectIO	1.10	Open & Claim
ResetStatistics	1.10	Open, Claim, & Enable
RetrieveStatistics	1.10	Open, Claim, & Enable
UpdateFirmware	1.10	Open, Claim, & Enable
UpdateStatistics	1.10	Open, Claim, & Enable

専用		
BeginEnrollCapture	1.10	Open, Claim,＆Enable
BeginVerifyCapture	1.10	Open, Claime,＆Enable
EndCapture	1.10	Open, Claim,＆Enable
Identify	1.11	Open, Claim,＆Enable
IdentifyMatch	1.11	Open, Claim,＆Enable
ProcessPrematchData	1.10	Open, Claim,＆Enable
Verify	1.10	Open, Claim,＆Enable
FARPrecedence	1.10	Open, Claim,＆Enable
VerifyMatch	1.10	Open, Claim,＆Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.5	Open,Claim,&Enable
DirectIOEvent	1.5	Open & Claim
ErrorEvent	1.5	Open,Claim,&Enable
OutputCompleteEvent	1.5	Open,Claim,&Enable
StatusUpdateEvent	1.5	Open,Claim,&Enable

概説

バイオメトリクスコントロールのOLEプログラマティックIDは“OPOS.Biometorics”です。

このデバイスは、第1.10版で追加されました。

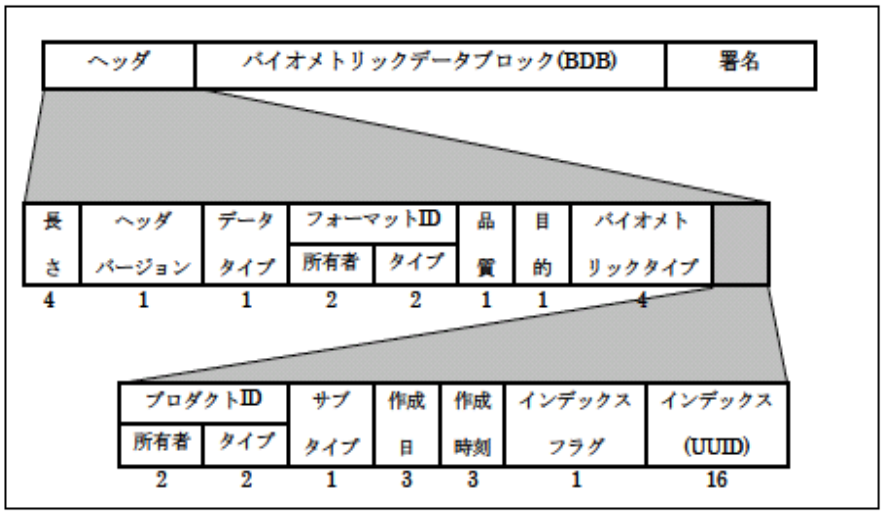
機能

バイオメトリクスデバイスは次の機能を持っています。

デバイスはバイオメトリクスセンサーからバイオメトリクスのデータをキャプチャーします。バイオメトリクスのデータは、バイオメトリックデータサンプルかバイオメトリックテンプレートが順番に一つ以上入ったBiometorics Data Block（BDB）を一つ以上含むBiometorics Information Recoed（BIR）の形式で構成されます。

この仕様書では､バイオメトリクスのユーザー登録データを述べる際、（BioAPI[1]で採用された）テンプレートの名称を使用します。BIRの名称で、アプリケーションに渡される次のようなバイオメトリックデータを示します。登録データの他に、生データ、中間データ、照合や識別に用意された処理済データ。一般的に、アプリケーションに永続的に保存されるデータのみが、登録用に生成されるBIR（i.e テンプレート）となります。不透明なBDBのフォーマットは、ヘッダのフォーマットフィールドで指定されます。これには、標準フォーマットと独自フォーマットがあります。BDBは暗号化されているかもしれません。デジタル署名はオプションであり、転送と保存の間でデータの整合性を保証するために使用されます。存在するなら、ヘッダ＋BDB上で算出されます。標準BIRフォーマットでは、署名は（フォーマットが標準化された時点に決定される）標準形式となります。（現時点で存在しているすべての）独自のBIRフォーマットでは、署名にはサービスに適合するどのような形式でも利用できます。このため、署名のC言語構造体は存在しません。BIRデータタイプは、BIRが署名されているか暗号化されているかを示します。

[1] BioAPIコンソーシアムによって定義されています。(www.bioapi.org)



デバイスは、登録を目的とするバイオメトリックデータをキャプチャーします。登録を意図する際、生成された最終BIRは高水準の品質が要求されます。一般的に、BIRは一連のバイオメトリック取得データの集合体です。

デバイスは、照合を目的とするバイオメトリックデータをキャプチャーします。照合の際には、登録での品質までは要求されません。

デバイスは、False Accept Rate（FAR：誤合致率）とFalse Reject Rate（FRR：本人拒否率）で特徴付けられるエラー率の範囲内で二つのBIRが一致するかを決定する機能を持ちます。FARは、二つの不一致バイオメトリックサンプルが一致すると誤認識されるエラー率の閾値です。FRRは、二つの一致するバイオメトリックサンプルが一致していないと誤認識されるエラー率の閾値です。

デバイスは、BIRのサンプル集合とBIRを比較し、識別のための類似度リストを作成する機能を持ちます。

いくつかのバイオメトリクスデバイスは、以下の追加機能を持つものもあります。

デバイスは、デバイスでキャプチャーされると“リアルタイム”に未加工バイオメトリックデータを返します。このケイパビリティがTRUEで、アプリケーションがRealTimeDataEnabledプロパティをTRUEに設定すると、バイオメトリックイメージの一部を表すSensorBPP、SensorColor、SensorHeight、SensorWidthで定義された未加工イメージがキャプチャーされるたびに、一連のStatusUpdateEventがエンキューされます。

モデル

バイオメトリクスデバイスの使用モデルです。

デバイスをOpenしClaimします。

DeviceEnabledプロパティとDataEventEnabledプロパティにTRUEをセットします。

非同期実行のBeginVerifyCaptureかBegineEnrollCaptureメソッドの呼び出しによりバイオメトリクスデータの取得を開始します。それらのメソッドは、各バイオメトリクスデバイスの固有の方法でデータの取得を開始するために、バイオメトリクスセンサーを動作させます。取得したバイオメトリクスデータは、BIRプロパティに保存されます。BIRデータは、比較と判定のためのIdentifyMatchとVerifyMatchメソッドで利用されます。将来の照合のためのBIRの保管処理はアプリケーションに依存します。

Verifyメソッドで同期的に照合処理を行い、Identifyメソッドも同期的に識別処理を行います。

デバイスがリアルタイムにバイオメトリックサンプルを提供する機能があり（CapRealTimeDataがTRUE）、RealTimeDataEnabledがTRUEなら、バイオメトリクスデータはRawSensorDataプロパティで一連の部分バイオメトリクスデータとしてアプリケーションに渡され、バイオメトリックサンプルが全て取得されるまでStatusdUpdateEventがアプリケーションに通知されます。RawSensorDataはキューイングされず、StatusUpdateEventの受理からデータをキャプチャーすることはアプリケーションの責任となります。

バイオメトリクスデバイスは、イベント駆動入力の一般的な入力モデルに従います。

コントロールは、入力を受信したときにDataEventを発生させます。

AutoDisableプロパティがTRUEなら、コントロールはDataEventがキューイングされた時に、自動的にディセーブルとなります。

DataEventEnabledプロパティがTRUEの時、キューイングされたDataEventはアプリケーションに通知されます。コントロールはこのイベントを発生させる直前に、データをプロパティにコピーし、さらなるデータイベントを発生させないために、DataEventEnabledプロパティをFALSEに設定します。これにより、アプリケーションが現在の入力の処理や関連するプロパティの処理をしている間、コントロールはその後の入力データをキューイングするようになります。アプリケーションが現在の入力データの処理を終え、次のデータ処理の準備ができたとき、DataEventEnabledプロパティをTRUEに設定することにより、再びデータイベントが通知されるようになります。

入力データの取り込みまたは処理中にコントロールでエラーが発生した場合、ErrorEvent（またはイベント）がキューイングされ、DataEventEnabledプロパティがTRUEの時、アプリケーションに通知されます。

DataCountプロパティをリードすることで、コントロールにキューイングされたDataEventの数を得られます。

コントロールにキューイングされたすべての入力データはClearInputメソッドを呼ぶことにより削除されます。

入力モデルから以下の点が外れています。

BeginEnrollCaptureかBeginVerifyCaptureの呼び出しで、バイオメトリクスデータの読み取りが開始されます。

もしバイオメトリクスデバイスがEndCaptureメソッドを呼び出して処理を終了した場合、DataEventとErrorEventイベントは発行されません。

デバイスの共有

バイオメトリクスコントロールは、下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければいけません。

アプリケーションは、いくつかの専用プロパティにアクセスする前に、排他アクセス権を獲得し、イネーブルしなければなりません。

アプリケーションは、デバイスを操作するメソッドを呼び出す前に、イネーブルしなければなりません。

使用上の詳細な必要条件は、「一覧」を参照してください。

プロパティ

Algorithmプロパティ R/W

形式 LONG Algorithm;

説明 バイオメトリクステンプレートを作成するために使われるバイオメトリクスアルゴリズムです。この値は、**AlgorithmList**の中の値を指定するために設定されます。

例)

値	意味
0	初期値
1	**AlgorithmList**の最初のデータ
2	**AlgorithmList**の2番目のデータ

このプロパティは、**Claim**メソッドで初期化されます。

参照 **AlgorithmList** プロパティ

AlgorithmListプロパティ

形式 BSTR AlgorithmList;

説明 デバイスによってサポートされるアルゴリズムのコンマ区切りのリストが含まれます。

参照 **Algorithm** プロパティ

BIR プロパティ[2] R/W

形式　BSTR BIR;

説明　この仕様書では、バイオメトリクスのユーザー登録データを述べる際、テンプレートの名称を使用します。**BIR**の名称で、アプリケーションに渡される次のようなバイオメトリックデータを示します。登録データの他に、生データ、中間データ、照合や識別に用意された処理済データ。一般的に、アプリケーションに永続的に保存されるデータのみが、登録用に生成される**BIR**（i.e テンプレート）となります。不透明な**BDB**のフォーマットは、ヘッダのフォーマットフィールドで指定されます。これには、標準フォーマットと独自フォーマットがあります。**BDB**は暗号化されているかもしれません。デジタル署名はオプションであり、転送と保存の間でデータの整合性を保証するために使用されます。存在するなら、ヘッダ＋**BDB**上で算出されます。標準**BIR**フォーマットでは、署名は（フォーマットが標準化された時点に決定される）標準形式となります。（現時点で存在しているすべての）独自の**BIR**フォーマットでは、署名にはサービスに適合するどのような形式でも利用できます。このため、署名のC言語構造体は存在しません。**BIR**データタイプは、**BIR**が署名されているか暗号化されているかを示します。

BeginEnrollCapture、**BeginVerifyCapture**、**Verify**のメソッドによって得られる処理済バイオメトリックデータは、正常終了時にこのプロパティに格納されます。

このデータのフォーマットは、**BinaryConversion**プロパティの値に依存します。詳細は**BinaryConversion**プロパティを参照してください。

参照　**BeginEnrollCapture** メソッド, **BeginVerifyCapture** メソッド, **Verify** メソッド

[2] Biometrics Information Record(BIR)は、BioAPIコンソーシアムで定義されています。

CapPrematchDataプロパティ

形式 BOOL CapPrematchData;

説明 TRUEのとき、OPOSサービスは、スマートカードにストアされた事前照合データを元に処理済**BIR**を生成するMOC(MATCH-On-Card)スマートカード技術を利用できる機能があります。

参照 **ProcessPrematchData**メソッド

CapRawSensorDataプロパティ *第1.12版にて修正*

形式 BOOL CapRawSensorData;

説明 TRUEのとき、OPOSサービスはバイオメトリクスセンサーから未処理の生データを返すことができます。

参照 **RawSensorData**メソッド

CapRealTimeDataプロパティ *第1.12版にて修正*

形式 BOOL CapRealTimeData;

説明 TRUEのとき、デバイスは、バイオメトリクス情報が採取されるや否や（“リアルタイムで”）生バイオメトリクスデータを通知する事ができます。**CapRawSensorData**がtrueである事で、リアルタイムにデータが、**RawSensorData**プロパティにストアされることで、**CapRawSensorData**プロパティがfalseになるならば、このプロパティの値もfalseになります。

このプロパティはOpenメソッドで初期化されます。

参照 **RawSensorData**プロパティ、**SensorBPP**プロパティ、**SensorColor**プロパティ、**SensorHeight**プロパティ、**SensorWidth**プロパティ

CapSensorColorプロパティ

形式　LONG CapSensorColor;

説明　この機能は、このデバイスが白黒以外のイメージフォーマットをサポートするかどうかを表します。

以下の値の論理ORの組み合わせがプロパティに格納されます。

値	意味
BIO_CSC_MONO	白黒
BIO_CSC_GRAYSCALE	グレースケール
BIO_CSC_16	16色
BIO_CSC_256	256色
BIO_CSC_FULL	フルカラー

このプロパティは、**Open**メソッドにより初期化されます。

CapSensorOrientationプロパティ

形式　LONG CapSensorOrientation;

説明　この機能は、処理の前に画像を回転させるセンサーの能力を示します。

以下の値の論理ORの組み合わせがプロパティに格納されます。

値	意味
BIO_CSO_NORMAL	0°
BIO_CSO_RIGHT	90°
BIO_CSO_INVERTED	180°
BIO_CSO_LEFT	270°

このプロパティは、**Open**メソッドにより初期化されます。

CapSensorTypeプロパティ

形式 LONG CapSensorType;

説明 この機能は、接続されたセンサーによって採取されることができるバイオメトリクスデータの型を示します。

以下の値の論理ORの組み合わせがプロパティに格納されます。

値	意味
BIO_CST_FACIAL_FEATURES	人相
BIO_CST_VOICE	音声
BIO_CST_FINGERPRINT	指紋
BIO_CST_IRIS	虹彩
BIO_CST_RETINA	網膜
BIO_CST_HAND_GEOMETRY	掌紋
BIO_CST_SIGNATURE_DYNAMICS	筆跡
BIO_CST_KEYSTOKE_DYNAMICS	キー押下
BIO_CST_LIP_MOVEMENT	口の動作
BIO_CST_THERMAL_FACE_IMAGE	赤外線センサーの顔のイメージ
BIO_CST_THERMAL_HEAD_IMAGE	赤外線センサーの頭のイメージ
BIO_CST_GAIT	歩幅
BIO_CST_PASSWORD	暗証コード

このプロパティは、**Open**メソッドにより初期化されます。

参照 **SensorType**プロパティ

CapTemplateAdaptationプロパティ

形式　BOOL CapTemplateAdaptation;

説明　TRUEのとき、OPOSサービスは、サンプルBIRかキャプチャしたBIRから取得した情報を基に照合BIRを更新した適応BIRを返すことができます。適応処理の目的は、バイオメトリックデータの過時変化に伴い、参照BIRを保守することです。

この機能は、選択されたアルゴリズムに依存するので、Open、Claim、DeviceEnabledをTRUEにした後に値が確定します。

参照　SensorTypeプロパティ

RawSensorDataプロパティ　*第1.12版にて修正*

形式　BSTR RawSensorData;

説明　左上から右下へ走査するスキャンラインの生ピクセルデータとしてバイオメトリクスイメージを保持します。SensorHeightとSensorWidthはピクセルの数を定義します。SensorBPPは１ピクセルを構成するビット数を定義します。SensorColorは、ピクセルデータの解釈を定義します。CapRawSensorDataがfalseのとき、このプロパティの値は意味を持ちません。このデータのフォーマットは、BinaryConversionプロパティの値に依存します。詳細はBinaryConversionプロパティを参照してください。

参照　CapRawSensorDataプロパティ、CapRealTimeDataプロパティ、RealTimeDataEnabledプロパティ、SensorBPPプロパティ、SensorColorプロパティ、SensorHeightプロパティ、SensorWidthプロパティ、

RealTimeDataEnabledプロパティ　R/W　　*第1.12版にて修正*

形式　　BOOL RealTimeDataEnabled;

説明　　この値がtrueの場合、その時にStatusUpdateEventが、バイオメトリクスの採取が完了されるまで、更新される断片的なバイオメトリクスデータが採取処理される毎に通知されます。一方で、バイオメトリクスの採取が完了された時に、採取されたバイオメトリクスデータはひとつのDataEventとしてキューイングされます。

RealTimeDataEnabledをセットすると、その後のBeginEnrollCaptureかBeginVerifyCaptureメソッドが実行されるまで、システムの動作は変化しません。これは、BeginEnrollCapture－EndCaptureかBeginVerifyCapture－EndCaptureの間の状態変化による混乱を防止するためです。

このプロパティは、Openメソッドにより FALSEに初期化されます。

戻り値　　次の値のいずれかが戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定された。
OPOS_E_ILLEGAL	CapRealTimeDataがFALSEです。

参照　　CapRealTimeDataプロパティ、RawSensorDataプロパティ、SensorBPPプロパティ、SensorColorプロパティ、SensorHeightプロパティ、SensorWidthプロパティ、BeginEnrollCaptureメソッド、BeginVerifyCaptureメソッド、EndCaptureメソッド

SensorBPPプロパティ

形式　　LONG SensorBPP;

説明　　RawSensorDataのBit Per Pixel(BPP)エンコーディングを保持します。

SensorColorプロパティ　R/W　　*第1.11版で更新*

形式　　LONG SensorColor;

説明　　このプロパティは、連続するバイオメトリクスの取得処理でのイメージ取得モードを選択するために使われます。全ての“色”やカラーイメージタイプの機能が無いSensorTypeのデバイスでは、機能しません。SensorColorプロパティを変更しても、RawSensorDataプロパティやBIRプロパティに既に格納されているデータには影響しません。

以下の値の何れかひとつがプロパティに格納されます。

値	意味
BIO_SC_MONO	白黒
BIO_SC_GRAYSCALE	グレースケール
BIO_SC_16	16色
BIO_SC_256	256色
BIO_SC_FULL	フルカラー（XMLデータブロックモード)

CapSensorTypeプロパティで定義されている値がセットされます。

このプロパティは、Openメソッドにより初期化されます。

参照　　CapSensorColorプロパティ、RawSensorDataプロパティ、SensorBPPプロパティ、SensorHeightプロパティ、SensorWidthプロパティ、

SensorHeightプロパティ

形式　　LONG SensorHeight;

説明　　RawSensorDataの高さをピクセル数で保持します。

SensorOrientationプロパティ R/W *第1.11版で更新*

形式 LONG SensorOrientation;

説明 **BIR**の作成前の入力センサーデータの回転方向を格納します。

値	意味
BIO_SO_NORMAL	0°
BIO_SO_RIGHT	90°
BIO_SO_INVERTED	180°
BIO_SO_LEFT	270°

このプロパティは、デバイスが**Open**され**Claim**され、**DeviceEnabled**にTRUEをセットする前に値が更新されます。

CapSensorOrientationプロパティで定義された値がセットされます。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定された。
OPOS_E_ILLEGAL	**CapSensorOrientation**プロパティがFALSEです。

参照 **CapSensorOrientation**プロパティ

SensorTypeプロパティ　R/W　　*第1.11版で更新*

形式　　LONG SensorType;

説明　　接続されるバイオメトリクスセンサーの型を保持します。

値	意味
BIO_CST_FACIAL_FEATURES	人相
BIO_CST_VOICE	音声
BIO_CST_FINGERPRINT	指紋
BIO_CST_IRIS	虹彩
BIO_CST_RETINA	網膜
BIO_CST_HAND_GEOMETRY	掌紋
BIO_CST_SIGNATURE_DYNAMICS	筆跡
BIO_CST_KEYSTOKE_DYNAMICS	キー押下
BIO_CST_LIP_MOVEMENT	口の動作
BIO_CST_THERMAL_FACE_IMAGE	赤外線センサーの顔のイメージ
BIO_CST_THERMAL_HEAD_IMAGE	赤外線センサーの頭のイメージ
BIO_CST_GAIT	歩幅
BIO_CST_PASSWORD	暗証コード

CapSensorTypeプロパティで定義された値がセットされます。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定された。
OPOS_E_ILLEGAL	**CapSensorOrientation**プロパティがFALSEです。

参照　　**CapSensorType**プロパティ

SensorWidthプロパティ

形式　　LONG SensorWidth;

説明　　RawSensorDataプロパティの幅をピクセル数で格納します。

参照　　RawSensorDataプロパティ

メソッド

BeginEnrollCapture メソッド　　*第1.11版で更新*

形式　LONG BeginEnrollCapture (BSTR *ReferenceBIR*, BSTR *Payload*);

パラメータ	説明
ReferenceBIR	適応（更新）のためのオプション**BIR**。空文字列なら無視されます。
Payload	BSP[3] (Biometrics Service Provider)に格納されるデータ。空文字列なら無視されます。

説明　登録のためにバイオメトリクスデータのキャプチャを開始します。必須ではありませんが、登録のキャプチャーは一般的に、一連のバイオメトリクスデータのキャプチャーが集まって最終**BIR**となります。**CapTemplateAdaptation**プロパティがTRUEならオプションとして、*ReferenceBIR*により登録の適応が提供されます。*Payload*が設定されていれば、生成された**BIR**に付加されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理されました。
OPOS_E_FAILURE	*ReferenceBIR*の値が不正です。
OPOS_E_ILLEGAL	バイオメトリクスキャプチャーは既に動作しています。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

参照　**BIR**プロパティ、**CapTenplateAdaptation**プロパティ、**EndCapture**メソッド

[3] BioAPIで認証技術を提供するモジュール。

BeginVerifyCapture メソッド *第1.11版で更新*

形式 LONG BeginVerifyCapture();

説明 照合の目的でバイオメトリクスデータの採取を開始します。生成された加工済データは**BIR**に格納されます。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理されました。
OPOS_E_ILLEGAL	バイオメトリクスキャプチャーは既に動作しています。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

参照 **BIRプロパティ**、**EndCapture**メソッド

EndCapture メソッド

形式　LONG EndCapture ();

説明　バイオメトリクスデータの採取を停止（終了）します。

RealTimeDataEnabledメソッドがFALSEで、バイオメトリクスデータが採取されたなら、キャプチャしたデータは**BIR**と**RawSensorData**プロパティに格納されます。バイオメトリクスデータが採取されていなければ、**BIR**と**RawSensorData**は空文字列です。

RealTimeDataEnabledがTRUEで、**StatusUpdateEvent**によるアプリケーション通知がなされていないバイオメトリクスデータが残っているなら、残りのバイオメトリクスデータは**BIR**と**RawSensorData**プロパティに格納されます。キャプチャーされたバイオメトリクスデータがないか、すべてのバイオメトリクスデータがアプリケーションに通知されたなら、**BIR**と**RawSensorData**は空文字列です。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理されました。
OPOS_E_ILLEGAL	バイオメトリクスのキャプチャーは動作しません。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

参照　**BIR**プロパティ、**RawSensorData**プロパティ、**RealTimeDataEnabled**プロパティ、**BeginEnrollCapture**メソッド、**BeginVerifyCapture**メソッド、**DataEvent**イベント

Identify メソッド

第1.12版で更新

形式 LONG Identyfy (LONG *MaxFARRequest*, LONG *MaxFRRRequest*, BOOL *FARPrecedence*, BSTR *ReferenceBIRPopulation*, BSTR* *CandidateRanking*, LONG *Timrout*);

パラメータ	説明
MaxFARRequest	照合結果を正と判断するためのFAR基準値。
MaxFRRRequest	照合結果を正と判断するためのFRR基準値。ヌルポインタならこの値を使用しません。
FARPrecedence	両方の基準値が提供されたならば、このパラメータはどちらが優先するかを示します。BIO_FAR_PRECEDENCEなら*MaxFARRequested*が優先されます。BIO_FRR_PRECEDENCEなら*MaxFRRRequested*が優先されます。
ReferenceBIRPopulation	識別判定に使用される**BIR**の配列。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion**を参照してください。
CandidateRanking	近い順にリストされた*ReferenceBIRPopulation*の**BIR**インデックスの配列。
Timeout	失敗せずに正常にバイオメトリクスのキャプチャーを実行する最大ミリ秒時間。

説明 このファンクションは、指定された時間内で接続されたデバイスからバイオメトリクスデータをキャプチャーし、*ReferenceBIRPopulation*の集合と比較します。そして*CandidateRanking*に、*ReferenceBIRPopulation*のインデックスを近い順の配列で返します。もし一致するものが無ければ、ゼロの要素で配列が返されます。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理されました。
OPOS_E_FAILURE	*ReferenceBIRPopulation*が不正です。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

参照　　BIRプロパティ、CapTenplateAdaptationプロパティ、EndCaptureメソッド

IdentifyMatch メソッド *第1.12版で更新*

形式 LONG **IdentyfyMatch** (**LONG** *MaxFARRequest*, **LONG** *MaxFRRRequest*, **BOOL** *FARPrecedence*, **BSTR** *SampleBIR*, **BSTR** *ReferenceBIRPopulation*, **BSTR*** *CandidateRanking*);

パラメータ	説明
MaxFARRequest	照合結果を正と判断するためのFAR基準値。
MaxFRRRequest	照合結果を正と判断するためのFRR基準値。ヌルポインタならこの値を使用しません。
FARPrecedence	両方の基準値が提供されたならば、このパラメータはどちらが優先するかを示します。BIO_FAR_PRECEDENCEなら*MaxFARRequested*が優先されます。BIO_FRR_PRECEDENCEなら*MaxFRRRequested*が優先されます。
SampleBIR	識別する**BIR**。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
ReferenceBIRPopulation	識別判定に使用される**BIR**の配列。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
CandidateRanking	近い順にリストされた*ReferenceBIRPopulation*の**BIR**インデックスの配列。

説明 このファンクションは*SampleBIR*を受け取り、*ReferenceBIRPopulation*の集合と比較します。そして*CandidateRanking*に、*ReferenceBIRPopulation*のインデックスを近い順の配列で返します。もし一致するものが無ければ、ゼロの要素で配列が返されます。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理されました。。
OPOS_E_FAILURE	*ReferenceBIRPopulation*が不正です。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。

その他	**ResultCode**の項目を参照してください。

参照　**BIR**メソッド、**CapTenplateAdaptation**メソッド、**EndCapture**メソッド

ProcessPrematchData メソッド　*第1.11版で更新*

形式　LONG **ProcessPrematchData** (**BSTR** *SampleBIR*、**BSTR** *PrematchDataBIR*、**BSTR*** *ProcessedBIR*);

パラメータ	説明
SampleBIR	処理される**BIR**。
PrematchDataBIR	関連MOCライブラリーにより提供された事前比較データの**BIR**。
ProcessedBIR	新しく作成された処理済**BIR**。

説明　この関数はマッチ・オン・カード（MOC）用に使用される処理済バイオメトリックサンプルを作成します。次のマッチング処理の前にカードからの「マッチング前」データを取得することが必要なMOCインプレメンテーションを実行可能にします。スマートカードにはバイオメトリックサンプルのキャプチャ及び処理を行う機能が通常ないため、カード上のMOC機能では、サンプル取得や特徴抽出などのカード外処理はホスト側で行う必要があります。この場合、カードから取得されるマッチング前データを基にホストで処理を実行する必要があります。このメソッドの引数のフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理されました。。
OPOS_E_FAILURE	*SampleBIR*の値が不正です。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

Verify メソッド

第1.12版で更新

形式 LONG **Verify (LONG** *MaxFARRequest*, **LONG** *MaxFRRRequest*, **BOOL** *FARPrecedence*, **BSTR** *ReferenceBIR*, **BSTR*** *AdaptedBIR*, **BOOL*** *Result*, **LONG*** *FARAchived*, **LONG*** *FRRAchived*, **BSTR*** *Payload*, **LONG** *Timeout*);

パラメータ	説明
MaxFARRequest	照合結果を正と判断するためのFAR基準値。
MaxFRRRequest	照合結果を正と判断するためのFRR基準値。ヌルポインタならこの値を使用しません。
FARPrecedence	両方の基準値が提供されたならば、このパラメータはどちらが優先するかを示します。BIO_FAR_PRECEDENCEなら*MaxFARRequested*が優先されます。BIO_FRR_PRECEDENCEなら*MaxFRRRequested*が優先されます。
ReferenceBIR	照合に用いられる**BIR**。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
AdaptedBIR	適応処理された**BIR**を返すポインタ。適応**BIR**が必要なければエンプティ（0x00）をセットします。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
Result	照合が成功した場合はTRUE、失敗した場合はFALSEが設定されます。
FARAchieved	照合結果のFARが返されます。
FRRAchieved	照合結果のFRRが返されます。
Payload	*ReferenceBIR*にペイロードが含まれ、*FARAchived*がサービスの基準値を満たしているなら、バイナリでアロケートされたデータが設定されます。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
Timeout	失敗せずに正常にバイオメトリクスのキャプチャーを実行する最大ミリ秒時間。

説明　このファンクションは、指定された時間内で接続されたデバイスからバイオメトリクスデータをキャプチャーし、*ReferenceBIR*と照合します。照合が一致と判定され、*AdaptedBIR*ハンドルが提供されているなら、サービスはキャプチャー**BIR**の情報から*ReferenceBIR*を適合処理しようとします。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理されました。。
OPOS_E_FAILURE	*ReferenceBIR*が不正です。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

参照　**BIR**メソッド、**CapTenplateAdaptation**プロパティ

VerifyMatch メソッド

第1.12版で更新

形式 LONG **VerifyMatch** (**LONG** *MaxFARRequest*, **LONG** *MaxFRRRequest*, **BOOL** *FARPrecedence*, **BSTR** *SampleBIR*, **BSTR** *ReferenceBIR*, **BSTR*** *AdaptedBIR*, **BOOL*** *Result*, **LONG*** *FARAchived*, **LONG*** *FRRAchived*, **BSTR*** *Payload*);

パラメータ	説明
MaxFARRequest	照合結果を正と判断するためのFAR基準値。
MaxFRRRequest	照合結果を正と判断するためのFRR基準値。ヌルポインタならこの値を使用しません。
FARPrecedence	両方の基準値が提供されたならば、このパラメータはどちらが優先するかを示します。BIO_FAR_PRECEDENCEなら*MaxFARRequested*が優先されます。BIO_FRR_PRECEDENCEなら*MaxFRRRequested*が優先されます。
SampleBIR	処理される**BIR**。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
ReferenceBIR	照合に用いられる**BIR**。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
AdaptedBIR	適応処理された**BIR**を返すポインタ。適応**BIR**が必要なければエンプティ（0x00）をセットします。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
Result	照合が成功した場合はTRUE、失敗した場合はFALSEが設定されます。
FARAchieved	照合結果のFARが返されます。
FRRAchieved	照合結果のFRRが返されます。

Payload	*ReferenceBIR*にペイロードが含まれ、*FARAchived* がサービスの基準値を満たしているなら、バイナリ配列でアロケートされたデータが設定されます。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

説明　このファンクションは、*SampleBIR*を*ReferenceBIR*と照合します。照合が一致と判定され、*AdaptedBIR*ハンドルが提供されているなら、サービスはキャプチャー**BIR**の情報から*ReferenceBIR*を適合処理しようとします。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理されました。
OPOS_E_FAILURE	*ReferenceBIR*が不正です。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

イベント

DataEvent イベント

形式 void DataEvent (LONG *Status*);

*Status*には、次の値のいずれかに設定されます。

値	意味
BIO_DATA_ENROLL	登録が完了しました。
BIO_DATA_VERIFY	認証が完了しました。

説明 BIRプロパティとRawSensorDataプロパティには、DataEventがアプリケーションに提供される前に適切な値がセットされます。

ErrorEventイベント

形式　　void ErrorEvent(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**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**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。

ResultCode が OPOS_E_EXTENDED の場合、**RsultCodeExtended** は次のいずれかの値をとります。

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。

値	意味
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。

アプリケーションはエラーイベントハンドラ内で*pErrorResponse*に次のいずれかの値を設定できます。

値	意味

OPOS_ER_CLEAR

非同期出力、バッファリングされている出力データ（全ての非同期出力を含む)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 のときに限り設定されます。エラーを認識し、デバイスに処理の継続を指示します。デバイスはエラー状態のままですが、**DataEventEnabled** プロパティの設定にしたがって、別の**DataEvent** が引き続き通知されます。すべての入力データが通知されて、**DataEventEnabled** が再びTRUEに設定されたとき、*ErrorLocus*がOPOS_EL_INPUT に設定された新たな**ErrorEvent** 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　エラーが検知され、アプリケーションによる適切な応答がエラー状態の処理に必要なことをアプリケーションに通知します。

DataEventEnabled がTRUEにならないかぎりこのイベントは通知されないので、アプリケーションは通常のシーケンスでイベントを処理できます。エラー状態は、イベントが通知される前に、適切なプロパティに設定されます。

適切なプログラミングにおいて、**ErrorEvent**の引数*errorlocus*にEL_INPUT_DATAは発生しません。なぜなら、おのおののバイオメトリクスキャプチャー動作には、明らかに**BeginXxxxxCapture**メソッドを必要とします。それらメソッドは、少なからずひとつの**DataEvent**を生成します。アプリケーションは、**DataEventEnabled**にFalseをセットして**DataEvent**を保留にする必要がありますし、EL_INPUT_DATAが可能になる前に他のキャプチャーを要求します。

参照　**入力モデル、出力モデル**

StatusUpdateEventイベント　　*第1.12版にて修正*

形式　　void StatusUpdateEvent(LONG *Status*);

説明　　バイオメトリクスデバイスの電源状態が変化したときにアプリケーションへ通知されます。

パラメータ	説明
Status	バイオメトリクスデバイスの電源状態の変化、または、キャプチャーを完了するために、バイオメトリクスセンサーとの通信状態を要求されたユーザーに通知します。

*Status*には、次の値のいずれかに設定されます。

値	意味
BIO_SUE_RAW_DATA	生のイメージデータが入手できます。
BIO_SC_SUE_MOVE_LEFT	右過ぎます。
BIO_SC_SUE_MOVE_RIGHT	左過ぎます。
BIO_SC_SUE_MOVE_DOWN	高すぎます。
BIO_SC_SUE_MOVE_UP	低すぎます。
BIO_SC_SUE_MOVE_CLOSER	離れすぎています。
BIO_SC_SUE_MOVE_AWAY	近くにあります。
BIO_SC_SUE_MOVE_BACKWARD	前に位置します。
BIO_SC_SUE_MOVE_FORWARD	後ろに位置します。
BIO_SC_SUE_MOVE_SLOWER	ゆっくり操作してください。
BIO_SC_SUE_MOVE_FASTER	早く操作してください。
BIO_SC_SUE_SENSOR_DIRTY	センサーが汚れているので清掃が必要です。
BIO_SC_SUE_FAILED_READ	センサーからデータを採取できません。もう一度処理してください。

説明　　バイオメトリクスデバイスの電源状態か操作者の状態が変化したときに通知されます。

第　２６章
電子ジャーナル

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.10	Boolean	R/W	Open
BinaryConversion	1.10	Long	R/W	Open
CapCompareFirmwareVersion	1.10	Boolean	R	Open
CapPowerReporting	1.10	Long	R	Open
CapStatisticsReporting	1.10	Boolean	R	Open
CapUpdateFirmware	1.10	Boolean	R	Open
CapUpdateStatistics	1.10	Boolean	R	Open
CheckHealthText	1.10	String	R	Open
Claimed	1.10	Boolean	R	Open
DataCount	1.10	Long	R	Open
DataEventEnabled	1.10	Boolean	R/W	Open
DeviceEnabled	1.10	Boolean	R/W	Open & Claim
FreezeEvents	1.10	Boolean	R/W	Open
OpenResult	1.10	Long	R	無し
OutputID	1.10	Long	R	Open
PowerNotify	1.10	Long	R/W	Open
PowerState	1.10	Long	R	Open
ResultCode	1.10	Long	R	--
ResultCodeExtended	1.10	Long	R	Open
State	1.10	Long	R	--
ControlObjectDescription	1.10	String	R	--
ControlObjectVersion	1.10	Long	R	--
ServiceObjectDescription	1.10	String	R	Open
ServiceObjectVersion	1.10	Long	R	Open
DeviceDescription	1.10	String	R	Open
DeviceName	1.10	String	R	Open
専用				
AsyncMode	1.10	Boolean	R/W	Open
CapAddMarker	1.10	Boolean	R	Open

CapErasableMedium	1.10	Boolean	R	Open
CapInitializeMedium	1.10	Boolean	R	Open
CapMediumIsAvailable	1.10	Boolean	R	Open
CapPrintContent	1.10	Boolean	R	Open
CapPrintContentFile	1.10	Boolean	R	Open
CapRetrieveCurrentMarker	1.10	Boolean	R	Open
CapRetrieveMarker	1.10	Boolean	R	Open
CapRetrieveMarkerByDateTime	1.10	Boolean	R	Open
CapRetrieveMarkersDateTime	1.10	Boolean	R	Open
CapStation	1.10	Long	R	Open
CapStorageEnabled	1.10	Boolean	R	Open
CapSuspendPrintContent	1.10	Boolean	R	Open
CapSuspendQueryContent	1.10	Boolean	R	Open
CapWaterMark	1.10	Boolean	R	Open
FlagWhenIdle	1.10	Boolean	R/W	Open
MediumFreeSpace	1.10	Currency	R	Open, Claim, & Enable
MediumID	1.10	String	R	Open, Claim, & Enable
MediumIsAvailable	1.10	Boolean	R	Open, Claim, & Enable
MediumSize	1.10	Currency	R	Open, Claim, & Enable
Station	1.10	Long	R/W	Open
StorageEnabled	1.10	Boolean	R/W	Open, Claim, & Enable
Suspended	1.10	Boolean	R	Open
WaterMark	1.10	Boolean	R/W	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.10	--
Close	1.10	Open
ClaimDevice	1.10	Open
ReleaseDevice	1.10	Open & Claim
CheckHealth	1.10	Open, Claim, & Enable
ClearInput	1.10	Open, Claim, & Enable
ClearInputProperties	1.10	*適用外*
ClearOutput	1.10	Open & Claim
DirectIO	1.10	Open
CompareFirmwareVersion	1.10	Open, Claim, & Enable
ResetStatistics	1.10	Open, Claim, & Enable
RetrieveStatistics	1.10	Open, Claim, & Enable
UpdateFirmware	1.10	Open, Claim, & Enable

UpdateStatistics	1.10	Open, Claim, & Enable

専用

AddMarker	1.10	Open, Claim, & Enable
CancelPrintContent	1.10	Open, Claim, & Enable
CancelQueryContent	1.10	Open, Claim, & Enable
EraseMedium	1.10	Open, Claim, & Enable
InitializeMedium	1.10	Open, Claim, & Enable
PrintContent	1.10	Open, Claim, & Enable
PrintContentFile	1.10	Open, Claim, & Enable
QueryContent	1.10	Open, Claim, & Enable
ResumePrintContent	1.10	Open, Claim, & Enable
ResumeQueryContent	1.10	Open, Claim, & Enable
RetrieveCurrentMarker	1.10	Open, Claim, & Enable
RetrieveMarker	1.10	Open, Claim, & Enable
RetrieveMarkerByDateTime	1.10	Open, Claim, & Enable
RetrieveMarkersDateTime	1.10	Open, Claim, & Enable
SusoendPrintContent	1.10	Open, Claim, & Enable
SuspendQueryContent	1.10	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.10	Open, Claim, & Enable
DirectIOEvent	1.10	Open, Claim
ErrorEvent	1.10	Open, Claim, & Enable
OutputCompleteEvent	1.10	Open, Claim, & Enable
StatusUpdateEvent	1.10	Open, Claim, & Enable

概説

電子ジャーナルコントロールのOLEプログラマティクIDは、
“OPOS.ElectronicJournal”です。
このデバイスは、第1.10版で追加されました。

機能

電子ジャーナルは、取引の記録を電子データとしてメディアへ蓄積するデバイスです。電子ジャーナルデバイスを記録可能な状態に設定すると、POSプリンタ、またはフィスカルプリンタへ出力した印刷データが記録メディアへ書き込まれます。フィスカルプリンタの場合、印刷データは常に書き込まれます。

電子ジャーナルは次の機能を持っています。

取引データを記録します。

取引データを取得します。

また、以下の機能がオプションとして実装されます。

記録したデータをPOSプリンタまたはフィスカルプリンタで印刷します。

記録したデータを消去します。

記録用のメディアを初期化します。

また、フィスカル環境では以下の特別な機能を持っている場合があります。

全フィスカルドキュメントおよび特定のチケット番号の範囲またはチケットの日付によって指定されたチケットを再印刷します。

モデル

電子ジャーナルは、POSプリンタやフィスカルプリンタへの印刷が行われると、自動的に書き込み処理を行います。全ての出力は、FIFOで実行されることを保証しています。書き込みに失敗した場合、**ErrorEvent**が通知されます。

POSプリンタまたはフィスカルプリンタへの書き込み処理が失敗した場合も、**ErrorEvent**が通知されます。

次のメソッドは、常に同期的に実行されます。
AddMarker、RetrieveCurrentMarker、RetrieveMarker、RetrieveMarkerByDateTime、RetrieveMarkersDateTime、CheckHealth
もし、POSプリンタまたはフィスカルプリンタが非同期出力中ならば、これらのメソッドは失敗します。

次のメソッドも、常に同期的に実行されます。
SuspendPrintContent、SuspendQueryContent
これらのメソッドは、印字またはデータの取得を直ちに停止しようとします。（つまり、直後のプリンタ動作または電子ジャーナルの動作に組み込まれます。）非同期処理中であっても、このメソッドは呼び出せます。これらのメソッドの目的は、非同期処理中に例外的に印字またはデータの取得を停止することです。

次のメソッドは、同期的または非同期的に実行されます。それは、AsyncModeプロパティの値に依存します。
EraseMedium、InitializeMedium、PrintContent、PrintContentFile、QueryContent
AsyncModeがFALSEの時、これらのメソッドは同期的に実行されます。

これからデータを書き込もうとする位置へマーカを設定し、インデックスとして利用することができます。記録されたデータを取得、印刷する際には、始点と終点を示す2つのマーカを指定することでデータの範囲を指示することができます。

非同期モードで、印刷またはデータの取得を行っている途中で、処理の中断を指示することができます。

フィスカル環境では、マーカはフィスカルプリンタによって自動的に設定されます。データは、フィスカル日（Fiscal Day）に対応するセッションとして記録されます。これらのセッションは、フィスカルチケットに対応するドキュメントを含んでいます。セッションとドキュメントは、**RetrieveMarker**、**RetrieveMarkerByDateTime**、**RetrieveCurrentMarker**メソッドを使用することにより、取得することができます。取得したマーカは、**PrintContent**メソッドや**QueryContent**メソッドで利用することができます。マーカの内容やフォーマットは、実装に依存しており、アプリケーションが知る必要はありません。

電子ジャーナルは、入力デバイス（データ取得）と出力デバイス（記録・印刷）の両方の特性を持つデバイスです。

電子ジャーナルメディアに記録されるデータは、POSプリンタ、またはフィスカルプリンタデバイスに発行された印字データです。保存された情報のデータフォーマットは物理デバイスに依存しています。データを保存する物理メディアとして、不揮発性記憶装置、例えば、フラッシュカード、メモリカード、CD-RWやHDD等が利用できます。アプリケーションは、物理メディアの違いを意識する必要はありません。

記録メディアを抜き差しして交換することが可能なデバイスでは、メディアの状態変化は**StatusUpdateEvent**で通知されます。この時、記録メディアの状態をチェックして、メディアの初期化を指示することもできます。

重要なのはトランザクションデータをそのまま保存することであり、データの変換や再処理をするような機能はありません。

デバイスの共有

電子ジャーナルは以下のように排他的な利用をするデバイスです。

アプリケーションは、デバイスをイネーブルにする前に、そのデバイスの排他アクセス権を獲得しなければなりません。

アプリケーションは、大半の電子ジャーナル特有のプロパティにアクセスする前に、そのデバイスの排他アクセス権を獲得し、イネーブルにしなければなりません。

アプリケーションは、デバイスを操作するメソッドを呼び出す前に、そのデバイスの排他アクセス権を獲得し、イネーブルにしなければなりません。

使用上の必要条件の詳細は、「一覧」の表を参照してください。

プロパティ

AsyncMode プロパティ　R/W

形式　BOOL AsyncMode;

説明　TRUE：　EraseMedium、InitializeMedium、PrintContent、PrintContentFile、QueryContentメソッドは、非同期に実行されます。
FALSE：これらのメソッドは同期的に印刷されます。

このプロパティは、Openメソッドにより FALSEに初期化されます。

戻り値　本プロパティ設定時、次の値がResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
その他	ResultCodeの項目を参照してください。

CapAddMarker プロパティ

形式　BOOL CapAddMarker;

説明　TRUE：　AddMarkerメソッドを利用することができます。
FALSE：　AddMarkerメソッドを利用することができません。
フィスカル用の電子ジャーナルデバイスの場合は、通常FALSEに設定されています。

このプロパティは、Openメソッドにより初期化されます。

参照　AddMarkerメソッド

CapErasableMedium プロパティ

形式　BOOL CapErasableMedium;

説明　TRUE：　記録メディアの内容を消去することができます。
FALSE：記録メディアの内容を消去することができません。

このプロパティは、Openメソッドにより初期化されます。

CapInitializeMedium プロパティ

形式　　BOOL CapInitializeMedium;

説明　　TRUE：　記録メディアを初期化することができます。
FALSE：　記録メディアを初期化することができません。

このプロパティは、Openメソッドにより初期化されます。

CapMediumIsAvailable プロパティ　　*第1.11版にて更新*

形式　　BOOL CapMediumIsAvailable;

説明　　TRUE：　記録メディアを利用可能か判別することができます。
FALSE：　記録メディアを利用可能か判別することができません。

このプロパティは、Openメソッドにより初期化されます。

参照　MediumIsAvailableプロパティ

CapPrintContent プロパティ　　*第1.11版にて更新*

形式　　BOOL CapPrintContent;

説明　　TRUE：　記録されているジャーナルドキュメントを、接続されている印刷用デバイスから、直接印刷することができます。
FALSE：　記録されているジャーナルドキュメントを、接続されている印刷用デバイスから、直接印刷することができません。

このプロパティは、Openメソッドにより初期化されます。

参照　PrintContentメソッド

CapPrintContentFile プロパティ　　*第1.11版にて更新*

形式　　BOOL CapPrintContentFile;

説明　　TRUE：　記録メディアから取得したジャーナルドキュメント（ファイル）を、接続されている印刷用デバイスから、印刷することができます。
FALSE：記録メディアから取得したジャーナルドキュメント（ファイル）を、接続されている印刷用デバイスから、印刷することができません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　PrintContentFileメソット

CapRetrieveCurrentMarker プロパティ

形式　　BOOL CapRetrieveCurrentMarker;

説明　　TRUE：　**RetrieveCurrentMarker**メソッドを利用することができます。
FALSE：**RetrieveCurrentMarker**メソッドを利用することができません。
フィスカル用の電子ジャーナルデバイスの場合は、通常TRUEに設定されています。

このプロパティは、**Open**メソッドにより初期化されます。

参照　　**RetrieveCurrentMaker**メソッド

CapRetrieveMarker プロパティ

形式　　BOOL CapRetrieveMarker;

説明　　TRUE：　**RetrieveMarker**メソッドを利用することができます。
FALSE：**RetrieveMarker**メソッドを利用することができません。
フィスカル用の電子ジャーナルデバイスの場合は、通常TRUEに設定されています。

このプロパティは、**Open**メソッドにより初期化されます。

参照　　**RetrieveMaker**メソッド

CapRetrieveMarkerByDateTime プロパティ

形式　　BOOL CapRetrieveMarkerByDateTime;

説明　　TRUE：　RetrieveMarkerByDateTimeメソッドを利用することができます。
FALSE：RetrieveMarkerByDateTimeメソッドを利用することができません。
フィスカル用の電子ジャーナルデバイスの場合は、通常TRUEに設定されています。

このプロパティは、Openメソッドにより初期化されます。

参照　　RetrieveMakerByDateTimeメソッド

CapRetrieveMarkersDateTime プロパティ

形式　　BOOL CapRetrieveMarkersDateTime;

説明　　TRUE：　RetrieveMarkersDateTimeメソッドを利用することができます。
FALSE：RetrieveMarkersDateTimeメソッドを利用することができません。
フィスカル用の電子ジャーナルデバイスの場合は、通常TRUEに設定されています。

このプロパティは、Openメソッドにより初期化されます。

参照　　RetrieveMakersDateTimeメソッド

CapStation プロパティ

形式　LONG CapStation;

説明　データの記録対象とすることが可能な印刷用デバイスのステーションを示します。

このプロパティには、次に示すいずれかの値の論理和が設定されます。

値	意味
EJ_S_RECEIPT	レシートステーションへ出力されたデータをメディアに記録することができます。
EJ_S_SLIP	スリップステーションへ出力されたデータをメディアに記録することができます。
EJ_S_JOURNAL	ジャーナルステーションへ出力されたデータをメディアに記録することができます。

このプロパティは、**Open**メソッドにより初期化されます。

CapStorageEnabled プロパティ

形式　BOOL CapStorageEnabled;

説明　TRUE：　**StorageEnabled**プロパティによって、印刷データの記録を行うかどうかを制御することができます。
FALSE：**StorageEnabled**プロパティによって、印刷データの記録を行うかどうかを制御することができません。この場合、**StorageEnabled**プロパティは常にTRUEに設定されます。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**StorageEnabled**プロパティ

CapSuspendPrintContent プロパティ

形式　　BOOL CapSuspendPrintContent;

説明　　TRUE：　印刷処理を中断することができます。
FALSE：　印刷処理を中断することができません。

このプロパティは、Openメソッドにより初期化されます。

参照　　Suspendedプロパティ

CapSuspendQueryContent プロパティ

形式　　BOOL CapSuspendQueryContent;

説明　　TRUE：　データの取得処理を中断することができます。
FALSE：　データの取得処理を中断することができません。

このプロパティは、Openメソッドにより初期化されます。

参照　　Suspendedプロパティ

CapWaterMark プロパティ

形式　　BOOL CapWaterMark;

説明　　TRUE：　ジャーナルドキュメントの再印刷時に、背景を印刷することができます。
FALSE：　ジャーナルドキュメントの再印刷時に、背景を印刷することができません。

FlagWhenIdle プロパティ　R/W　*第1.11版にて更新*

形式　BOOL FlagWhenIdle;

説明　TRUE： 電子ジャーナルコントロールがアイドル状態であれば、**StatusUpdateEvent**を通知します。

このステータスイベントが通知されると、**FlagWhenIdle**は自動的にFALSEにリセットされます。

このプロパティを使用してステータスイベントを利用することにより、アプリケーションは全ての非同期出力が終了した時を知ることができます。出力が正常終了した時、または**ClearOutput**メソッドもしくは**ErrorEvent**を受け取ったイベントハンドラにより出力が削除された時に、イベントが通知されます。

もし、**FlagWhenIdle**プロパティをTRUEに設定する時に、**State**プロパティが既にOPOS_S_IDLEであれば、**StatusUpdateEvent**は直ちに通知されます。従って、非同期出力の終了とこのフラグの設定のすれ違いを気にせずに、アプリケーションはこのイベントを使用できます。

このプロパティは、**Open**メソッドによりFALSEに初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
その他	**ResultCode**の項目を参照してください。

参照　**State**プロパティ、**ClearOutput**メソッド

MediumFreeSpace プロパティ

形式　CURRENCY MediumFreeSpace;

説明　記録メディアの残容量をバイト単位で保持します。POSプリンタまたはフィスカルプリンタへの印刷メソッドが実行されてデータが記録される都度、値が減少します。残容量が少なくなる、もしくは空になると、**StatusUpdateEvent**が通知されます。

このプロパティは、**Open**メソッド後最初にデバイスをイネーブルにした時に初期化されます。

MediumID プロパティ

形式　BSTR MediumID;

説明　現在装着されている記録メディアの識別情報を保持します。物理メディアから取得した値を保持するため、デバイスをイネーブルにした時に初期化されます。
物理メディアから情報を取得できない場合は、このプロパティは空文字列に初期化されます。

MediumIsAvailable プロパティ　*第1.11版にて更新*

形式　BOOL MediumIsAvailable;

説明　記録メディアの装着状態を保持します。CapMediumIsAvailableプロパティがTRUEの場合のみ有効です。
TRUE： 記録メディアが装着されています。
FALSE： 記録メディアが装着されていません。

記録メディアを差し替えることができない場合、このプロパティは常にTRUEに設定されます。

このプロパティは、Openメソッド後最初にデバイスをイネーブルにした時に初期化されます。

参照　CapMediumIsAvailableプロパティ

MediumSize プロパティ

形式　CURRENCY MediumSize;

説明　記録メディアの容量をバイト単位で保持します。

このプロパティは、Openメソッド後最初にデバイスをイネーブルにした時に初期化されます。

Station プロパティ　R/W

形式　LONG Station;

説明　データを記録する対象となる印刷用デバイスのステーションを、下記の値の論理和で示します。

値	意味
EJ_S_RECEIPT	POSプリンタ、またはフィスカルプリンタのレシートステーションに出力したデータを、メディアに記録します。
EJ_S_SLIP	POSプリンタ、またはフィスカルプリンタのスリップステーションに出力したデータを、メディアに記録します。
EJ_S_JOURNAL	POSプリンタ、またはフィスカルプリンタのジャーナルステーションに出力したデータを、メディアに記録します。

このプロパティは、**Open**メソッドにより EJ_S_RECEIPTに初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
その他	**ResultCode**の項目を参照してください。

StorageEnabled プロパティ　R/W　　*第1.11版にて更新*

形式　　BOOL StorageEnabled;

説明　　TRUE： デバイスは記録可能な状態です。POSプリンタ、またはフィスカルプリンタへの出力データは、電子データとして記録メディア上に逐次的に記録していきます。どのステーションへ出力したデータを記録するかを、予めStationプロパティで選択しておく必要があります。
FALSE： デバイスは記録可能な状態ではありません。

このプロパティは、Openメソッドにより初期化されます。

戻り値　　本プロパティ設定時、次の値がResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
その他	ResultCodeの項目を参照してください。

参照　　Stationプロパティ

Suspended プロパティ

形式　　BOOL Suspended;

説明　　TRUE： 印刷またはデータ取得処理を中断中です。
FALSE： 印刷またはデータ取得処理の中断中ではありません。

CapSuspendPrintContentプロパティとCapSuspendQueryContentプロパティが両方ともFALSEの場合、処理を中断する機能がなく、このプロパティは常にFALSEが設定されます。

このプロパティは、Openメソッドにより初期化されます。

参照　　CapSuspendPrintContentプロパティ、CapSuspendQueryContentプロパティ

WaterMark プロパティ　R/W

形式　　BOOL WaterMark;

説明　　このプロパティは、ジャーナルドキュメントを印刷する際に、予め定義された特定の背景を印刷するかどうかを指定します。
TRUE：背景を印刷することで、記録データの再印刷であることを明示します。
FALSE： 背景を印刷しません。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値　　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
その他	**ResultCode**の項目を参照してください。

メソッド

AddMarker メソッド

形式　LONG AddMarker (BSTR *Marker*);

パラメータ	説明
Marker	マーカの識別情報です。

説明　データをファイルとして取得する場合や、接続されているPOSプリンタ、またはフィスカルプリンタに印刷する場合に、データの範囲を指定するためのインデックスとしたい識別情報を指定します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	*Marker*パラメータとして使用できない文字が含まれているか、*Marker*パラメータとして使用できる文字数を超えています。
OPOS_E_BUSY	出力中なので実行できません（POSプリンタまたはフィスカルプリンタが印刷中の場合も含まれます）。
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_EEJ_EXISTING: 現在のメディアで既に登録されているマーカと重複しています。 **ResultCodeExtended** = OPOS_EEJ_MEDIUM_FULL: 容量不足により、マーカを書き込むことができません。
その他	**ResultCode**の項目を参照してください。

CancelPrintContent メソッド

形式　LONG CancelPrintContent ();

説明　一時中断している印刷処理をキャンセルします。このメソッドが成功した場合、残りのデータは印刷されません。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
その他	**ResultCode**の項目を参照してください。

CancelQueryContent メソッド

形式　LONG CancelQueryContent ();

説明　一時中断しているデータ取得処理をキャンセルします。このメソッドを実行した場合、記録データを格納するためのファイルは作成されません。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
その他	**ResultCode**の項目を参照してください。

EraseMedium メソッド

形式　LONG EraseMedium ();

説明　記録メディア上に記録されている全てのデータ（マーカも含みます）を削除します。

このメソッドは、**AsyncMode**がFALSEならば、同期的に実行され、**AsyncMode**がTRUEならば、非同期的に実行されます。

非同期で実行した場合、メソッドの実行結果はイベントで通知されます。メソッドが成功した場合は**OutputCompleteEvent**が、失敗した場合は**ErrorEvent**が通知されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_FAILURE	データの削除に失敗しました。
その他	**ResultCode**の項目を参照してください。

参照　**AsyncMode**プロパティ

InitializeMedium メソッド

形式　LONG InitializeMedium (BSTR *MediumID*);

パラメータ	説明
MediumID	記録メディアの識別情報です。

説明　記録メディアを初期化します。このとき、メディアに識別情報を示す文字列を指定することができます。

記録メディアに識別情報を指定することができない場合、**MediumID**プロパティには空文字列が格納されます。

このメソッドは、**AsyncMode**がFALSEならば、同期的に実行され、**AsyncMode**がTRUEならば、非同期的に実行されます。

非同期で実行した場合、メソッドの実行結果はイベントで通知されます。メソッドが成功した場合は**OutputCompleteEvent**が、失敗した場合は**ErrorEvent**が通知されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_BUSY	出力中なので実行できません（POSプリンタまたはフィスカルプリンタが印刷中の場合も含まれます）。
その他	**ResultCode**の項目を参照してください。

参照　**AsyncMode**プロパティ、**MediumID**プロパティ

PrintContent メソッド　　*第1.11版にて更新*

形式　LONG PrintContent (BSTR *FromMarker*, BSTR *ToMarker*);

パラメータ	説明
FromMarker	データの開始位置を示すマーカの識別情報。記録メディアの先頭を指定したい場合は、空文字列を指定します。
ToMarker	データの終了位置を示すマーカの識別情報。記録メディアの末尾を指定したい場合は、空文字列を指定します。

説明　接続されているプリンタに対して、メディアに記録されているジャーナルドキュメントを印刷する時にこのメソッドを呼び出します。

このメソッドは**CapPrintContent**プロパティがTRUEの場合のみサポートされています。

*FromMarker*パラメータに空文字列を指定した場合は、記録メディアの先頭を指定したことになります。*ToMarker*パラメータに空文字列を指定した場合は、記録メディアの末尾を指定したことになります。

このメソッドは、**AsyncMode**がFALSEならば、同期的に実行され、**AsyncMode**がTRUEならば、非同期的に実行されます。

非同期で実行した場合、メソッドの実行結果はイベントで通知されます。メソッドが成功した場合は**OutputCompleteEvent**が、失敗した場合は**ErrorEvent**が通知されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
その他	**ResultCode**の項目を参照してください。

参照　**AsyncMode**プロパティ、**CapPrintContent**プロパティ

PrintContentFile メソッド　　*第1.11版にて更新*

形式　LONG PrintContentFile (BSTR *FileName*);

パラメータ	説明
FileName	印刷データが格納されているファイル名を指定します。

説明　記録メディアから取得したデータを保存したファイルから、接続されている印刷デバイスに対して印刷する時にこのメソッドを呼び出します。ファイルに保存されている全データが印刷されます。このメソッドは**CapPrintContentFile**プロパティがTRUEの場合のみ、サポートされています。

このメソッドは、**AsyncMode**がFALSEならば、同期的に実行され、**AsyncMode**がTRUEならば、非同期的に実行されます。

非同期で実行した場合、メソッドの実行結果はイベントで通知されます。メソッドが成功した場合は**OutputCompleteEvent**が、失敗した場合は**ErrorEvent**が通知されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	ファイル名に不正な文字が含まれています。
OPOS_E_NOEXIST	*FileName*パラメータで指定されたファイルが見つかりません。
その他	**ResultCode**の項目を参照してください。

参照　**AsyncMode**プロパティ、**CapPrintContentFile**プロパティ

QueryContent メソッド　　*第1.11版にて更新*

形式　LONG QueryContent (BSTR *FileName*, BSTR *FromMarker*, BSTR *ToMarker*);

パラメータ	説明
FileName	取得したデータを保存するファイル名を指定します。
FromMarker	データの開始位置を示すマーカの識別情報。記録メディアの先頭を指定したい場合は、空文字列を指定します。
ToMarker	データの終了位置を示すマーカの識別情報。記録メディアの末尾を指定したい場合は、空文字列を指定します。

説明　電子ジャーナルに保存されたデータを取得し、*FileName*で指定したファイルに保存する時にこのメソッドを呼び出します。

AsyncModeがTRUEの場合、データ取得は非同期的に実行されます。メソッドは直ちに処理をアプリケーションに返します。データ取得が成功した場合は**DataEvent**が、失敗した場合は**ErrorEvent**が通知されます。

*FromMarker*パラメータに空文字列を指定した場合は、記録メディアの先頭を指定したことになります。*ToMarker*パラメータに空文字列を指定した場合は、記録メディアの末尾を指定したことになります。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_BUSY	出力中なので実行できません（POSプリンタまたはフィスカルプリンタが印刷中の場合も含まれます）。
OPOS_E_EXISTS	*FileName*パラメータで指定されたファイルが既に存在します。
OPOS_E_ILLEGAL	ファイル名に不正な文字が含まれています。
その他	**ResultCode**の項目を参照してください。

参照　**AsyncMode**プロパティ

ResumePrintContent メソッド

形式　LONG ResumePrintContent ();

説明　一時中断していたデータ印刷処理を再開します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
その他	**ResultCode**の項目を参照してください。

ResumeQueryContent メソッド

形式　LONG ResumeQueryContent ();

説明　一時中断していたデータ取得処理を再開します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
その他	**ResultCode**の項目を参照してください。

RetrieveCurrentMarker メソッド

形式　LONG RetrieveCurrentMarker (LONG *MarkerType*, BSTR* *pMarker*);

パラメータ	説明
MarkerType	取得するマーカの種類を指定します。下記の値を参照。
pMarker	戻り値としてマーカを格納するための文字ストリングのポインタ。マーカのフォーマットは、実装に依存しています。

*MarkerType*パラメータの値は、

値	意味
EJ_MT_SESSION_BEG	直前に生成されたセッションの開始位置を示すマーカ。
EJ_MT_SESSION_END	直前に生成されたセッションの終了位置を示すマーカ。
EJ_MT_DOCUMENT	直前に生成されたドキュメントまたはチケットの位置を示すマーカ。
EJ_MT_HEAD	電子ジャーナルメディアの先頭を示すマーカ。
EJ_MT_TAIL	電子ジャーナルメディアの末尾を示すマーカ。

説明　直前に保存されたマーカを取得する時にこのメソッドを呼び出します。取得するマーカの種類は、*MarkerType*パラメータで指定します。取得したマーカは、*pMarker*パラメータで指定した文字ストリングに格納されます。マーカを示す文字列の内容やフォーマットは実装に依存しており、アプリケーションが知る必要はありません。戻り値として得られたマーカは、**PrintContent**メソッドや**QueryContent**メソッドのパラメータとして利用できます。

EJ_MT_HEADとEJ_MT_TAILは、電子ジャーナルメディア全体の中で、有効なデータの範囲の値を意味しています。

このメソッドは、**CapRetrieveCurrentMaker**プロパティがTRUEの場合のみサポートされます。

このメソッドは、通常フィスカル用の電子ジャーナルデバイスで使用されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	*MarkerType*パラメータに不正な値が指定されています。
OPOS_E_NOEXIST	指定された種類のマーカが存在しませんでした。
その他	**ResultCode**の項目を参照してください。

参照　**CapRetrieveCurrentMarker**プロパティ、**PrintContent**メソッド、**QueryContent**メソッド

RetrieveMarker メソッド

形式　LONG RetrieveMarker (LONG *MarkerType*, LONG *SessionNumber*, LONG *DocumentNumber*, BSTR* *pMarker*);

パラメータ	説明
MarkerType	取得するマーカの種類を指定します。下記の値を参照。
SessionNumber	取得するマーカが含まれているセッションの番号。セッションの概念をサポートしていないデバイスの場合、負の値を指定する必要があります。
DocumentNumber	取得するマーカが含まれているドキュメントの番号。*MarkerType*パラメータがEJ_MT_SESSION_BEGまたはEJ_MT_SESSION_ENDの場合、このパラメータは無視されます。
pMarker	戻り値としてマーカを格納するための文字ストリングのポインタ。マーカのフォーマットは、実装に依存しています。

*MarkerType*パラメータの値は、

値	意味
EJ_MT_SESSION_BEG	セッションの開始位置を示すマーカ。
EJ_MT_SESSION_END	セッションの終了位置を示すマーカ。
EJ_MT_DOCUMENT	ドキュメントまたはチケットを示すマーカ。

説明　マーカを取得する時にこのメソッドを呼び出します。取得するマーカは、*MarkerType*、*SessionNumber*、*DocumentNumber*パラメータで指定します。取得したマーカは、*pMarker*パラメータで指定した文字ストリングに格納されます。マーカを示す文字列の内容やフォーマットは実装に依存しており、アプリケーションが知る必要はありません。戻り値として得られたマーカは、**PrintContent**メソッドや**QueryContent**メソッドのパラメータとして利用できます。

フィスカル用の電子ジャーナルデバイスの場合、*SessionNumber*はフィスカルプリンタから返されたフィスカル日（Fiscal Day）のカウンタ番号に対応しています（**GetData**メソッドのパラメータ用の値であるFPTR_GD_Z_REPORTを参照してください）。同様に、*DocumentNumber*はフィスカルチケット番号に対応しています。

このメソッドは、**CapRetrieveMaker**プロパティがTRUEの場合のみサポートされます。

このメソッドは、通常フィスカル用の電子ジャーナルデバイスで使用されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	パラメータに不正な値が指定されたか、*MarkerType*パラメータに不正な値が指定されています。
OPOS_E_NOEXIST	指定されたマーカが存在しませんでした。
その他	**ResultCode**の項目を参照してください。

参照　**CapRetrieveMarker**プロパティ、**PrintContent**メソッド、**QueryContent**メソッド、フィスカルプリンタの**GetData**メソッド

RetrieveMarkerByDateTime メソッド

形式　LONG RetrieveMarkerByDateTime (LONG *MarkerType*, BSTR *DateTime*, BSTR *MarkerNumber*, BSTR* *pMarker*);

パラメータ	説明
MarkerType	取得するマーカの種類を指定します。下記の値を参照。
DateTime	取得するマーカの日付・時間。'YYYYMMDDhhmmss'の形式で指定します。アプリケーションが、時、分、秒の一部または全部を指定することができなければ、省略することが可能です。
MarkerNumber	要求した形式のマーカが、*DateTime*パラメータで指定した日付・時間に複数存在する場合、取得したいマーカの番号を指定します。番号は1から始まり、1つのマーカに対して１ずつインクリメントされます。
pMarker	戻り値としてマーカを格納するための文字ストリングのポインタ。マーカのフォーマットは、実装に依存しています。

*MarkerType*パラメータの値は、

値	意味
EJ_MT_SESSION_BEG	セッションの開始位置を示すマーカ。
EJ_MT_SESSION_END	セッションの終了位置を示すマーカ。
EJ_MT_DOCUMENT	ドキュメントまたはチケットを示すマーカ。

説明　マーカを取得する時にこのメソッドを呼び出します。取得するマーカは、*MarkerType*、*DateTime*、*MarkerNumber*パラメータで指定します。取得したマーカは、*pMarker*パラメータで指定した文字ストリングに格納されます。マーカを示す文字列の内容やフォーマットは実装に依存しており、アプリケーションが知る必要はありません。戻り値として得られたマーカは、**PrintContent**メソッドや**QueryContent**メソッドのパラメータとして利用できます。

このメソッドは、**CapRetrieveMakerByDateTime**プロパティがTRUEの場合のみサポートされます。

このメソッドは、通常フィスカル用の電子ジャーナルデバイスで使用されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	パラメータのどれかに不正な値が指定されました。*MarkerType*パラメータに不正な値が指定された、*DateTime*パラメータが不正、または*MarkerNumber*で指定されたマーカは、検索範囲には存在していません。
OPOS_E_NOEXIST	指定された日付・時間にマーカが存在しませんでした。
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_EEJ_MULTIPLE_MARKER: 指定された日付・時間に1つ以上のマーカが存在します。
その他	**ResultCode**の項目を参照してください。

参照　**CapRetrieveMarkerByDateTime**プロパティ、**PrintContent**メソッド、**QueryContent**メソッド

RetrieveMarkersDateTime メソッド

形式　LONG RetrieveMarkersDateTime (BSTR *Marker*, BSTR* *pDateTime*);

パラメータ	説明
Marker	日付・時間を取得したいマーカを指定します。
pDateTime	戻り値として日付・時間文字列を格納するための文字ストリングのポインタ。

説明　指定したマーカの日付・時間を取得する時にこのメソッドを呼び出します。マーカには、**AddMarker**メソッドで設定したものと、**RetrieveMarker**または**RetrieveCurrentMarker**メソッドで取得したものの、どちらも指定できます。得られた日付・時間は、*pDateTime*パラメータで指定した文字ストリングに、'YYYYMMDDhhmmss"の形式で格納されます。時、分、秒の一部または全部が不明な場合、不明箇所は'?'マークで埋められます。

このメソッドは、**CapRetrieveMakersDateTime**プロパティがTRUEの場合のみサポートされます。

このメソッドは、通常フィスカル用の電子ジャーナルデバイスで使用されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	*Marker*パラメータが不正です。
その他	**ResultCode**の項目を参照してください。

参照　**CapRetrieveMarkersDateTime**プロパティ、**AddMarker**メソッド、**RetrieveCurrentMarker**メソッド、**RetrieveMarker**メソッド

SuspendPrintContent メソッド

形式　LONG SuspendPrintContent ();

説明　印刷処理を一時中断し、中断状態へ遷移させる時にこのメソッドを呼び出します。非同期で印刷を行っている場合のみ有効です。このメソッドは、非同期出力中に例外状態に陥ってしまった場合に、エラーイベントハンドラの中等で使用することを意図したものです。

メソッド実行後、**Suspended**プロパティがTRUEに変化し、**StatusUpdateEvent**が通知されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	印刷中ではありません。
その他	**ResultCode**の項目を参照してください。

参照　**Suspended**プロパティ

SuspendQueryContent メソッド

形式　LONG SuspendQueryContent ();

説明　記録メディアからのデータ取得処理を一時中断し、中断状態へ遷移させる時にこのメソッドを呼び出します。このメソッドは、非同期処理中に例外状態に陥ってしまった場合に、エラーイベントハンドラの中等で使用することを意図したものです。

メソッド実行後、**Suspended**プロパティがTRUEに変化し、**StatusUpdateEvent**が通知されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	データ取得中ではありません。
その他	**ResultCode**の項目を参照してください。

参照　**Suspended**プロパティ

イベント

DataEvent イベント

形式　　void DataEvent (LONG *Status*);

パラメータ	説明
Status	0が設定されます。

説明　　電子ジャーナルからのデータ取得とファイルへの保存が完了したことを、アプリケーションに通知します。

このイベントは、**DataEventEnabled**プロパティがTRUEに設定され、非同期的に実行された**QueryContent**メソッドの処理が終了した後に通知されます。

参照　　「**入力モデルについて**」、**DataEventEnabled**プロパティ、**FreezeEvents**プロパティ

ErrorEvent イベント

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

パラメータ	説明
ResultCode	エラーイベントを起こした原因のコード。値は**ResultCode**参照。
ResultCodeExtended	エラーイベントを起こした原因の拡張コード。*ResultCode*がOPOS_E_EXTENDEDの場合については、下記の値を参照。それ以外の場合、デバイスサービス固有の値が格納されている場合があります。
ErrorLocus	エラーの位置です。下記の値を参照。
pErrorResponse	エラーイベント応答へのポインタ。下記の値を参照。

*ErrorLocus*パラメータは、次に示す値の内の1つに設定されます。

値	意味
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、入力データを処理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。
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。

*ResultCode*がOPOS_E_EXTENDEDの場合、*ResultCodeExtended*は次に示す値のいずれかに設定されます。

値	意味
OPOS_EEJ_UNINITIALIZED_MEDIUM	メディアが初期化されていません。
OPOS_EEJ_CORRUPTED_MEDIUM	メディアもしくはメディア上のデータが破損しています。

OPOS_EEJ_UNKNOWN_DATAFORMAT
サポートしていない未知のデータフォーマットです。

OPOS_EEJ_NOT_ENOUGH_SPACE
メディアに、データを保存するだけの十分な残容量がありません。

OPOS_EEJ_MULTIPLE_MARKERS
1つ以上のマーカを要求したが、1つのマーカしか返すことができない。

*pErrorResponse*によって指定された位置の内容は、*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、次に示す値の内の1つに変更することが可能です。

値	意味
OPOS_ER_CLEAR	その非同期出力を含むすべてのバッファリングされたデータ、またはバッファリングされている入力データを削除します（**ClearOutput**メソッドまたは**ClearInput**メソッドと同様の効果があります）。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。
OPOS_ER_RETRY	通常は、*ErrorLocus*がOPOS_EL_OUTPUTの場合に使用します。 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の場合でも指定できる場合があります。 *ErrorLocus*がOPOS_EL_OUTPUTの場合のデフォルトです。

説明　エラーを察知し、電子ジャーナルコントロールの状態がエラー状態に移った時に通知されます。アプリケーションは、適切な応答を行うことが必要です。

ErrorEventは、以下の場合に通知されます。

POSプリンタまたはフィスカルプリンタで非同期印刷とともに行われている、電子ジャーナルメディアへの書き込みが失敗したとき。

非同期モードで実行したQueryContentメソッドが失敗したとき。

非同期モードで実行したInitializeMedium、EraseMedium、PrintContent、PrintContentFileメソッドのいずれかが失敗したとき。

入力のエラーイベントは、電子ジャーナルデバイスからデータを取得している最中にエラーが発生した場合に通知されます。正しいアプリケーションシーケンシングが行われるよう、**DataEventEnabled**プロパティがTRUEになるまで、入力エラーイベントは通知されません。全てのエラー情報は、イベントが発生する前に適切なプロパティに設定されます。

出力のエラーイベントは、非同期出力中にエラーが発生した場合に通知されます。全てのエラー情報は、イベントが発生する前に適切なプロパティに設定されます。

参照　**「ステータス、結果コード、状態モデル」**

OutputCompleteEvent イベント

形式　　void OutputCompleteEvent (LONG *OutputID*);

パラメータ	説明
OutputID	完了した非同期出力リクエストのID番号。

説明　　前に開始された、*OutputID*で示される非同期出力リクエストが正常に終了したときに通知します。

OutputCompleteEventは、以下の場合に通知されます。

非同期モードで実行したInitializeMedium、EraseMedium、PrintContent、PrintContentFileメソッドのいずれかが成功したとき。

要求したデータが送信され、デバイスによって正しく処理されたことが確認された後で、このイベントが通知されます。

参照　　「出力モデルについて」

StatusUpdateEvent イベント　　*第1.12版にて更新*

形式　　void StatusUpdateEvent (LONG *Status*);

パラメータ	説明
Status	電子ジャーナルデバイスの状態変化を示します。以下の値を参照。

*Status*は、次に示す値のいずれかに設定されます。

値	意味
EJ_SUE_MEDIUM_NEAR_FULL	メディアの空き容量が残りわずかです。
EJ_SUE_MEDIUM_FULL	記録メディアが一杯になりました。
EJ_SUE_MEDIUM_REMOVED	メディアがデバイスから取り外されました。
EJ_SUE_MEDIUM_INSERTED	メディアがデバイスに装着されました。
EJ_SUE_ SUSPENDED	データの印刷または取得処理が一時停止状態になりました。
EJ_SUE_IDLE	全ての非同期出力が、成功するか、または削除されることによって終了しました。電子ジャーナルの**State**は現在OPOS_S_IDLEです。**FlagWhenIdle**プロパティは、このイベントが通知するためにTRUEでなければなりません。そして、電子ジャーナルコントロールは、イベントが通知する前に、そのプロパティをFALSEへ自動的にリセットします。

説明　　電子ジャーナルデバイスの状態が変化した時に通知されます。

参照　　「出力モデルについて」

第　２７章

モーションセンサ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.3	Boolean	R/W	*適用外*
BinaryConversion	1.3	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.3	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.3	String	R	Open
Claimed	1.3	Boolean	R	Open
DataCount	1.3	Long	R	Open
DataEventEnabled	1.3	Boolean	R/W	Open
DeviceEnabled	1.3	Boolean	R/W	Open,Claim
FreezeEvents	1.3	Boolean	R/W	Open
OpenResult	1.5	Long	R	無し
OutputID	1.3	Long	R	Open
PowerNotify	1.3	Long	R/W	Open
PowerState	1.3	Long	R	Open
ResultCode	1.3	Long	R	無し
ResultCodeExtended	1.3	Long	R	Open
State	1.3	Long	R	無し
ControlObjectDescription	1.3	String	R	無し
ControlObjectVersion	1.3	Long	R	無し
ServiceObjectDescription	1.3	String	R	Open
ServiceObjectVersion	1.3	Long	R	Open
DeviceDescription	1.3	String	R	Open
DeviceName	1.3	String	R	Open

プロパティ (続き)

専用	*版数*	*型*	*アクセス*	*使用可能条件*
CapTransaction	1.3	Boolean	R	Open
AsyncMode	1.3	Boolean	R/W	Open, Claim, & Enable
EventType	1.3	Long	R/W	Open
SystemClocks	1.3	Long	R	Open. Claim, & Enable
SystemVideoSaveBuffers	1.3	Long	R	Open. Claim, & Enable
Timeout	1.3	Long	R/W	Open
UnitsOnline	1.3	Long	R	Open, Claim, & Enable
CurrentUnitID	1.3	Long	R/W	Open, Claim, & Enable
CapSelectCharacterSet	1.3	Boolean	R	Open, Claim, & Enable
CapTone	1.3	Boolean	R	Open, Claim, & Enable
CapTouch	1.3	Boolean	R	Open, Claim, & Enable
AutoToneDuration	1.3	Long	R/W	Open, Claim, & Enable
AutoToneFrequency	1.3	Long	R/W	Open, Claim, & Enable
CharacterSet	1.3	Long	R	Open, Claim, & Enable
CharacterSetList	1.3	String	R	Open, Claim, & Enable
Clocks	1.3	Long	R	Open, Claim, & Enable
VideoDataCount	1.3	Long	R	Open, Claim, & Enable
VideoMode	1.3	Long	R/W	Open, Claim, & Enable
VideoModeList	1.3	String	R	Open, Claim, & Enable
VideoSaveBuffers	1.3	Long	R	Open, Claim, & Enable
ErrorUnits	1.3	Long	R	Open
ErrorString	1.3	String	R	Open
EventUnitID	1.3	Long	R	Open, Claim
EventUnits	1.3	Long	R	Open, Claim
EventString	1.3	String	R	Open, Claim

メソッド

共通	*版数*	*使用可能条件*
Open	1.3	無し
Close	1.3	Open
ClaimDevice	1.3	Open
ReleaseDevice	1.3	Open, Claim
CheckHealth	1.3	Open, Claim, & Enable
ClearInput	1.3	Open, Claim
ClearInputProperties	1.10	*適用外*
ClearOutput	1.3	Open, Claim
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.3	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用		
WaitForMotion	1.7	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.3	Open, Claim, & Enable
DirectIOEvent	1.3	Open, Claim
ErrorEvent	1.3	Open, Claim, & Enable
OutputCompleteEvent	1.3	Open, Claim, & Enable
StatusUpdateEvent	1.3	Open, Claim, & Enable

概説

モーションセンサーのOLEプログラマティクIDは、
“OPOS.MotionSensor”です。

このデバイスは、第1.7版にて追加されました。

詳細はUPOS仕様書Version1.14を参照ください。

モーションセンサは、以下の機能を最低限有します。

POSで人を発見する機能を提供する機能をサポートします。

ハードウェアかソフトウェアの検知機能で動作の検知した変化を通知する機能をサポートします。

第 28 章
紙幣入金機

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.11	Boolean	R/W	*適用外*
BinaryConversion	1.11	Long	R/W	Open
CapCompareFirmwareVersion	1.11	Boolean	R	Open
CapPowerReporting	1.11	Long	R	Open
CapStatisticsReporting	1.11	Boolean	R	Open
CapUpdateFirmware	1.11	Boolean	R	Open
CapUpdateStatistics	1.11	Boolean	R	Open
CheckHealthText	1.11	String	R	Open
Claimed	1.11	Boolean	R	Open
DataCount	1.11	Long	R	Open
DataEventEnabled	1.11	Boolean	R/W	Open
DeviceEnabled	1.11	Boolean	R/W	Open & Claim
FreezeEvents	1.11	Boolean	R/W	Open
OpenResult	1.11	Long	R	無し
OutputID	1.11	Long	R	*適用外*
PowerNotify	1.11	Long	R/W	Open
PowerState	1.11	Long	R	Open
ResultCode	1.11	Long	R	無し
ResultCodeExtended	1.11	Long	R	Open
State	1.11	Long	R	無し
ControlObjectDescription	1.11	String	R	無し
ControlObjectVersion	1.11	Long	R	無し
ServiceObjectDescription	1.11	String	R	Open
ServiceObjectVersion	1.11	Long	R	Open
DeviceDescription	1.11	String	R	Open
DeviceName	1.11	String	R	Open

専用				
CapDiscrepancy	1.11	Boolean	R	Open
CapFullSensor	1.11	Boolean	R	Open
CapJamSensor	1.11	Boolean	R	Open
CapNearFullSensor	1.11	Boolean	R	Open
CapPauseDeposit	1.11	Boolean	R	Open
CapRealTimeData	1.11	Boolean	R	Open
CurrencyCode	1.11	String	R/W	Open
DepositAmount	1.11	Long	R	Open
DepositCashList	1.11	String	R	Open
DepositCodeList	1.11	String	R	Open
DepositCounts	1.11	String	R	Open
DepositStatus	1.11	Long	R	Open, Claim, & Enable
FullStatus	1.11	Long	R	Open
RealTimeDataEnabled	1.11	Boolean	R/W	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.11	無し
Close	1.11	Open
ClaimDevice	1.11	Open
ReleaseDevice	1.11	Open & Claim
CheckHealth	1.11	Open, Claim, & Enable
ClearInput	1.11	Open & Claim
ClearInputProperties	1.11	*適用外*
ClearOutput	1.11	*適用外*
CompareFirmwareVersion	1.11	Open, Claim, & Enable
DirectIO	1.11	Open
ResetStatistics	1.11	Open, Claim, & Enable
RetrieveStatistics	1.11	Open, Claim, & Enable
UpdateFirmware	1.11	Open, Claim, & Enable
UpdateStatistics	1.11	Open, Claim, & Enable
専用		
AdjustCashCounts	1.11	Open, Claim, & Enable
BeginDeposit	1.11	Open, Claim, & Enable
EndDeposit	1.11	Open, Claim, & Enable
FixDeposit	1.11	Open, Claim, & Enable
PauseDeposit	1.11	Open, Claim, & Enable
ReadCashCounts	1.11	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.11	Open, Claim&Enable
DirectIOEvent	1.11	Open, Claim
ErrorEvent	1.11	*適用外*
OutputCompleteEvent	1.11	*適用外*
StatusUpdateEvent	1.11	Open, Claim, & Enable

概説

紙幣入金機コントロールのOLEプログラマティクIDは“OPOS.BillAcceptor”です。

このデバイスは第1.11版にて追加されました。

機能

紙幣入金機は次の機能を持っています。

紙幣入金機で利用できる金種と、対応する枚数を報告します。

デバイス内の紙幣詰まり状態を報告します。

複数の通貨をサポートします。

また、次の機能は、オプションとして実装されます。

紙幣入金機の金種の充填レベルを報告します。示される状態としては、フル、ニアフルがあります。

ReadCashCountsメソッドで報告されるデータに、現金在高不一致がありそう（もしくは、ある）かどうかが報告されます。

入金開始から入金終了の間に入金された紙幣は入金されるたびにその金種と対応する枚数を報告します。

モデル

紙幣入金機の一般的なモデルは次の通りです。

紙幣入金機は、複数の額面種類の紙幣サポートします。各通貨毎にサポートされる現金タイプは、DepositCashListプロパティの金種リストに記述されています。

紙幣入金機デバイスは、現金投入口、現金を格納する複数のスロットや収納庫、現金出口といった現金処理機能を援助する各種の要素から構成されています。

紙幣入金機デバイスが現金の在高を認識できない場合は、デバイスからの現金の取り出しの都度AdjustCashCountsメソッドで在高を制御します。アプリケーションはReadCashCountsメソッドを呼び出して、それぞれのユニットの在高を確認することができます。

現金スロット（もしくは現金収納庫）の状態は、フルとニアフルを示すFullStatusプロパティに設定されます。1つ以上の現金スロットがフルならFullStatusプロパティはBACC_STATUS_FULLとなります。

アプリケーションがBeginDepositメソッドを呼び出した時、DepositCountsプロパティとDepositAmountプロパティはゼロに初期化されます。

CapPauseDepositプロパティがTRUEなら、アプリケーションはBACC_DEPOSIT_PAUSEパラメータのPauseDepositメソッドを呼び出して入金を一時停止することができます。その時、DepositCountsとDepositAmountプロパティの値が確定します。アプリケーションが入金を再開したいなら、BACC_DEPOSIT_RESTARTのパラメータ値でPauseDepositメソッドを再度呼び出します。DepositCountsプロパティかDepositAmountプロパティを読み取り、アプリケーションは正確な入金を得ることができ、入金合計を販売合計と比較した結果をチェックできます。入金合計≧販売合計なら、アプリケーションはFixDepositメソッドを呼び出します。入金合計＜販売合計なら、アプリケーションはBACC_DEPOSIT_RESTARTでPauseDepositメソッドを呼び出します。入金総額を得るため、アプリケーションはPauseDepositメソッドを繰り返し呼び出せます。

ClearInputメソッドが実行された時、入金に伴いキューされた全てのDataEventはクリアされますが、DepositCountsプロパティとDepositAmountプロパティはクリアされません。

デバイスの共有

紙幣入金機は次の通り、排他的な利用をするデバイスです。

アプリケーションはデバイスをイネーブルにする前に、デバイスの排他アクセス権を獲得しなければなりません。

アプリケーションは、プロパティにアクセスしたり、払い出しや回収を行ったり、状態更新イベントを受け取る前に、デバイスの排他アクセス権を獲得しイネーブルにしなければなりません。

使用上の必要条件の詳細は、「一覧」の表を参照してください。

プロパティ

CapDiscrepancy プロパティ

形式　　BOOL CapDiscrepancy;

説明　　TRUE：**ReadCashCounts**メソッドは*pDiscrepancy*に有効な値を設定します。
FALSE：サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　　**ReadCashCounts**メソッド

CapFullSensor プロパティ

形式　　BOOL CapFullSensor;

説明　　TRUE：紙幣入金機は、フルになっている現金スロットがあることを報告します。
FALSE：サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　　**FullStatus**プロパティ、**StatusUpdateEvent**イベント

CapJamSensor プロパティ

形式　　BOOL CapJamSensor;

説明　　TRUE：紙幣入金機は、紙幣詰まりもしくは故障している現金スロットがあることを報告します。
FALSE：サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　　**StatusUpdateEvent**イベント

CapNearFullSensor プロパティ

形式　　BOOL CapNearFullSensor;

説明　　TRUE：紙幣入金機は、ニアフルになっている現金スロットがあることを報告します。
FALSE：サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**FullStatusプロパティ、StatusUpdateEventイベント**

CapPauseDeposit プロパティ

形式　**BOOL CapPauseDeposit;**

説明　TRUE：紙幣入金機は、入金機構を一時停止する機能があります。
FALSE：一時停止する機能はありません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**PauseDepositメソッド**

CapRealTimeData プロパティ

形式　**BOOL CapRealTimeData;**

説明　TRUE：紙幣入金機は、リアルタイムに入金情報を通知する機能があります。
FALSE：リアルタイムに入金情報を通知する機能はありません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**RealTimeDataEnabledプロパティ**

CurrencyCode プロパティ　R/W

形式　**BSTR CurrencyCode;**

説明　紙幣入金機の操作で現在使用されている通貨コードが設定されます。

このプロパティは**Open**メソッドで適当な値に初期化されます。有効な値は、**CurrencyCodeList**プロパティに記される通貨の中の一つです。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	このプロパティは正常に設定されました。
OPOS_E_ILLEGAL	**CurrencyCodeList**以外の値が指定されました。

参照　**CurrencyCodeListプロパティ**

DepositAmount プロパティ

形式　　LONG DepositAmount;

説明　　入金された合計金額を表します。

例えば、CurrencyCodeプロパティが日本円（JPY）で、
DepositAmountプロパティで表される値が、
　18057
なら、18,057円がBeginDepositメソッドの呼び出し以降の入金された現金金額となります。

参照　　CurrencyCodeプロパティ

DepositCashList プロパティ

形式　　BSTR DepositCashList;

説明　　CurrencyCodeプロパティで示される通貨での、紙幣入金機がサポートする金種を表す文字列です。入金機構がサポートされていない場合は空の文字列になります。

文字列は、ASCIIセミコロン(“;”)から、紙幣の金種を示すコンマで区切られたASCII数字が続きます。

下記は、日本での、DepositCashListの値の例です。

- “;1000,5000,10000” —
  1000、5000、10000円紙幣。

このプロパティはOpenメソッドにより初期化され、CurrencyCodeが設定された時に更新されます。

参照　　CurrencyCodeプロパティ

DepositCodeList プロパティ

形式　　BSTR DepositCodeList;

説明　　入金の通貨コードの文字列を表します。

文字列は、コンマで区切られたASCII 3 文字のISO 4217通貨コードにより構成されます。

例えば、文字列が”JPY,USD”なら、紙幣入金機は日本とU.S.の通貨単位をサポートします。

このプロパティはOpenメソッドで初期化されます。

参照　CurrencyCodeプロパティ、DepositCashListプロパティ

DepositCounts プロパティ

形式　BSTR DepositCounts;

説明　入金された金種の文字列を表します。文字列内の金種はDepositCashListプロパティと同じで、同じ順序で並んでいます。

例えば、日本円（JPY）で、DepositCountsプロパティで表される文字列が、
　1000:80, 5000:77, 10000:0
なら、80枚の1000円紙幣と、77枚の5000円紙幣がBeginDepositメソッドの呼び出し以降の入金された現金枚数となります。

参照　CurrencyCodeプロパティ

DepositStatus プロパティ

形式　LONG DepositStatus;

説明　紙幣入金機の入金機構の現在の状態を示します。次のいずれかとなります。

値	意味
BACC_STATUS_DEPOSIT_START	入金可能状態です。
BACC_STATUS_DEPOSIT_END	入金停止状態です。
BACC_STATUS_DEPOSIT_COUNT	入金計数中または返却中です。
BACC_STATUS_DEPOSIT_JAM	機器障害が発生しました。

このプロパティはイネーブルの設定で初期化され、イネーブルの間は最新の状態になります。このプロパティはBACC_STATUS_DEPOSIT_ENDに初期化されます。

FullStatus プロパティ

形式　LONG FullStatus;

説明　現金スロットの現在のフル状態を示します。次のいずれかとなります。

値	意味
BACC_STATUS_OK	ニアフル、フルの現金スロットは有りません。
BACC_STATUS_FULL	フルの現金スロットが有ります。
BACC_STATUS_NEARFULL	ニアフルの現金スロットが有ります。

このプロパティは初期化された後、デバイスがイネーブルの間は最新の状態となります。

RealTimeDataEnabled プロパティ

形式　**BOOL RealTimeDataEnabled;**

説明　TRUE：**CapRealTimeData**プロパティがTRUEであれば、**DepositAmount**と**DepositCounts**プロパティはイベントの通知と同時に更新されます。
FALSE：**DepositAmount**と**DepositCounts**プロパティは**FixDeposit**メソッドの呼び出すことにより更新されます。

RealTimeDataEnabledプロパティの設定は、**BeginDeposit**メソッドが実行されるまでの間のほかの動作には影響を与えません。このことは、**BeginDeposit**メソッドと**EndDeposit**メソッドの間の変更によって引き起こされるかもしれない混乱を防ぎます。

このプロパティは、**Open**メソッドによりFALSEに初期化されます。

参照　**CapRealTimeData**プロパティ、**DepositAmount**プロパティ、**DepositCounts**プロパティ、**BeginDeposit**メソッド、**EndDeposit**メソッド、**FixDeposit**メソッド

メソッド

AdjustCashCounts メソッド

形式　LONG AdjustCashCounts (BSTR *CashCounts*);

パラメータ	意味
CashCounts	初期化する現金枚数データを指定します。

説明　このメソッドは、紙幣入金機を導入した後に現金在高の初期データを設定する際や、現金の補充や回収などを行った後に現金在高を補正するために呼び出されます。

また、自動的には現金在高が確定できないデバイスの場合、在高を確定するために明示的に呼び出します。もしデバイスが自動的に現金在高を確定することができる場合、呼び出しは無視されます。

アプリケーションは、はじめに**ReadCashCount**メソッドを呼び出して現金在高を取得し、その内容を補充する金額に従って変更します。その後、アプリケーションはこのメソッドを呼び出して、紙幣入金機に現金在高を設定します。

現金在高を全て０にする場合には、全ての金種の在高に０を設定します。

例えば、通貨が日本円で、*CashCounts*パラメータに
　1000:80,5000:77,10000:0
が指定されて**AdjustCashCounts**メソッドが呼び出されたとき、80枚の1000円紙幣と、77枚の5000円紙幣が紙幣入金機の現金在高に設定されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に終了しました。
その他	**ResultCode**の項目を参照してください。

参照　**ReadCashCounts**メソッド

BeginDeposit メソッド

形式　LONG BeginDeposit ();

説明　紙幣入金機の入金処理を開始します。

このメソッドの呼び出しで以下のプロパティ値は初期化されます。

DepositCountsプロパティは各金種の枚数値が0にセットされます。

DepositAmountプロパティは0がセットされます。

このメソッドが呼び出されると、**FixDeposit**メソッドが呼び出されるまでの間、入金処理が一時停止にされていない限り、紙幣入金機が紙幣を受け付けるたびに**DataEvent**イベントにて入金イベントが報告されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	呼び出しシーケンスが正しくありません。
その他	**ResultCode**の項目を参照してください。

参照　**DepositCounts**プロパティ、**DepositAmount**プロパティ、**EndDeposit**メソッド,**FixDeposit**メソッド、**PauseDeposit**メソッド

EndDeposit メソッド

形式　　**LONG EndDeposit (LONG** *Success***);**

*Success*パラメータには、入金した現金の処理を設定します。この値は、以下のいずれかの値です。

値	意味
BACC_DEPOSIT_COMPLETE	入金した現金を収納し、入金処理を終了。

説明　　紙幣入金機の入金処理を終了します。

このメソッドを呼び出す前に、アプリケーションは釣り銭額の計算をする必要があります。

このメソッドを呼び出す前には必ず**FixDeposit**メソッドにて入金の確定を行ってください。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	次に示すいずれかのエラーが発生しています。 呼び出しシーケンスが正しくありません。このメソッドを呼び出す前に**BeginDeposit**メソッドおよび**FixDeposit**メソッドを呼び出す必要があります。
その他	**ResultCode**の項目を参照してください。

参照　　**DepositCounts**プロパティ、**DepositAmount**プロパティ、**BeginDeposit**メソッド、**FixDeposit**メソッド、**PauseDeposit**メソッド

FixDeposit メソッド

形式　LONG FixDeposit ();

説明　紙幣入金機の入金確定を行います。このとき、紙幣入金機は入金処理を停止し、すべてのプロパティの値が確定します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	次に示すいずれかのエラーが発生しています。 呼び出しシーケンスが正しくありません。このメソッドを呼び出す前に**BeginDeposit**メソッドを呼び出す必要があります。
その他	**ResultCode**の項目を参照してください。

参照　**DepositCounts**プロパティ、**DepositAmount**プロパティ、**BeginDeposit**メソッド、**EndDeposit**メソッド、**PauseDeposit**メソッド

PauseDeposit メソッド

形式　　**LONG PauseDeposit (LONG *Control*);**

このメソッドは入金処理の一時停止を行うか、入金処理を再開するときに使います。*Control*パラメータの値は、以下のいずれかの値です。

値	意味
BACC_DEPOSIT_PAUSE	入金の一時停止。
BACC_DEPOSIT_RESTART	入金の再開。

説明　　入金処理を一時停止または再開します。

*Control*がBACC_DEPOSIT_PAUSEなら、入金処理を一時停止します。このメソッドがBACC_DEPOSIT_RESTARTで呼び出されるまで、入金処理は一時停止状態になります。また、入金処理が一時停止の状態で、**FixDeposit**メソッドを呼び出すことが出来ます。

入金処理が一時停止するとき、**DepositCounts**プロパティと**DepositAmount**プロパティは紙幣入金機の現在の状態に更新されます。入金処理が再開されるまで、上記プロパティ値は更新されません。

*Control*がBACC_DEPOSIT_RESTARTなら、入金処理は再開されます。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	次に示すいずれかのエラーが発生しています。 呼び出しシーケンスが正しくありません。このメソッドを呼び出す前に**BeginDeposit**メソッドを呼び出す必要があります。 すでに一時停止している状態で、BACC_DEPOSIT__PAUSEが発行されたか、一時停止していない状態で、BACC_DEPOSIT__RESTARTが発行された。
その他	**ResultCode**の項目を参照してください。

参照　　**DepositCountsプロパティ**、**DepositAmountプロパティ**、**BeginDepositメソッド**、**EndDepositメソッド**、**FixDepositメソッド**

ReadCashCounts メソッド

形式　LONG ReadCashCounts (BSTR* *pCashCounts*, BOOL* *pDiscrepancy*);

パラメータ	意味
pCashCounts	現金枚数データを格納するエリアを示します。
pDiscrepancy	TRUE： *pCashCounts*で報告された現金枚数に含まれない現金が有ります。 FALSE：現金有り高は*pCashCounts*の内容と一致します。

説明　*pCashCounts*で指定される文字列のフォーマットと順序はDepositCashListプロパティと同じです。

例えば、通貨が日本円で、ReadCashCountsメソッドの呼び出し結果として*pCashCounts*パラメータに返される文字列が、
　1000:80,5000:77,10000:0
なら、80枚の1000円紙幣と、77枚の5000円紙幣が紙幣入金機にあります。

通常、*pCashCounts*パラメータから計算される現金在高は、紙幣入金機内の現金在高に一致します。しかし、紙幣入金機内に計数不可能な現金があるといった場合に、食い違いが発生する場合があります。例えば、現金の検出・監視の能力を超える現金スロットのオーバーフローなどがあります。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に終了しました。
OPOS_E_BUSY	非同期メソッド実行中のため、金種枚数は読み出せませんでした。
その他	ResultCodeの項目を参照してください。

参照　DepositCashListプロパティ

イベント

DataEvent イベント

形式　　void DataEvent (LONG *Status*);

*Status*パラメータには、0が設定されます。

説明　　紙幣入金機の入金の現金枚数が変化した時に通知されます。

第　29 章
硬貨入金機

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.11	Boolean	R/W	*適用外*
BinaryConversion	1.11	Long	R/W	Open
CapCompareFirmwareVersion	1.11	Boolean	R	Open
CapPowerReporting	1.11	Long	R	Open
CapStatisticsReporting	1.11	Boolean	R	Open
CapUpdateFirmware	1.11	Boolean	R	Open
CapUpdateStatistics	1.11	Boolean	R	Open
CheckHealthText	1.11	String	R	Open
Claimed	1.11	Boolean	R	Open
DataCount	1.11	Long	R	Open
DataEventEnabled	1.11	Boolean	R/W	Open
DeviceEnabled	1.11	Boolean	R/W	Open & Claim
FreezeEvents	1.11	Boolean	R/W	Open
OpenResult	1.11	Long	R	無し
OutputID	1.11	Long	R	*適用外*
PowerNotify	1.11	Long	R/W	Open
PowerState	1.11	Long	R	Open
ResultCode	1.11	Long	R	無し
ResultCodeExtended	1.11	Long	R	Open
State	1.11	Long	R	無し
ControlObjectDescription	1.11	String	R	無し
ControlObjectVersion	1.11	Long	R	無し
ServiceObjectDescription	1.11	String	R	Open
ServiceObjectVersion	1.11	Long	R	Open
DeviceDescription	1.11	String	R	Open
DeviceName	1.11	String	R	Open

専用				
CapDiscrepancy	1.11	Boolean	R	Open
CapFullSensor	1.11	Boolean	R	Open
CapJamSensor	1.11	Boolean	R	Open
CapNearFullSensor	1.11	Boolean	R	Open
CapPauseDeposit	1.11	Boolean	R	Open
CapRealTimeData	1.11	Boolean	R	Open
CurrencyCode	1.11	String	R/W	Open
DepositAmount	1.11	Long	R	Open
DepositCashList	1.11	String	R	Open
DepositCodeList	1.11	String	R	Open
DepositCounts	1.11	String	R	Open
DepositStatus	1.11	Long	R	Open, Claim, & Enable
FullStatus	1.11	Long	R	Open, Claim, & Enable
RealTimeDataEnabled	1.11	Boolean	R/W	Open, Claim, & Enable

メソッド

共通	*版数*	*使用可能条件*
Open	1.11	無し
Close	1.11	Open
ClaimDevice	1.11	Open
ReleaseDevice	1.11	Open & Claim
CheckHealth	1.11	Open, Claim, & Enable
ClearInput	1.11	Open & Claim
ClearInputProperties	1.11	*適用外*
ClearOutput	1.11	*適用外*
CompareFirmwareVersion	1.11	Open, Claim, & Enable
DirectIO	1.11	Open
ResetStatistics	1.11	Open, Claim, & Enable
RetrieveStatistics	1.11	Open, Claim, & Enable
UpdateFirmware	1.11	Open, Claim, & Enable
UpdateStatistics	1.11	Open, Claim, & Enable
専用		
AdjustCashCounts	1.11	Open, Claim, & Enable
BeginDeposit	1.11	Open, Claim, & Enable
EndDeposit	1.11	Open, Claim, & Enable
FixDeposit	1.11	Open, Claim, & Enable
PauseDeposit	1.11	Open, Claim, & Enable
ReadCashCounts	1.11	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.11	Open, Claim&Enable
DirectIOEvent	1.11	Open, Claim
ErrorEvent	1.11	*適用外*
OutputCompleteEvent	1.11	*適用外*
StatusUpdateEvent	1.11	Open, Claim, & Enable

概説

硬貨入金機コントロールのOLEプログラマティクIDは“OPOS.CoinAcceptor”です。

このデバイスは第1.11版にて追加されました。

機能

硬貨入金機は次の機能を持っています。

硬貨入金機で利用できる金種と、対応する枚数を報告します。

現金受け取りの開始から終了までの間にデバイスに入金された硬貨をアプリケーションに報告します。報告される内容は、金種と枚数です。

デバイス内の障害状態を報告します。

複数の通貨をサポートします。

また、次の機能は、オプションとして実装されます。

硬貨入金機の金種の充填レベルを報告します。示される状態としては、フル、ニアフルがあります。

ReadCashCountsメソッドで報告されるデータに、現金枚数不一致がありそう（もしくは、ある）かどうかが報告されます。

モデル

硬貨入金機の一般的なモデルは次の通りです。

複数の金種をサポートします。特定の地域でサポートされる金種は、DepositCashListプロパティの金種に列挙されています。

この仕様では、*現金の入金のみ*をプログラムで制御できます。デバイスからの現金回収（例えば、蓄積された現金の取り出し）は、デバイスが自分自身内の現金有高を決定できないなら、AdjustCashCountsメソッドにより制御されます。アプリケーションは金種毎の現在の枚数を読み出すためにReadCashCountsメソッドを呼び出すことができますが、デバイスに何時・どれくらい追加されたかを制御することはできません。

複数の通貨をサポートする場合があります。DepositCodeListの通貨一覧から選択し、CurrecyCodeプロパティに通貨を設定します。DepositCashListとReadCashCountsの両者は、現在の通貨上で動作します。

現金スロット（もしくは現金収納庫）の状態をFullStatusに設定し、フルとニアフル状態を示します。一つ以上の現金スロットがフルなら、FullStatusはCACC_STATUS_FULLとなります。

“硬貨入機構”への硬貨入金は、BeginDepositメソッドを起動して始まります。DepositCountsとDepositAmountプロパティは前もってゼロに初期化されます。

デバイスに入金された現金合計は、FixDepositメソッドかPauseDepositメソッドが実行されるまで、加算され続けます。FixDepositメソッドが実行された時、蓄積された現金合計は、DepositCountsかDepositAmountプロパティに設定されます。PauseDepositメソッドがCACC_DEPOSIT_PAUSEパラメータで実行されれば、入金金額の算出は中断され、入金された合計はDepositCountsかDepositAmountプロパティに更新されます。PauseDepositメソッドがCACC_DEPOSIT_RESTARTパラメータで実行されれば、入金金額の算出は再開され、入金合計は加算されてゆきます。FixDepositメソッドが実行されると、蓄積された現金の現在の合計はDepositCountsとDepositAmountプロパティに更新され、入金を終了させるためにCACC_DEPOSIT_COMPLETEパラメータでEndDepositメソッドが起動されると停止状態になります。

ClearInputメソッドが実行されると、現金受け取りでキューされたDataEventはクリアされます。DepositCountsとDepositAmountプロパティは設定されたままとなり、クリアされません。

デバイスの共有

硬貨入金機は次の通り、排他的な利用をするデバイスです。

アプリケーションはデバイスをイネーブルにする前に、デバイスの排他アクセス権を獲得しなければなりません。

アプリケーションは、プロパティにアクセスしたり、払い出しや回収を行ったり、状態更新イベントを受け取る前に、デバイスの排他アクセス権を獲得しイネーブルにしなければなりません。

使用上の必要条件の詳細は、「一覧」の表を参照してください。

プロパティ

CapDiscrepancy プロパティ

形式　　BOOL CapDiscrepancy;

説明　　TRUE：**ReadCashCounts**メソッドは*pDiscrepancy*に有効な値を設定します。
FALSE：サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　　**ReadCashCounts**メソッド

CapFullSensor プロパティ

形式　　BOOL CapFullSensor;

説明　　TRUE：硬貨入金機は、フルになっている現金スロットがあることを報告します。
FALSE：サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　　**FullStatus**プロパティ、**StatusUpdateEvent**イベント

CapJamSensor プロパティ

形式　　BOOL CapJamSensor;

説明　　TRUE：コインディスペンサは硬貨詰まりや機器の異常を通知します。

FALSE：通知しません。

このプロパティは**Open**メソッドにより初期化されます。

CapNearFullSensor プロパティ

形式　　BOOL CapNearFullSensor;

説明　　TRUE：硬貨入金機は、ニアフルになっている現金スロットがあることを報告します。
FALSE：サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　　FullStatusプロパティ、StatusUpdateEventイベント

CapPauseDeposit プロパティ

形式　　BOOL CapPauseDeposit;

説明　　TRUE：硬貨入金機は、入金機構を一時停止する機能があります。
FALSE：一時停止する機能はありません。

このプロパティは、Openメソッドにより初期化されます。

参照　　PauseDepositメソッド

CapRealTimeData プロパティ

形式　　BOOL CapRealTimeData;

説明　　TRUE：硬貨入金機は、（“リアルタイムで”）入金データを報告できます。
FALSE：報告できません。

このプロパティは、Openメソッドにより初期化されます。

参照　　RealTimeDataEnabledプロパティ

CurrencyCode プロパティ　R/W

形式　BSTR CurrencyCode;

説明　硬貨入金機の操作で現在使用されている通貨コードが設定されます。

このプロパティは**Open**メソッドで適当な値に初期化されます。有効な値は、**DepositCodeList**プロパティに記される通貨の中の一つです。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	このプロパティは正常に設定されました。
OPOS_E_ILLEGAL	**CurrencyCodeList**以外の値が指定されました。

参照　**DepositCodeList**プロパティ

DepositAmount プロパティ

形式　　LONG DepositAmount;

説明　　入金された合計金額を表します。

例えば、CurrencyCodeプロパティが日本円（JPY）で、DepositAmountプロパティで表される値が、
18057
なら、18,057円がBeginDepositメソッドの呼び出し以降の入金された現金金額となります。

参照　　CurrencyCodeプロパティ

DepositCashList プロパティ

形式　　BSTR DepositCashList;

説明　　CurrencyCodeプロパティで示される通貨での、硬貨入金機がサポートする金種を表す文字列です。

文字列は、硬貨入金機で使用できる硬貨の金種を示すコンマで区切られたASCII数字と、ASCIIセミコロン(";")です。

下記は、日本での、DepositCashListの値の例です。

- "1,5,10,50,100,500;" —
  1、5、10、50、100、500円硬貨。

このプロパティはOpenメソッドにより初期化され、CurrencyCodeが設定された時に更新されます。

参照　　CurrencyCodeプロパティ

DepositCodeList プロパティ

形式　　**BSTR DepositCodeList;**

説明　　入金の通貨コードの文字列を表します。

文字列は、コンマで区切られたASCII 3 文字のISO 4217通貨コードにより構成されます。

例えば、文字列が”JPY,USD”なら、硬貨入金機は日本とU.S.の通貨単位をサポートします。

このプロパティは**Open**メソッドで初期化されます。

参照　　**CurrencyCodeプロパティ**

DepositCounts プロパティ

形式　　BSTR DepositCounts;

説明　　入金された金種の文字列を表します。文字列のフォーマットは**DepositCashList**プロパティと同じで、同じ順序で並んでいます。

例えば、通貨が日本円で、**DepositCounts**プロパティで表される文字列が、
　1:80,5:77,10:0,50:54,100:0,500:87
なら、80枚の1円硬貨と、77枚の5円硬貨と、54枚の50円硬貨、87枚の500円硬貨が**BeginDeposit**メソッドの呼び出し以降の入金された現金枚数となります。

参照　　CurrencyCodeプロパティ

DepositStatus プロパティ

形式　　LONG DepositStatus;

説明　　硬貨入金機の入金機構の現在の状態を示します。次のいずれかとなります。

値	意味
CACC_STATUS_DEPOSIT_START	入金可能状態です。
CACC_STATUS_DEPOSIT_END	入金停止状態です。
CACC_STATUS_DEPOSIT_COUNT	入金計数中または返却中です。
CACC_STATUS_DEPOSIT_JAM	機器障害が発生しました。

このプロパティはイネーブルの設定で初期化され、イネーブルの間は最新の状態になります。このプロパティはCACC_STATUS_DEPOSIT_ENDに初期化されます。

FullStatus プロパティ

形式　　LONG FullStatus;

説明　　現金スロットの現在のフル状態を示します。次のいずれかとなります。

値	意味
CACC_STATUS_OK	ニアフル、フルの現金スロットは有りません。
CACC_STATUS_FULL	フルの現金スロットが有ります。
CACC_STATUS_NEARFULL	ニアフルの現金スロットが有ります。

このプロパティは初期化された後、デバイスがイネーブルの間は最新の状態となります。

RealTimeDataEnabled プロパティ　R/W

形式　　**BOOL RealTimeDataEnabled;**

説明　　TRUEで**CapRealTimeData**がTRUEなら、データイベントの度に**DepositAmount**と**DepositCounts**プロパティを更新します。FALSEなら、**FixDeposit**が呼び出される時に金額が収集されて**DepositAmount**と**DepositCounts**が更新されます。**RealTimeDataEnabled**を設定しても、継続して**BeginDeposit**が実行されなければ、システムの動作に影響を与えません。このプロパティにより、**BeginDeposit**と**EndDeposit**の間の状態変化に対する混乱を防ぎます。

このプロパティは**Open**メソッドで初期化されます。

戻り値　　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	このプロパティは正常に設定されました。
OPOS_E_ILLEGAL	**CapRealTimeData**がFALSEのため設定できません。

参照　　**CapRealTimeDataプロパティ、DepositAmountプロパティ、DepositCountsプロパティ、BeginDepositメソッド、EndDepositメソッド、FixDepositメソッド**

メソッド

AdjustCashCounts メソッド

形式　LONG AdjustCashCounts (BSTR *CashCounts*);

パラメータ	説明
CashCount	*CashCount*パラメータは、初期化する現金の金種と枚数を指定します。

説明　最初の設置後にコンディスペンサの開始現金有高を設定したり、入出金操作などの補充・回収後の現金有高の補正を行うため、このメソッドは呼び出されます。コインディスペンサ内の正確な現金合計を自動的に判定できないデバイスでは、このメソッドの呼び出しが必要です。デバイスが正確な合計額を判定できるなら、このメソッドの呼び出しは無視されます。アプリケーションは最初に**ReadCashCounts**を呼び出して現在の有高を取得し、実際の有高に補正します。さらに、アプリケーションはこのメソッドを呼び出し、ディスペンサ内の現在の有高を設定するでしょう。

すべての現金有高をリセットするには、各々の金種をゼロに設定します。

例えば、通貨が日本円で*CashCounts*パラメータに“1:80,5:77,50:54,100:0,500:87”を設定して**AdjustCashCounts**メソッドを呼び出せば、0枚の1円と、77枚の5円と、54枚の50円と、0枚の100円と、87枚の500円の硬貨がコインディスペンサに有ります。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは成功しました。
その他	**ResultCode**の項目を参照してください。

参照　**ReadCashCounts**メソッド

BeginDeposit メソッド

形式　LONG BeginDeposit ();

説明　硬貨入金機の入金処理を開始します。

このメソッドの呼び出しで以下のプロパティ値は初期化されます。

DepositCountsプロパティは各金種の枚数値が0にセットされます。

DepositAmountプロパティは0がセットされます。

このメソッドの呼び出し後、入金処理が中断されずに**FixDeposit**が呼び出されるまで、**DataEvent**イベントで現金の入金が報告されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	呼び出しシーケンスが正しくありません。
その他	**ResultCode**の項目を参照してください。

参照　**DepositAmount**プロパティ、**DepositCounts**プロパティ、**EndDeposit**メソッド、**FixDeposit**メソッド、**PauseDeposit**メソッド

EndDeposit メソッド

形式　　**LONG EndDeposit (LONG *Success*);**

*Success*パラメータには、入金した現金の処理を設定します。この値は、以下のいずれかの値です。

値	意味
CACC_DEPOSIT_COMPLETE	現金が入金され、入金金額は必要な金額に等しいか少ないです。

説明　　入金処理を終了します。

このメソッドを呼び出す前に、アプリケーションは入金合計と必要な金額の差を計算する必要があります。

アプリケーションは、このメソッドを呼び出す前に、**FixDeposit**メソッドを呼び出さなければいけません。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	次に示すいずれかのエラーが発生しています。 ・呼び出しシーケンスが正しくありません。このメソッドを呼び出す前に**BeginDeposit**メソッドおよび**FixDeposit**メソッドを呼び出す必要があります。
その他	**ResultCode**の項目を参照してください。

参照　　**DepositAmount**プロパティ、**DepositCounts**プロパティ、**BeginDeposit**メソッド、**FixDeposit**メソッド、**PauseDeposit**メソッド

FixDeposit メソッド

形式　　**LONG FixDeposit ();**

説明　　このメソッドが呼び出されると、全てのプロパティの値は硬貨入金機の最新の値に更新されます。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	次に示すいずれかのエラーが発生しています。 ・呼び出しシーケンスが正しくありません。このメソッドを呼び出す前に**BeginDeposit**メソッドを呼び出す必要があります。
その他	**ResultCode**の項目を参照してください。

参照　　**DepositAmount**プロパティ、**DepositCounts**プロパティ、**BeginDeposit**メソッド、**EndDeposit**メソッド、**PauseDeposit**メソッド

PauseDeposit メソッド

形式　　**LONG PauseDeposit (LONG *Control*);**

*Control*パラメータの値は、以下のいずれかの値です。

値	意味
CACC_DEPOSIT_PAUSE	入金の一時停止。
CACC_DEPOSIT_RESTART	入金の再開。

説明　　入金処理を一時停止または再開します。

*Control*がCACC_DEPOSIT_PAUSEなら、入金処理を一時停止します。このメソッドがCACC_DEPOSIT_RESTARTで呼び出されるまで、入金処理は一時停止状態になります。また、入金処理が一時停止の状態で、**FixDeposit**メソッドを呼び出すことが出来ます。

入金処理が一時停止するとき、**DepositCounts**プロパティと**DepositAmount**プロパティは硬貨入金機の現在の状態に更新されます。入金処理が再開されるまで、上記プロパティ値は更新されません。

*Control*がCACC_DEPOSIT_RESTARTなら、入金処理は再開されます。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	次に示すいずれかのエラーが発生しています。 ・呼び出しシーケンスが正しくありません。このメソッドを呼び出す前に**BeginDeposit**メソッドを呼び出す必要があります。 ・すでに一時停止している状態で、CACC_DEPOSIT_PAUSEが発行されたか、一 時停止していない状態で、CACC_DEPOSIT_RESTARTが発行された。
その他	**ResultCode**の項目を参照してください。

参照　　CapPauseDepositプロパティ、DepositAmountプロパティ、DepositCountsプロパティ、BeginDepositメソッド、EndDepositメソッド、FixDepositメソッド

ReadCashCounts メソッド

形式　LONG ReadCashCounts (BSTR* *pCashCounts*, BOOL* *pDiscrepancy*);

パラメータ	意味
pCashCounts	現金枚数データを格納するエリアを示します。
pDiscrepancy	TRUE： *pCashCounts*で報告された現金枚数に含まれない現金が有ります。 FALSE：現金有り高は*pCashCounts*の内容と一致します。

説明　*pCashCounts*内の金種は**DepositCashList**プロパティと同じで、同じ順序で並んでいます。

例えば、通貨が日本円で、**ReadCashCounts**メソッドの呼び出し結果として*pCashCounts*パラメータに返される文字列が、
　1:80,5:77,10:0,50:54,100:0,500:87
なら、80枚の1円硬貨と、77枚の5円硬貨と、54枚の50円硬貨、87枚の500円硬貨が硬貨入金機に有ります。

CapDiscrepancyがFALSEなら、*pDiscrepancy*は常にFALSEです。

通常、*pCashCounts*パラメータから計算される現金有り高は、硬貨入金機内の現金有り高に一致します。しかし、硬貨入金機内に計数不可能な現金があるといった場合に、食い違いが発生する場合があります。例えば、現金の検出・監視の能力を超える現金スロットのオーバーフローなどがあります。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に終了しました。
OPOS_E_BUSY	非同期メソッド実行中のため、金種枚数は読み出せませんでした。
その他	**ResultCode**の項目を参照してください。

参照　**DepositCashList**プロパティ

イベント

DataEvent イベント

形式　　void DataEvent (LONG *Status*);

*Status*パラメータには、0が設定されます。

説明　　硬貨入金機に入金された時に通知されます。

StatusUpdateEvent イベント

形式　　void StatusUpdateEvent (LONG *Status*);

*Status*パラメータは、硬貨入金機の状態が設定されます。

値	意味

CACC_STATUS_FULL　フルの現金スロットが有ります。

CACC_STATUS_NEARFULL
　　ニアフルの現金スロットが有ります。

CACC_STATUS_FULLOK
　　フル、ニアフルの状態が解除されました。

CACC_STATUS_JAM機器障害が発生しました。

CACC_STATUS_JAMOK
　　機器障害は回復しました。

*電源通知の**StatusUpdateEvent**の値*
　　「第2章 共通プロパティ、メソッド、イベント」の**StatusUpdateEvent**の記述を参照。

説明　　硬貨入金機の状態が変化した時に通知されます。

デバイスがフル、ニアフル状態の検出センサを持っている時にのみ、フルレベルの変化に従い、硬貨入金機は**StatusUpdateEvent**を通知し、関連するケイパビリティプロパティを設定します。

障害が発生したなら、障害状態が通知されます。

第　３０章
紙幣出金機

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.11	Boolean	R/W	*適用外*
BinaryConversion	1.11	Long	R/W	Open
CapCompareFirmwareVersion	1.11	Boolean	R	Open
CapPowerReporting	1.11	Long	R	Open
CapStatisticsReporting	1.11	Boolean	R	Open
CapUpdateFirmware	1.11	Boolean	R	Open
CapUpdateStatistics	1.11	Boolean	R	Open
CheckHealthText	1.11	String	R	Open
Claimed	1.11	Boolean	R	Open
DataCount	1.11	Long	R	*適用外*
DataEventEnabled	1.11	Boolean	R/W	*適用外*
DeviceEnabled	1.11	Boolean	R/W	Open & Claim
FreezeEvents	1.11	Boolean	R/W	Open
OpenResult	1.11	Long	R	無し
OutputID	1.11	Long	R	*適用外*
PowerNotify	1.11	Long	R/W	Open
PowerState	1.11	Long	R	Open
ResultCode	1.11	Long	R	無し
ResultCodeExtended	1.11	Long	R	Open
State	1.11	Long	R	無し
ControlObjectDescription	1.11	String	R	無し
ControlObjectVersion	1.11	Long	R	無し
ServiceObjectDescription	1.11	String	R	Open
ServiceObjectVersion	1.11	Long	R	Open
DeviceDescription	1.11	String	R	Open
DeviceName	1.11	String	R	Open

専用				
CapDiscrepancy	1.11	Boolean	R	Open
CapEmptySensor	1.11	Boolean	R	Open
CapJamSensor	1.11	Boolean	R	Open
CapNearEmptySensor	1.11	Boolean	R	Open
AsyncMode	1.11	Boolean	R/W	Open
AsyncResultCode	1.11	Long	R	Open, Claim, & Enable
AsyncResultCodeExtended	1.11	Long	R	Open, Claim, & Enable
CurrencyCashList	1.11	String	R	Open
CurrencyCode	1.11	String	R/W	Open
CurrencyCodeList	1.11	String	R	Open
CurrentExit	1.11	Long	R/W	Open
DeviceExits	1.11	Long	R	Open
DeviceStatus	1.11	Long	R	Open, Claim, & Enable
ExitCashList	1.11	String	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.11	無し
Close	1.11	Open
ClaimDevice	1.11	Open
ReleaseDevice	1.11	Open & Claim
CheckHealth	1.11	Open, Claim, & Enable
ClearInput	1.11	*適用外*
ClearInputProperties	1.11	*適用外*
ClearOutput	1.11	Open, Claim, & Enable
CompareFirmwareVersion	1.11	Open, Claim, & Enable
DirectIO	1.11	Open
ResetStatistics	1.11	Open, Claim, & Enable
RetrieveStatistics	1.11	Open, Claim, & Enable
UpdateFirmware	1.11	Open, Claim, & Enable
UpdateStatistics	1.11	Open, Claim, & Enable
専用		
AdjustCashCounts	1.11	Open, Claim, & Enable
DispenseCash	1.11	Open, Claim, & Enable
ReadCashCounts	1.11	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.11	*適用外*
DirectIOEvent	1.11	Open, Claim
ErrorEvent	1.11	*適用外*
OutputCompleteEvent	1.11	*適用外*
StatusUpdateEvent	1.11	Open, Claim, & Enable

概説

紙幣出金機コントロールのOLEプログラマティックIDは“OPOS.BillDispenser”です。

このデバイスは、第1.11版で追加されました。

機能

紙幣出金機は、次の機能を持っています。

紙幣出金機で利用可能な金種と、対応枚数を報告します。

指定された枚数の金種を、ユーザ指定の払い出し口に払い出します。

デバイスの障害状態を通知します。

複数の通貨をサポートします。

次の機能はオプションとして実装されます。

紙幣出金機の金種の充填レベルを報告します。示される状態として、エンプティ，ニアエンプティがあります。

ReadCashCountsメソッドで報告されるデータに、現金枚数不一致がありそう（もしくはある）かどうかが報告されます。

モデル

紙幣出金機の一般的なモデルは、次のようなものです。

- 紙幣出金機は、数種類の単位の紙幣をサポートします。通貨毎にサポートされる紙幣は、**CurrencyCashList**プロパティの金種リストに記述されています。
- 紙幣出金機は、現金を格納する複数のスロットや収納庫、現金払い出し口といった現金処理機能を援助する各種の要素から構成されています。
- *現金の払い出しのみ*をプログラムで制御できます。デバイスが自分自身の現金有り高を決定できない場合、（例えば、現金補充のための）デバイスでの現金の受納は、**AdjustCashCounts**メソッドにより制御されます。アプリケーションは金種毎の現在の枚数を読み出すために**ReadCashCounts**メソッドを呼び出すことができますが、デバイスに何時・どれくらい追加されたかを制御することはできません。

- 自動釣り銭機は各種の払い出し口を持つことがあります。払い出し口の個数は**DeviceExits**プロパティで規定されます。アプリケーションは**CurrentExit**プロパティを設定して払い出し口を選択します。現在の払い出し口に払い出せる金種は、**ExitCashList**プロパティで表されます。**CurrentExit**の値が1なら、払い出し口は基本払い出し口となり、小売り販売に伴う顧客への現金払い出し処理に通常使用されます。**CurrentExit**の値が1より大きいなら、払い出し口はその他払い出し口_となります。その他払い出し口は、基本払い出し口を出口としない、多量の、もしくは別金種の現金の払い出しといった特殊な目的に通常使用されます。
- **CurrentExit**プロパティで指定される払い出し口へ現金を払い出すには、**DispenseCash**メソッドを使用します。
  DispenseCashメソッドでアプリケーションは払い出す金種毎の枚数を指定します。

- 現金の払い出しは、**AsyncMode**プロパティの値に従い、同期的、もしくは非同期的に実行されます。**AsyncMode**がFALSEなら、現金払い出しメソッドは同期的に実行され、払い出しメソッドはアプリケーションに終了ステータスを返します。
  AsyncModeがTRUEで、**DispenseCash**がOPOS_SUCCESSを返したら、メソッドは非同期的に実行され、終了はBDSP_STATUS_ASYNCを*Status*の値に持つ**StatusUpdateEvent**イベントで示されます。
  AsyncResultCodeと**AsyncResultCodeExtended**プロパティの値は、同期的な払い出しが選択された際に返される**ResultCode**と**ResultCodeExtended**プロパティの値と同じです。
  非同期的な自動釣り銭機の操作を多重に行うことは禁止されています。同時に処理される非同期的なメソッドは1つだけです。
  ReadCashCountsは、非同期メソッドが実行中の間、不正な現金枚数を報告する可能性があるため、実行できません。
- 紙幣出金機は複数の通貨をサポートできます。
  CurrencyCodeプロパティは、**CurrencyCodeList**の一覧から選択された通貨が設定されます。**CurrencyCashList**、**ExitCashList**プロパティと**DispenseCash**、**ReadCashCounts**メソッドは、現在設定されている通貨にて動作します。
- 現金スロット（もしくは現金収納庫）の状態は、エンプティとニアエンプティを示す**DeviceStatus**プロパティに設定されます。1つ以上の現金スロットがエンプティなら**DeviceStatus**プロパティはBDSP_STATUS_EMPTYとなります。

デバイスの共有

紙幣出金機は、次のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、プロパティにアクセスしたり、払い出しや回収、イベントを受け取る前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の必要条件については、「一覧」の表を参照してください。

プロパティ

AsyncMode プロパティ　R/W

形式　BOOL AsyncMode;

説明　TRUE：**DispenseCash**メソッドは、非同期に実行されます。
FALSE：**DispenseCash**メソッドは、同期的に実行されます。

このプロパティは**Open**メソッドにより初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

参照　**AsyncResultCode** プロパティ、**AsyncResultCodeExtended**プロパティ、 **DispenseCash**メソッド

AsyncResultCode プロパティ

形式　LONG AsyncResultCode;

説明　最後の非同期の払い出しの処理が完了した時（**DispenseCash**メソッドが**AsyncMode**=TRUEで呼び出された時）に終了ステータスを返します。このプロパティは、CHAN_STATUS_ASYNCの*Status*値の**StatusUpdateEvent**イベントで通知する前に、コントロールにより設定されます。このプロパティの値は、メソッドが同期的に実行された際の**ResultCode**プロパティの値と同じです。

参照　**AsyncMode**プロパティ、**DispenseCash**メソッド

AsyncResultCodeExtended プロパティ

形式　LONG AsyncResultCodeExtended;

説明　最後の非同期の払い出しの処理が完了した時（**DispenseCash**メソッドが**AsyncMode**=TRUEで呼び出された時）に終了ステータスを返します。このプロパティは、CHAN_STATUS_ASYNCの*Status*値の**StatusUpdateEvent**イベントで通知する前に、コントロールにより設定されます。このプロパティの値は、メソッドが同期的に実行された際の**ResultCodeExtended**プロパティの値と同じです。

参照　**AsyncMode**プロパティ、**DispenseCash**メソッド

CapDiscrepancy プロパティ

形式　BOOL CapDiscrepancy;

説明　TRUE：**ReadCashCounts**メソッドは、*pDiscrepancy*に有効な値を設定します。

FALSE：サポートしません。

このプロパティは**Open**メソッドにより初期化されます。

参照　**ReadCashCounts**メソッド

CapEmptySensor プロパティ

形式　BOOL CapEmptySensor;

説明　TRUE：紙幣出金機は、エンプティになっている現金スロットがあることを報告します。

FALSE：サポートしません。

このプロパティは**Open**メソッドにより初期化されます。

参照　**DeviceStatus** プロパティ, **StatusUpdateEvent**.イベント

CapJamSensor プロパティ

形式　BOOL CapJamSensor;

説明　TRUE：紙幣出金機は、機器異常を通知します。

FALSE：サポートしません。

このプロパティは**Open**メソッドにより初期化されます。

参照　**DeviceStatus** プロパティ, **StatusUpdateEvent**.イベント

CapNearEmptySensor プロパティ

形式　　BOOL CapNearEmptySensor;

説明　　TRUE：紙幣出金機は、ニアエンプティになっている現金スロットがあることを報告します。

FALSE：サポートしません。

このプロパティはOpenメソッドにより初期化されます。

参照　　**DeviceStatus** プロパティ, **StatusUpdateEvent**.イベント

CurrencyCashList プロパティ

形式　　BSTR CurrencyCashList;

説明　　CurrencyCodeプロパティで示される通貨での、紙幣出金機がサポートする金種を表す文字列です。

文字列は、紙幣出金機で使用できる紙幣の金種を示すコンマで区切られたASCII数字がASCIIセミコロン(“;”)の後に続きます。
セミコロン(“;”)があれば、金種は紙幣を表します。これは、多重デバイスサービスを自動釣銭機にマージするために使用されます。

下記は、日本での、CurrencyCashListの値の例です。

- “;1000,5000,10000” —
  1000、5000、10000円紙幣。

このプロパティはOpenメソッドにより初期化され、CurrencyCodeが設定された時に更新されます。

参照　　CurrencyCodeプロパティ

CurrencyCode プロパティ　R/W

形式　BSTR CurrencyCode;

説明　紙幣出金機の操作で現在使用されている通貨コードが設定されます。

このプロパティは**Open**メソッドで適当な値に初期化されます。有効な値は、**CurrencyCodeList**プロパティに記される通貨の中の一つです。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	このプロパティは正常に設定されました。
OPOS_E_ILLEGAL	**CurrencyCodeList**以外の値が指定されました。

参照　**CurrencyCodeList**プロパティ

CurrencyCodeList プロパティ

形式　BSTR CurrencyCodeList;

説明　通貨コードの文字列を表します。

文字列は、コンマで区切られたASCII 3 文字のISO 4217通貨コードにより構成されます。

例えば、文字列が”JPY,USD”なら、紙幣出金機は日本とU.S.の通貨単位をサポートします。

このプロパティは**Open**メソッドで初期化されます。

参照　**CurrencyCode**プロパティ

CurrentExit プロパティ　R/W

形式　**LONG CurrentExit;**

説明　現在の払い出し口を示します。値が１なら基本（もしくは通常）の払い出し口を示し、１より大きい値ならその他の払い出し口を示します。適切な値は、１から**DeviceExits**までの間です。
このプロパティは**Open**メソッドにより初期化されます。

下例は、日本での典型的なプロパティ値集合のサンプルです。
CurrencyCode 、**CurrencyCodeList** ともに“JPY”です。

- 紙幣出金機は紙幣をサポート; その他払い出し口は大量の紙幣払い出しに使用:
  CurrencyCashList = “;1000,5000,10000”
  DeviceExits = 2
  CurrentExit = 1の時: **ExitCashList** = “;1000,5000”
  CurrentExit = 2の時: **ExitCashList** = “;1000,5000,10000”

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_ ILLEGAL	無効な**CurrentExit**の値が指定されました。

参照　**CurrencyCashList** プロパティ、**DeviceExits** プロパティ、**ExitCashList** プロパティ

DeviceExits プロパティ

形式　　**LONG DeviceExits;**

説明　　現金の払い出し口の個数が設定されています。

このプロパティは**Open**メソッドにより初期化されます。

参照　　CurrentExit プロパティ

DeviceStatus プロパティ

形式　　**LONG DeviceStatus;**

説明　　紙幣出金機の現在の状態を示します。次のいずれかとなります。

値	意味
BDSP_STATUS_OK	紙幣出金機の現在の状態は正常です。
BDSP_STATUS_EMPTY	エンプティの現金スロットが有ります。
BDSP_STATUS_NEAREMPTY	ニアエンプティの現金スロットが有ります。
BDSP_STATUS_JAM	機器障害が発生しました。

このプロパティは初期化された後、デバイスがイネーブルの間は最新の状態となります。複数の状態が発生した時、優先順位は障害、エンプティ、ニアエンプティの順となります。

ExitCashList プロパティ

形式　　**BSTR ExitCashList;**

説明　　**CurrentExit**プロパティで示された払い出し口へ払い出せる金種の文字列です。サポートされる金種は、**CurrencyCashList**と同じかその一部です。文字列のフォーマットは**CurrencyCashList**と同じです。

このプロパティは**Open**メソッドで初期化され、**CurrencyCode**または**CurrentExit**が設定された時に更新されます。

参照　　**CurrencyCode**プロパティ、**CurrencyCashList**プロパティ、**CurrentExit**プロパティ

メソッド

AdjustCashCounts メソッド

形式　LONG AdjustCashCounts (BSTR *CashCounts*);

パラメータ	説明
CashCounts	*CashCount*パラメータは、初期化する現金の金種と枚数を指定します。

説明　最初の設置後にコンディスペンサの開始現金有高を設定したり、入出金操作などの補充・回収後の現金有高の補正を行うため、このメソッドは呼び出されます。コインディスペンサ内の正確な現金合計を自動的に判定できないデバイスでは、このメソッドの呼び出しが必要です。デバイスが正確な合計額を判定できるなら、このメソッドの呼び出しは無視されます。アプリケーションは最初に**ReadCashCounts**を呼び出して現在の有高を取得し、実際の有高に補正します。さらに、アプリケーションはこのメソッドを呼び出し、ディスペンサ内の現在の有高を設定するでしょう。

全ての現金有高をリセットするには、各々の金種をゼロに設定します。

例えば、通貨が日本円で、**ReadCashCounts**メソッドの呼び出し結果として*CashCounts*パラメータに返される文字列が、
1000:80,5000:77,10000:0,50000:54,100:0,500000:87
なら、80枚の1000円紙幣と、77枚の5000円紙幣と、54枚の50000円紙幣、87枚の500000円紙幣が紙幣出金機に有ります。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは成功しました。
OPOS_E_BUSY	非同期メソッドによって金種と枚数が初期化されていません。
その他	**ResultCode**の項目を参照してください。

参照　**ReadCashCounts**メソッド

DispenseCash メソッド

形式　　LONG DispenseCash (BSTR *CashCounts*);

パラメータ	説明
CashCounts	金種:枚数, ..;..金種:枚数のフォーマットで表された払い出す金種と枚数です。 紙幣であることを表すためには、金種の前に";"が必要です。

説明　　**CurrentExit**で指定された払い出し口に、自動釣り銭機から現金を払い出します。払い出す現金は金種と枚数のペアで指定されます。

AsyncModeがFALSEならこのメソッドは同期的に実行され、**AsyncMode**がTRUEなら非同期的に実行されます。

通貨として日本円を使用した際の、*CashCounts*の例は下記の通りです。

- ";1000:10"
  10枚の1000円紙幣を払い出します。
- ";1000:10,10000:5"
  10枚の1000円紙幣と、5枚の10000円紙幣を払い出します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	指定の現金は正常に払い出されました、または**DispenseCash**メソッドは非同期的に実行されました。
OPOS_E_BUSY	非同期メソッドが実行中のため、現金は払い出せません。
OPOS_E_ILLEGAL	*CashCounts*パラメータの値は、現在の払い出し口に対して不正です。または、入金動作が実行中のため現金は払い出せません
OPOS_E_EXTENDED	**ResultCodeExtended** = OPOS_ECHAN_OVERDISPENSE：現金不足で、指定の現金を払い出せません。
その他	**ResultCode**の項目を参照してください。

参照　**AsyncMode** プロパティ, **CurrentExit** プロパティ

.

ReadCashCounts メソッド

形式　LONG ReadCashCounts (BSTR* *pCashCounts*,BOOL* *pDiscrepancy*);

パラメータ	説明
pCashCounts	現金枚数データを格納するエリアを示します。
pDiscrepancy	TRUE： *pCashCounts*で報告された現金枚数に含まれない現金が有ります。 FALSE：現金有り高は*pCashCounts*の内容と一致します。

説明　*pCashCounts*で指定される文字列のフォーマットはDispenseCashメソッド内の*CashCounts*と同じです。*pCashCounts*内の金種はCurrencyCashListプロパティと同じで、同じ順序で並んでいます。

例えば、通貨が日本円で、ReadCashCountsメソッドの呼び出し結果として*pCashCounts*パラメータに返される文字列が、
1000:80,5000:77,10000:0,50000:54,100:0,500000:87
なら、80枚の1000円紙幣と、77枚の5000円紙幣と、54枚の50000円紙幣、87枚の500000円紙幣が紙幣出金機に有ります。

CapDiscrepancyがFALSEなら、*pDiscrepancy*は常にFALSEです。

通常、*pCashCounts*パラメータから計算される現金有り高は、紙幣出金機内の現金有り高に一致します。しかし、紙幣出金機内に計数不可能な現金があるといった場合に、食い違いが発生する場合があります。
例えば、使用できない紙幣を特定場所に保持することがあります。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常に終了しました。
OPOS_E_BUSY	非同期メソッド実行中のため、金種枚数は読み出せませんでした。
その他	ResultCodeの項目を参照してください。

参照　CapDiscrepancy プロパティ、CurrencyCashList プロパティ、DispenseCash メソッド

イベント

StatusUpdateEvent イベント

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

*Status*パラメータは、紙幣出金機の状態が設定されます。

値	意味
BDSP_STATUS_EMPTY	エンプティの紙幣スロットが有ります。
BDSP_STATUS_NEAREMPTY	ニアエンプティの紙幣スロットが有ります。
BDSP_STATUS_EMPTYOK	エンプティ、ニアエンプティの状態が解除されました。
BDSP_STATUS_JAM	機器障害が発生しました。
BDSP_STATUS_JAMOK	機器障害から回復しました。
BDSP_STATUS_ASYNC	非同期メソッドの実行が終了しました。
*電源通知の**StatusUpdateEvent**の値*	「第2章 共通プロパティ、メソッド、イベント」の**StatusUpdateEvent**の記述を参照。

説明　　紙幣出金機の状態が変化した際に通知されます。

デバイスがフル、ニアフル、エンプティ、ニアエンプティ状態の検出センサを持ち、これらの状態の機能プロパティが設定されている時にのみ、自動釣り銭機は釣り銭の充填レベルを**StatusUpdateEvent**で通知できます。

障害が発生したなら、障害状態が通知されます。これは非同期メソッド実行の終了と同じです。

非同期メソッド実行の終了ステータスは、AsyncResultCodeとAsyncResultCodeExtendedプロパティに設定されます。

参照　　**AsyncResultCode** プロパティ, **AsyncResultCodeExtended** プロパティ

第　3 1 章
イメージスキャナ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1. 11	Boolean	R/W	Open
BinaryConversion	1. 11	Long	R/W	Open
CapCompareFirmwareVersion	1. 11	Boolean	R	Open
CapPowerReporting	1. 11	Long	R	Open
CapStatisticsReporting	1. 11	Boolean	R	Open
CapUpdateFirmware	1. 11	Boolean	R	Open
CapUpdateStatistics	1. 11	Boolean	R	Open
CheckHealthText	1. 11	String	R	Open
Claimed	1. 11	Boolean	R	Open
DataCount	1. 11	Long	R	Open
DataEventEnabled	1. 11	Boolean	R/W	Open
DeviceEnabled	1. 11	Boolean	R/W	Open & Claim
FreezeEvents	1. 11	Boolean	R/W	Open
OpenResult	1. 11	Long	R	無し
OutputID	1. 11	Long	R	*適用外*
PowerNotify	1. 11	Long	R/W	Open
PowerState	1. 11	Long	R	Open
ResultCode	1. 11	Long	R	無し
ResultCodeExtended	1. 11	Long	R	Open
State	1. 11	Long	R	無し
ControlObjectDescription	1. 11	String	R	無し
ControlObjectVersion	1. 11	Long	R	無し
ServiceObjectDescription	1. 11	String	R	Open
ServiceObjectVersion	1. 11	Long	R	Open
DeviceDescription	1. 11	String	R	Open
DeviceName	1. 11	String	R	Open
専用				
CapAim	1. 11	Boolean	R	Open
CapDecodeData	1. 11	Boolean	R	Open

CapHostTriggered	1. 11	Boolean	R	Open
CapIlluminate	1. 11	Boolean	R	Open
CapImageData	1. 11	Boolean	R	Open
CapImageQuality	1. 11	Boolean	R	Open
CapVideoData	1. 11	Boolean	R	Open
AimMode	1. 11	Boolean	R/W	Open
BitsPerPixel	1. 11	Long	R	Open
FrameData	1. 11	String	R	Open
FrameType	1. 11	Long	R	Open
IlluminateMode	1. 11	Boolean	R/W	Open
ImageHeight	1. 11	Long	R	Open
ImageLength	1. 11	Long	R	Open
ImageMode	1. 11	Long	R/W	Open
ImageQuality	1. 11	Long	R/W	Open
ImageType	1. 11	Long	R	Open
ImageWidth	1. 11	Long	R	Open
VideoCount	1. 11	Long	R/W	Open
VideoRate	1. 11	Long	R/W	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1. 11	無し
Close	1. 11	Open
ClaimDevice	1. 11	Open
ReleaseDevice	1. 11	Open & Claim
CheckHealth	1. 11	Open, Claim, & Enable
ClearInput	1. 11	Open & Claim
ClearInputProperties	1.11	Open & Claim
ClearOutput	1. 11	*適用外*
CompareFirmwareVersion	1. 11	Open, Claim, & Enable
DirectIO	1. 11	Open
ResetStatistics	1. 11	Open, Claim, & Enable
RetrieveStatistics	1. 11	Open, Claim, & Enable
UpdateFirmware	1. 11	Open, Claim, & Enable
UpdateStatistics	1. 11	Open, Claim, & Enable
専用		
StartSession	1. 11	Open, Claim, & Enable
StopSession	1. 11	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1. 11	Open, Claim, & Enable
DirectIOEvent	1. 11	Open, Claim
ErrorEvent	1. 11	Open, Claim, & Enable
OutputCompleteEvent	1. 11	*適用外*
StatusUpdateEvent	1. 11	Open, Claim, & Enable

概説

イメージスキャナコントロールのOLEプログラマティクIDは“OPOS.ImageScanner”です。

このデバイスは、第1.11版で追加されました。

機能

イメージスキャナはそれ自身の引き金によって開始したスキャニングセッションから、イメージデータの単一フレームを読み込む能力を持っています。またイメージスキャナコントロールは、次の機能の１つ以上を持っています。（個々の内容についてはケイパビリティプロパティを参照）

バーコードラベルからエンコードデータを読み込みます。

照準の用途で低解像度ビデオストリームを読み込みます。

ホストはイメージスキャナのイルミネーション機能を制御することができます。

ホストはイメージスキャナの照準機能を制御することができます。

ホストはスキャンセッションを開始終了することができます。

モデル

イメージスキャナコントロールは、イベント駆動入力の一般的な入力モデルに従います。

コントロールは入力を受信したとき、DataEventを発生させます。

AutoDisableプロパティがTRUEならば、DataEventのキューイング時にコントロールは自動的にディセーブルとなります。AutoDisableプロパティは低解像度ビデオモード中は適用しません。

DataEventEnabledプロパティがTRUEならば、キューイングしたDataEventはアプリケーションに通知されます。コントロールはこのイベントを発生させる直前に、データをプロパティにコピーし、さらなるデータイベントを発生させないために、DataEventEnabledプロパティをFALSEに設定します。これにより、アプリケーションが現在の入力の処理や関連するプロパティの処理をしている間、コントロールはその後の入力データをキューイングするようになります。アプリケーションが現在の入力データの処理を終え、次のデータ処理の準備ができたとき、DataEventEnabledプロパティをTRUEに設定することにより、再びデータイベントが通知されるようになります。

入力データの読込み中または処理中に、コントロールにおいてエラーが発生した場合、ErrorEventが発行され、DataEventEnabledプロパティがTRUEのならば、アプリケーションに通知されます。

DataCountプロパティを読むことでコントロールによってキューイングされたデータイベント数が得られます。

コントロールにキューイングされたすべての入力データはClearInputメソッドを呼ぶことにより削除できます。

DataEventもしくはErrorEvent発生により入力された全てのデータプロパティは、ClearInputPropertiesメソッドを呼ぶ事によりデフォルト値に戻す事が出来ます。

バーコードのデコードも行うイメージスキャナは“ハイドラデバイス”として実装します。サービスはスキャナデバイスとイメージスキャナデバイスの両方をサポートします。

イメージデータからバーコードデータが出来る場合、スキャナサービスオブジェクトでDataEventがキューイングされます。

スキャンされたデータは、**ScanData**プロパティに格納されます。アプリケーションが**DecodeData**プロパティにTRUEを設定したならば、データは**ScanDataLabel**と**ScanDataType**にデコードされます。

デバイスの共有

イメージスキャナは下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、デバイスの読み込みを始める前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の必要条件については、「一覧」の表を参照してください。

プロパティ

AimModeプロパティ　R/W

形式　BOOL AimMode;

説明　TRUEならば、イメージスキャナはスキャンセッション中に照準のスポットやグリッドを表示します。FALSEならば表示しません。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	デバイスは**AimMode**の変更をサポートしていません。

参照　CapAimプロパティ

BitsPerPixel プロパティ

形式　LONG BitsPerPixel;

説明　イメージデータの1ピクセルに使われているBit数が設定されています。

本プロパティは、**DataEvent**が通知される直前にコントロールによって設定されます。

参照　「**入力モデルについて**」

CapAim プロパティ

形式　BOOL CapAim;

説明　TRUEならば、イメージスキャナは照準のスポットやグリッドの表示／非表示をサポートします。

このプロパティは、**Open**メソッドにより初期化されます。

CapHostTriggered プロパティ

形式 BOOL CapHostTriggered;

説明 TRUEならばイメージスキャナはStartSessionとStopSessionメソッドの呼び出しをサポートしています。

このプロパティは、Openメソッドにより初期化されます。

CapDecodeData プロパティ

形式 BOOL CapDecodeData;

説明 TRUEならばイメージスキャナはバーコードデータを読むことができます。読み取れたバーコードはスキャナサービスに送ります。

このプロパティは、Openメソッドにより初期化されます。

CapIlluminate プロパティ

形式 BOOL CapIlluminate;

説明 TRUEならば、イメージスキャナはイルミネーションの表示／非表示をサポートします。

このプロパティは、Openメソッドにより初期化されます。

CapImageData プロパティ

形式 BOOL CapImageData;

説明 TRUEならばイメージスキャナは静止画読み取りモードをサポートします。

このプロパティは、Openメソッドにより初期化されます。

CapImageQuality プロパティ

形式　BOOL CapImageQuality;

説明　TRUEならばイメージスキャナはImageQualityプロパティをサポートし、アプリケーションがイメージサイズを小さくするためにイメージの圧縮または読み取り解像度の制御に使用することができます。

このプロパティは、Openメソッドにより初期化されます。

参照　ImageQualityプロパティ

CapVideoData プロパティ

形式　BOOL CapVideoData;

説明　TRUEならばイメージスキャナは低解像度ビデオストリームモードをサポートします。

このプロパティは、Openメソッドにより初期化されます。

FrameData プロパティ

形式　　BSTR FrameData;

説明　　イメージスキャナから読み込まれた1つのイメージデータまたは1つ以上のビデオデータが設定されます。
このデータのフォーマットはBinaryConversionプロパティの値に依存します。詳細はBinaryConversionプロパティを参照してください。

イメージデータは、一般にイメージスキャナから通知された形式です。通知されたイメージの属性は、BitsPerPixel, ImageHeight, ImageWidth, ImageTypeとImageLengthプロパティに格納されます。

ビデオデータは、一般に1つのフレームはデータ圧縮無しの1つ以上の静止画が連結されて一緒になったものです。このビデオデータは典型的にはオペレータがイメージスキャナの照準として扱える“ビューファインダー“の（照準パターンを除いた）表示に使用します。各ブロックはだいたいVideoCountプロパティで指定された数のフレームを含んでいます。DataEventはビデオデータのブロックが送られる毎に通知されます。ビデオデータの多数のブロックは、サービスオブジェクトからVideoRateプロパティに設定された1秒あたりの最大フレーム数の割合で定期的に送られます。ビデオデータのブロックを作る各静止画の属性は、BitsPerPixel, ImageHeight, ImageWidth, ImageTypeとImageLengthプロパティに格納されます。

静止画でもビデオストリームでも、イメージデータはStartSessionメソッドで開始されるか、またはイメージスキャナで非同期的に開始されるスキャンセッションで取り込まれます。FrameTypeプロパティはFrameDataプロパティに１つの静止画とビデオデータのブロックのどちらが含まれているかを示します。

本プロパティは、DataEventが通知される直前にコントロールによって設定されます。

参照　　「入力モデルについて」、BitsPerPixelプロパティ、FrameTypeプロパティ、ImageHeightプロパティ、ImageLengthプロパティ、ImageTypeプロパティ、ImageWidthプロパティ、VideoCountプロパティ、VideoRateプロパティ

FrameTypeプロパティ

形式　　LONG FrameType；

説明　　**FrameData**プロパティの内容が設定されます。

値	意味
IMG_FRAME_STILL	**FrameData**プロパティは1つの静止画です。
IMG_FRAME_VIDEO	**FrameData**プロパティはビデオストリームフレームのブロックです（連結された1つ以上のデータ圧縮無し静止画）。

本プロパティは、**DataEvent**が通知される直前にコントロールによって設定されます。

参照　　**FrameData**プロパティ

IlluminateModeプロパティ　R/W

形式　　BOOL IlluminateMode;

説明　　TRUEならば、イメージスキャナはスキャンセッション中にイルミネーションを有効にします。FALSEならば無効にします。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値　　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	デバイスは**IlluminateMode**の変更をサポートしていません。

参照　　**CapIlluminate**プロパティ

ImageHeight プロパティ

形式　　LONG ImageHeight;

説明　　取り込んだイメージデータの高さのピクセル数が設定されています。

本プロパティは、**DataEvent**が通知される直前にコントロールによって設定されます。

参照　　「**入力モデルについて**」

ImageLength プロパティ

形式　　LONG ImageLength;

説明　　取り込んだイメージデータの長さのバイト数が設定されています。

本プロパティは、**DataEvent**が通知される直前にコントロールによって設定されます。

参照　　「**入力モデルについて**」

ImageModeプロパティ　R/W

形式　　LONG ImageMode;

説明　　イメージスキャナの操作モードを設定します。

本プロパティは、**DataEvent**で通知する**FrameData**プロパティに格納されるデータのタイプを示します。

値	意味
IMG_DECODE_ONLY	イメージスキャナは静止画やビデオをアプリケーションに送信せず、取り込んだイメージからデコードされたバーコードデータをスキャナサービスを通じて送信します。ハイドラ実装されたイメージスキャナとスキャナでは、このモードが必要です。
IMG_STILL_ONLY	イメージスキャナは静止画を送信します。バーコードデータの読み取りやビデオの送信は行いません。
IMG_STILL_DECODE	イメージスキャナは静止画の送信とバーコードデータの読み取りを行います。ビデオの送信は行いません。ハイドラ実装されたイメージスキャナとスキャナでは、このモードが必要です。
IMG_VIDEO_DECODE	イメージスキャナはビデオストリームの送信とバーコードデータの読み取りを行います。ハイドラ実装されたイメージスキャナとスキャナでは、このモードが必要です。
IMG_VIDEO_STILL	イメージスキャナはビデオストリームと静止画を送信します。バーコードデータの読み取りは行いません。ビデオデータのイメージ解像度と静止画の解像度は違う場合があります。
IMG_ALL	イメージスキャナはビデオストリームの送信とバーコードデータの読み取りを行います。バーコードデータが読み取れた場合は、バーコードを静止画と同様に送信します。ビデオデータのイメージ解像度と静止画の解像度は違う場合があります。ハイドラ実装されたイメージスキャナとスキャナでは、このモードが必要です。

このプロパティは、**Open**メソッドによりIMG_STILL_ONLYに初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	指定された値は**CapImageData, CapVideoData, CapDecodeData**の各プロパティであらわされる能力ではサポートしていません。

参照　**「入力モデルについて」、CapDecodeDataプロパティ、CapImageDataプロパティ、CapVideoDataプロパティ、FrameDataプロパティ、StartSessionメソッド**

ImageQualityプロパティ　R/W

形式　**LONG ImageQuality;**

説明　アプリケーションが必要とするイメージの解像度を設定します。

値	意味
IMG_QUAL_LOW	イメージデータは低解像度。
IMG_QUAL_MED	イメージデータは中解像度。
IMG_QUAL_HIGH	イメージデータは高解像度。

このプロパティは、**Open**メソッドによりIMG_QUAL_HIGHに初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

参照　**CapImageQualityプロパティ**

ImageTypeプロパティ

形式　LONG ImageType；

説明　FrameDataプロパティに格納されているイメージデータのフォーマットが設定されます。

値	意味
IMG_TYP_BMP	イメージデータはBMPフォーマットです。
IMG_TYP_JPEG	イメージデータはJPEGフォーマットです。
IMG_TYP_GIF	イメージデータはGIFフォーマットです。
IMG_TYP_PNG	イメージデータはPNGフォーマットです。
IMG_TYP_TIFF	イメージデータはTIFFフォーマットです。

本プロパティは、DataEventが通知される直前にコントロールによって設定されます。

参照　FrameDataプロパティ

ImageWidth プロパティ

形式　LONG ImageWidth;

説明　取り込んだイメージデータの幅のピクセル数が設定されています。

本プロパティは、DataEventが通知される直前にコントロールによって設定されます。

参照　「**入力モデルについて**」

VideoCount プロパティ　R/W

形式　LONG VideoCount;

説明　１つのDataEventで送信されたビデオデータに含まれるフレーム数を設定します。デフォルトの値は15です。VideoRateプロパティが30フレーム/秒に設定されている時に、この値ならば1秒間に2回のDataEventを発生させます。

指定した値がイメージスキャナの使用可能メモリより大きければ、サービスによってイメージスキャナのサポートする最大値が強制されます。

このプロパティは、Openメソッドにより初期化されます。

戻り値　本プロパティ設定時、次の値がResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

参照　「入力モデルについて」、VideoRateプロパティ

VideoRate プロパティ　R/W

形式　LONG VideoRate;

説明　アプリケーションが受け取ることのできる1秒あたりのビデオフレーム数を設定します。デフォルトの値は30フレーム/秒です。

アプリケーションはDataEventが発生する数を絞り込むために、このプロパティとVideoRateプロパティを設定することができます。例えば、VideoCountとVideoRateプロパティのデフォルト値では、アプリケーションは1秒に2回のDataEventを受け取ります。

指定した値がイメージスキャナの使用可能メモリより大きければ、サービスによってイメージスキャナのサポートする最大値が強制されます。

イメージスキャナはVideoRateプロパティの指定を超えるイメージデータのフレームを破棄するでしょう。

このプロパティは、Openメソッドにより初期化されます。

戻り値　本プロパティ設定時、次の値がResultCodeプロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。

OPOS_E_ILLEGAL　無効な値が指定されました。

参照　　「**入力モデルについて**」、VideoCountプロパティ

メソッド

StartSession メソッド

形式 LONG StartSession ();

説明 このメソッドは、イメージスキャナが**ImageMode**プロパティに選択されたモードでバーコードデータ、静止画、ビデオストリームデータを取得するトリガとして使用します。
セッションは**StopSession**メソッドが呼び出されるか、イメージスキャナが自分自身で終了させるまではアクティブです。セッションはイメージデータやバーコードデータを取得したり、タイムアウトが発生した場合は早期に終了することもあります。セッションを終了する基準はサービスに依存します。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に終了しました。
OPOS_E_ ILLEGAL	機能をサポートしていません。
その他	**ResultCode**を参照のこと。

参照 **CapHostTriggered**プロパティ、**ImageMode**プロパティ、**StopSession**メソッド

StartSession メソッド

形式 LONG StartSession ();

説明 このメソッドは、**StartSession**メソッドで開始されたセッションを終了させるために使います。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に終了しました。
OPOS_E_ ILLEGAL	機能をサポートしていません。
その他	**ResultCode**を参照のこと。

参照 **CapHostTriggered**プロパティ、**StartSession**メソッド

イベント

DataEvent イベント

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

*Status*パラメータはゼロが設定されます。

説明　　デバイスからアプリケーションに、入力データを送るために通知されます。このイベントが通知される前に、スキャナデータが**ScanData, ScanDataLabel, ScanDataType**プロパティに設定されます。

ErrorEvent イベント

形式　　**void ErrorEvent (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**ResultCode**の項目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**ResultCodeExtended**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_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が、すでにバッファリングされたデータの一部は利用できます。

*pErrorResponse*パラメータが示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_CLEAR	バッファリングされている入力データをクリアし、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、**DataEventEnabled**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、**DataEventEnabled**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　スキャナデータを読み込もうとしている間に、エラーを検出すると通知されます。

正しいアプリケーションシーケンシングが行われるよう、**DataEventEnabled**プロパティがTRUEになるまで、入力エラーイベントは通知されません。

参照　「**ステータス、結果コード、状態モデル**」

第 32 章
電子バリューリーダライタ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.12	Boolean	R/W	Open
BinaryConversion	1.12	Long	R/W	Open
CapCompareFirmwareVersion	1.12	Boolean	R	Open
CapPowerReporting	1.12	Long	R	Open
CapStatisticsReporting	1.12	Boolean	R	Open
CapUpdateFirmware	1.12	Boolean	R	Open
CapUpdateStatistics	1.12	Boolean	R	Open
CheckHealthText	1.12	String	R	Open
Claimed	1.12	Boolean	R	Open
DataCount	1.12	Long	R	Open
DataEventEnabled	1.12	Boolean	R/W	Open
DeviceEnabled	1.12	Boolean	R/W	Open & Claim
FreezeEvents	1.12	Boolean	R/W	Open
OpenResult	1.12	Long	R	無し
OutputID	1.12	Long	R	Open
PowerNotify	1.12	Long	R/W	Open
PowerState	1.12	Long	R	Open
ResultCode	1.12	Long	R	無し
ResultCodeExtended	1.12	Long	R	Open
State	1.12	Long	R	無し
ControlObjectDescription	1.12	String	R	無し
ControlObjectVersion	1.12	Long	R	無し
ServiceObjectDescription	1.12	String	R	Open
ServiceObjectVersion	1.12	Long	R	Open
DeviceDescription	1.12	String	R	Open
DeviceName	1.12	String	R	Open
専用				
CapActivateService	1.12	Boolean	R	Open

CapAdditionalSecurityInformation	1.15	Boolean	R	Open
CapAddValue	1.12	Boolean	R	Open
CapAuthorizeCompletion	1.15	Boolean	R	Open
CapAuthorizePreSales	1.15	Boolean	R	Open
CapAuthorizeRefund	1.15	Boolean	R	Open
CapAuthorizeVoid	1.15	Boolean	R	Open
CapAuthorizeVoidPreSales	1.15	Boolean	R	Open
CapCancelValue	1.12	Boolean	R	Open
CapCardSensor	1.12	LONG	R	Open
CapCashDeposit	1.15	Boolean	R	Open
CapCenterResultCode	1.15	Boolean	R	Open
CapCheckCard	1.15	Boolean	R	Open
CapDailyLog	1.15	Long	R	Open
CapDetectionControl	1.12	LONG	R	Open
CapElectronicMoney	1.12	Boolean	R	Open
CapEnumerateCardServices	1.12	Boolean	R	Open
CapIndirectTransactionLog	1.12	Boolean	R	Open
CapInstallments	1.15	Boolean	R	Open
CapLockTerminal	1.12	Boolean	R	Open
CapLogStatus	1.12	Boolean	R	Open
CapMediumID	1.12	Boolean	R	Open
CapMembershipCertificate	1.14.1	Boolean	R	Open
CapPaymentDetail	1.15	Boolean	R	Open
CapPINDevice	1.14	Boolean	R	Open
CapPoint	1.12	Boolean	R	Open
CapSubtractValue	1.12	Boolean	R	Open
CapTaxOthers	1.15	Boolean	R	Open
CapTrainingMode	1.14	Boolean	R	Open
CapTransaction	1.12	Boolean	R	Open
CapTransactionLog	1.12	Boolean	R	Open
CapTransactionNumber	1.15	Boolean	R	Open
CapUnlockTerminal	1.12	Boolean	R	Open
CapUpdateKey	1.12	Boolean	R	Open
CapVoucher	1.12	Boolean	R	Open
CapWriteValue	1.12	Boolean	R	Open
AccountNumber	1.14	String	R	Open
AdditionalSecurityInformation	1.12	String	R/W	Open
Amount	1.14	Currency	R/W	Open

ApprovalCode	1.12	String	R/W	Open
AsyncMode	1.12	Boolean	R/W	Open
Balance	1.14	Currency	R	Open
BalanceOfPoint	1.12	Currency	R	Open
CardCompanyID	1.15	String	R	Open
CardServiceList	1.12	String	R	Open
CenterResultCode	1.15	String	R	Open
CurrentService	1.12	String	R/W	Open
DailyLog	1.15	String	R	Open
DetectionControl	1.12	Boolean	R/W	Open
DetectionStatus	1.12	Long	R	Open
ExpirationDate	1.12	String	R	Open
LastUsedDate	1.12	String	R	Open
LogStatus	1.12	Long	R	Open
MediumID	1.14	String	R/W	Open
PaymentCondition	1.15	Long	R	Open
PaymentDetail	1.15	String	R	Open
PaymentMedia	1.15	Long	R/W	Open
PINEntry	1.14	Long	R/W	Open
Point	1.14	Currecny	R/W	Open
ReaderWriterServiceList	1.12	String	R	Open
SequenceNumber	1.12	Long	R	Open
ServiceType	1.14.1	Long	R	Open
SettledAmount	1.14	Currency	R	Open
SettledPoint	1.12	Currency	R	Open
SlipNumber	1.15	String	R	Open
TrainingModeState	1.14	Long	R/W	Open
TransactionNumber	1.15	String	R	Open
TransactionLog	1.12	String	R	Open
TransactionType	1.15	Long	R	Open
VoucherID	1.12	String	R/W	Open
VoucherIDList	1.12	String	R/W	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.12	無し
Close	1.12	Open
ClaimDevice	1.12	Open
ReleaseDevice	1.12	Open & Claim
CheckHealth	1.12	Open, Claim, & Enable
ClearInput	1.12	Open & Claim
ClearInputProperties	1.12	Open & Claim
ClearOutput	1.12	Open & Claim
CompareFirmwareVersion	1.12	Open, Claim, & Enable
DirectIO	1.12	Open & Claim
ResetStatistics	1.12	Open, Claim, & Enable
RetrieveStatistics	1.12	Open, Claim, & Enable
UpdateFirmware	1.12	Open, Claim, & Enable
UpdateStatistics	1.12	Open, Claim, & Enable
専用		
AccessDailyLog	1.15	Open, Claim, & Enable
AccessData	1.14.1	Open, Claim, & Enable
AccessLog	1.12	Open, Claim, & Enable
ActivateEVService	1.14.1	Open, Claim, & Enable
ActivateService	1.12	Open, Claim, & Enable
AddValue	1.12	Open, Claim, & Enable
AuthorizeCompletion	1.15	Open, Claim, & Enable
AuthorizePreSales	1.15	Open, Claim, & Enable
AuthorizeRefund	1.15	Open, Claim, & Enable
AuthorizeSales	1.15	Open, Claim, & Enable
AuthorizeVoid	1.15	Open, Claim, & Enable
AuthorizeVoidPreSales	1.15	Open, Claim, & Enable
BeginDetection	1.12	Open, Claim, & Enable
BeginRemoval	1.12	Open, Claim, & Enable
CancelValue	1.12	Open, Claim, & Enable
CaptureCard	1.12	Open, Claim, & Enable
CashDeposit	1.15	Open, Claim, & Enable
CheckCard	1.15	Open, Claim, & Enable
CheckServiceRegistrationToMedium	1.14.1	Open, Claim, & Enable
ClearParameterInformation	1.14	Open, Claim, & Enable
CloseDailyEVService	1.14.1	Open, Claim, & Enable

DeactivateEVService	1.14.1	Open, Claim, & Enable
EndDetection	1.12	Open, Claim, & Enable
EndRemoval	1.12	Open, Claim, & Enable
EnumerateCardServices	1.12	Open, Claim, & Enable
LockTerminal	1.12	Open, Claim, & Enable
OpenDailyEVService	1.14.1	Open, Claim, & Enable
QueryLastSuccessfulTransactionResult	1.14	Open, Claim, & Enable
ReadValue	1.12	Open, Claim, & Enable
RegisterServiceToMedium	1.14.1	Open, Claim, & Enable
RetrieveResultInformation	1.15	Open, Claim, & Enable
SetParameterInformation	1.14	Open, Claim, & Enable
SubtractValue	1.12	Open, Claim, & Enable
TransactionAccess	1.12	Open, Claim, & Enable
UnlockTermina;	1.12	Open, Claim, & Enable
UnregisterServiceToMedium	1.14.1	Open, Claim, & Enable
UpdateData	1.14.1	Open, Claim, & Enable
UpdateKey	1.12	Open, Claim, & Enable
WriteValue	1.12	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.12	Open, Claim, & Enable
DirectIOEvent	1.12	Open & Claim
ErrorEvent	1.12	Open, Claim, & Enable
OutputCompleteEvent	1.12	Open, Claim, & Enable
StatusUpdateEvent	1.12	Open, Claim, & Enable
TransitionEvent	1.15	Open, Claim, & Enable

概説

電子バリューリーダライタコントロールのOLEプログラマティクIDは、“OPOS.EVRW”です。

非接触・接触ICカード（以降、カードと呼びます）に格納された電子マネー、ポイントそしてクーポン／チケットといった、そのカードに保持されるさまざまな権利を電子バリューと定義します。電子バリューリーダライタとは、電子バリューを利用した決済（加算、減算、設定、読み込み）を行うための機能を提供するデバイスです。

電子マネーサービスは、ポストペイ型電子マネー決済、プリペイド型電子マネー決済、クレジット決済、デビット決済を想定します。

ポイントサービスは、カード内にポイントを保持しそのポイントと交換で対価を受け取ることができ、サービス提供者の都合で任意にポイントの加算を行うことができるサービスを想定しています。実装により、ポイントはカード内に保持されず、カードに保持された識別子により、別の場所に格納されたポイントと紐付けがされている場合もあります。

クーポン／チケットサービスは、カード内に権利を証明する識別子を複数個保持しその権利と交換で対価を受け取ることができ、サービス提供者の都合で任意に権利の追加を行うことができるサービスを想定しています。

１．１４版の変更内容

1.13版までの電子バリューリーダライタ仕様では、決済要求をする際にデバイスに指定する決済情報の構文と意味づけが、デバイスベンダー毎、決済事業者毎に自由であったため、**AdditionalSecurityInformation**プロパティに独自の定義を行って使用してきました。

1.14版では、決済要求をする際にデバイスに指定する情報の構文と意味づけの定義を行い、電子バリューリーダライタデバイス実装のポータビリティを高めることを目的としています。

1.14版では以下の機能が追加されています。

トレーニングモードのプロパティ
PIN入力の有無を示すプロパティ
決済の要求と結果の情報を設定／取得するメソッド
直前取引の確認を行うメソッド
非同期で実行している決済処理の間に発生した事象を通知するイベント

１．１４．１版の変更内容

1.14版に対して、以下の機能追加が行われています。

電子バリューリーダライタデバイスの保守・運用の機能を追加し、以下のメソッドを追加しました
AccessDataメソッド
ActivateEVServiceメソッド
CloseDailyEVServiceメソッド
DeactivateEVServiceメソッド
OpenDailyEVServiceメソッド
UpdateDataメソッド

電子バリューリーダライタデバイスの保守・運用の機能を追加し、以下のメソッド・イベントを変更しました
AccessLogメソッド
LockTerminalメソッド
UnlockTerminalメソッド
UpdateKeyメソッド
RetrieveResultInformationメソッドで定義されるタグ
TransitionEventイベントで定義されるイベント名

ポイントサービス、クーポン／チケットサービス、会員証サービスの機能を追加し、以下のメソッド・プロパティを追加しました
ServiceTypeプロパティ
CheckServiceRegistrationToMediumメソッド
RegisterServiceToMediumメソッド
UnregisterServiceToMediumメソッド

サブサービスのライフサイクルを明確にしました

サービスが店舗や場所により振る舞いを変えることができるバリエーションの概念を追加し、以下プロパティを変更しました
CardServiceListプロパティ
CurrentServiceプロパティ
ReaderWriterServiceListプロパティ

電子バリューリーダライタデバイスと決済センタの接続形態について明確にしました

ResultCodeExtendedプロパティに設定されるエラーコードを明確にしました

１．１５版の変更内容

クレジット決済（接触ICクレジット決済、非接触ICクレジット決済、磁気クレジット決済）機能を有した電子バリューリーダライタデバイスに対応するために、電子バリューリーダライタ仕様に従来のCAT仕様を包含しました。

追加された以下のプロパティ、メソッドはCAT仕様に準じます。

Capプロパティ	**CapAdditionalSecurityInformation** **CapAuthorizeCompletion** **CapAuthorizePreSales** **CapAuthorizeRefund** **CapAuthorizeVoid** **CapAuthorizeVoidPreSales** **CapCashDeposit** **CapCenterResultCode** **CapCheckCard** **CapDailyLog** **CapInstallments** **CapPaymentDetail** **CapTaxOthers** **CapTransactionNumber**
専用プロパティ	**CardCompanyID** **CenterResultCode** **DailyLog** **LogStatus** **PaymentCondition** **PaymentDetail** **PaymentMedia** **SlipNumber** **TransactionNumber** **TransactionType**

専用メソッド	**AccessDailyLog** **AuthorizeCompletion** **AuthorizePreSales** **AuthorizeRefund** **AuthorizeSales** **AuthorizeVoid** **AuthorizeVoidPreSales** **CashDeposit** **CheckCard**

CAT仕様のTrainingModeプロパティは電子バリューリーダライタ仕様で定義されるTrainingModeStateプロパティを利用してください。

また、クレジット処理を行うために、タグの定義の拡張、TransitionEventイベントの定義の拡張が行われています。

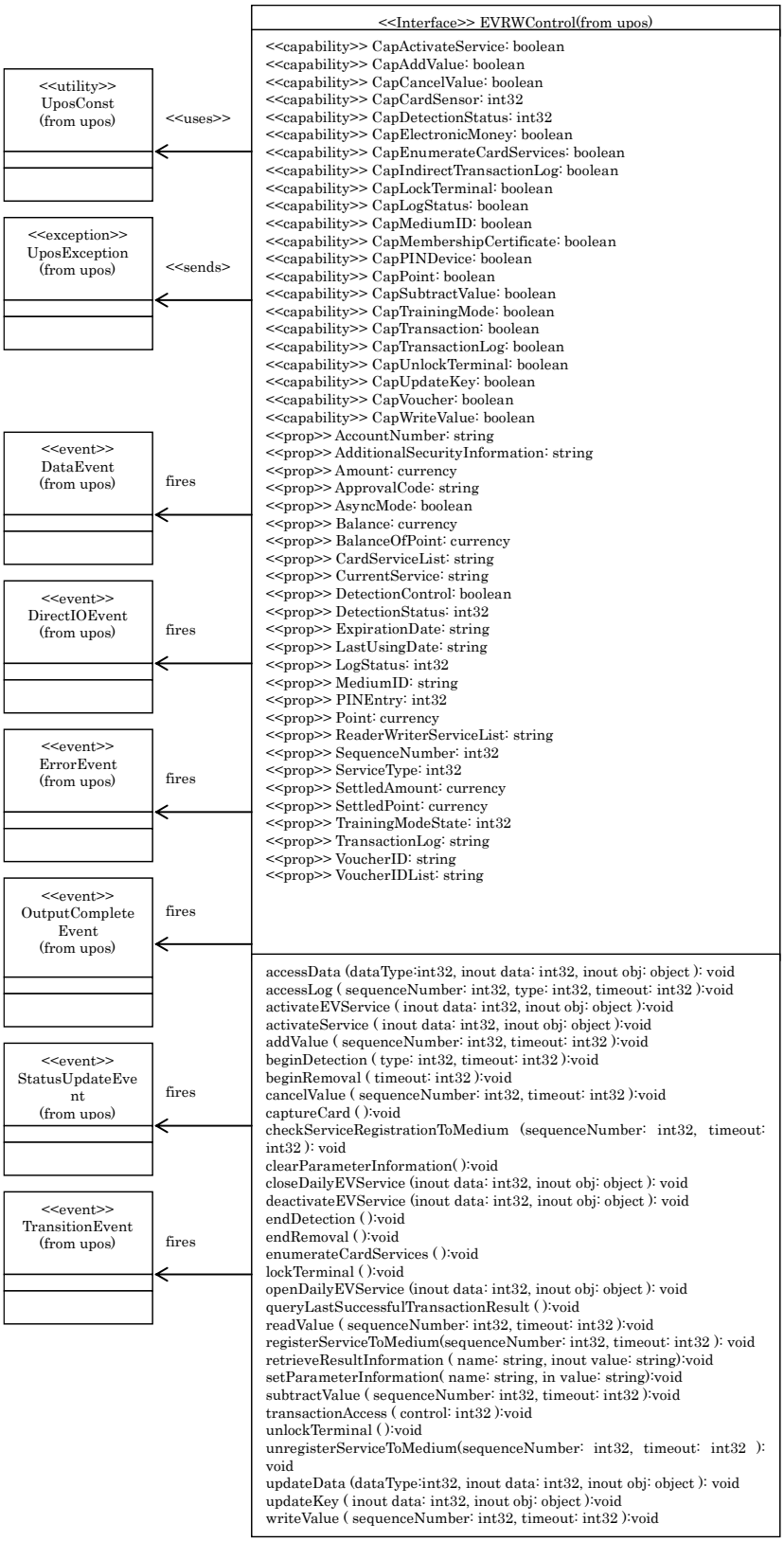
タグ	**TransactionType**タグに値の追加
TransitionEvent イベント	EVRW_TE_CONFIRM_SEARCH_TABLE EVRW_TE_CONFIRM_PAYMENT_CONDITION EVRW_TE_CONFIRM_AUTHORIZE EVRW_TE_NOTIFY_CHECK_CARD EVRW_TE_NOTIFY_ELECT_PAYMENT_CONDITION

機能

電子バリューリーダライタは、以下のような機能を持つものを想定しています。

- 決済などのためにカードにアクセスする機能
- カードから、決済に使用できる電子バリューの内容を読み取る機能
- 電子バリューを利用した決済サービスを実施する機能
- 決済の結果をログとしてデバイス内部に蓄積させる機能

クラス図


<<Interface>> EVRWControl(from upos)
<<capability>> CapActivateService: boolean
<<capability>> CapAddValue: boolean
<<capability>> CapCancelValue: boolean
<<capability>> CapCardSensor: int32
<<capability>> CapDetectionStatus: int32
<<capability>> CapElectronicMoney: boolean
<<capability>> CapEnumerateCardServices: boolean
<<capability>> CapIndirectTransactionLog: boolean
<<capability>> CapLockTerminal: boolean
<<capability>> CapLogStatus: boolean
<<capability>> CapMediumID: boolean
<<capability>> CapMembershipCertificate: boolean
<<capability>> CapPINDevice: boolean
<<capability>> CapPoint: boolean
<<capability>> CapSubtractValue: boolean
<<capability>> CapTrainingMode: boolean
<<capability>> CapTransaction: boolean
<<capability>> CapTransactionLog: boolean
<<capability>> CapUnlockTerminal: boolean
<<capability>> CapUpdateKey: boolean
<<capability>> CapVoucher: boolean
<<capability>> CapWriteValue: boolean
<<prop>> AccountNumber: string
<<prop>> AdditionalSecurityInformation: string
<<prop>> Amount: currency
<<prop>> ApprovalCode: string
<<prop>> AsyncMode: boolean
<<prop>> Balance: currency
<<prop>> BalanceOfPoint: currency
<<prop>> CardServiceList: string
<<prop>> CurrentService: string
<<prop>> DetectionControl: boolean
<<prop>> DetectionStatus: int32
<<prop>> ExpirationDate: string
<<prop>> LastUsingDate: string
<<prop>> LogStatus: int32
<<prop>> MediumID: string
<<prop>> PINEntry: int32
<<prop>> Point: currency
<<prop>> ReaderWriterServiceList: string
<<prop>> SequenceNumber: int32
<<prop>> ServiceType: int32
<<prop>> SettledAmount: currency
<<prop>> SettledPoint: currency
<<prop>> TrainingModeState: int32
<<prop>> TransactionLog: string
<<prop>> VoucherID: string
<<prop>> VoucherIDList: string
accessData (dataType:int32, inout data: int32, inout obj: object): void
accessLog (sequenceNumber: int32, type: int32, timeout: int32):void
activateEVService (inout data: int32, inout obj: object):void
activateService (inout data: int32, inout obj: object):void
addValue (sequenceNumber: int32, timeout: int32):void
beginDetection (type: int32, timeout: int32):void
beginRemoval (timeout: int32):void
cancelValue (sequenceNumber: int32, timeout: int32):void
captureCard ( ):void
checkServiceRegistrationToMedium (sequenceNumber: int32, timeout: int32): void
clearParameterInformation( ):void
closeDailyEVService (inout data: int32, inout obj: object): void
deactivateEVService (inout data: int32, inout obj: object): void
endDetection ( ):void
endRemoval ( ):void
enumerateCardServices ( ):void
lockTerminal ( ):void
openDailyEVService (inout data: int32, inout obj: object): void
queryLastSuccessfulTransactionResult ( ):void
readValue (sequenceNumber: int32, timeout: int32):void
registerServiceToMedium(sequenceNumber: int32, timeout: int32): void
retrieveResultInformation (name: string, inout value: string):void
setParameterInformation(name: string, in value: string):void
subtractValue (sequenceNumber: int32, timeout: int32):void
transactionAccess (control: int32):void
unlockTerminal ( ):void
unregisterServiceToMedium(sequenceNumber: int32, timeout: int32): void
updateData (dataType:int32, inout data: int32, inout obj: object): void
updateKey (inout data: int32, inout obj: object):void
writeValue (sequenceNumber: int32, timeout: int32):void
<<utility>>
UposConst
(from upos)
<<uses>>
<<exception>>
UposException
(from upos)
<<sends>>
<<event>>
DataEvent
(from upos)
fires
<<event>>
DirectIOEvent
(from upos)
fires
<<event>>
ErrorEvent
(from upos)
fires
<<event>>
OutputComplete
Event
(from upos)
fires
<<event>>
StatusUpdateEve
nt
(from upos)
fires
<<event>>
TransitionEvent
(from upos)
fires

モデル

電子バリューリーダライタがサポートするサービスとメソッドの対応は以下の通りです。

サービス	サービス機能	該当メソッド
共通	新規設置	**ActivateEVService**メソッド
	運用開始	**OpenDailyEVService**メソッド
	保守	**AccessData**メソッド **UpdateData**メソッド **AccessLog**メソッド **UpdateKey**メソッド
	運用終了	**CloseDailyEVService**メソッド
	取り外し	**DeactivateEVService**メソッド
電子マネー	残高照会	**ReadValue**メソッド **Balance**プロパティ
	決済	**SubtractValue**メソッド **Amount**プロパティ **SettledAmount**プロパティ
	チャージ	**AddValue**メソッド **Amount**プロパティ **SettledAmount**プロパティ
	取消	**CancelValue**メソッド **ApprovalCode**プロパティ
ポイント	媒体へのサービス登録	**RegisterServiceToMedium**メソッド **CheckServiceRegistrationToMedium**メソッド **Point**プロパティ(Input)

	媒体からのサービス削除	**UnregisterServiceToMedium**メソッド
	照会	**ReadValue**メソッド **BalanceOfPoint**タグ／プロパティ(Output)
	付与	**AddValue**メソッド **Point**タグ／プロパティ(Input) **SettledPoint**タグ／プロパティ(Output)
	減算	**SubtractValue**メソッド **Point**タグ／プロパティ(Input) **SettledPoint**タグ／プロパティ(Output)
	設定	**WriteValue**メソッド **Point**タグ／プロパティ(Input)
	取消	**CancelValue**メソッド **ApprovalCode**タグ／プロパティ(Input)
クーポン／チケット	媒体へのサービス登録	**RegisterServiceToMedium**メソッド **CheckServiceRegistrationToMedium**メソッド
	媒体からのサービス削除	**UnregisterServiceToMedium**メソッド
	照会／列挙	**ReadValue**メソッド **VoucherIDList**タグ／プロパティ(Output)
	発行	**AddValue**メソッド **VoucherID**タグ／プロパティ(Input)
	回収	**SubtractValue**メソッド **VoucherID**タグ／プロパティ(Input)

会員証	媒体へのサービス登録	**RegisterServiceToMedium**メソッド **CheckServiceRegistrationToMedium**メソッド
	媒体からのサービス削除	**UnregisterServiceToMedium**メソッド
	参照	**ReadValue**メソッド
	更新	**WriteValue**メソッド

入力モデル

ReadValueメソッドは、OPOS入力モデルに従います。

アプリケーションが電子バリューリーダライタからデータを取得する場合、**ReadValue**メソッドを呼び出します。データを取得すると、**DataEvent**イベントがキューイングされます、アプリケーションが**DataEventEnabled**プロパティをTRUEに設定すると、**DataEvent**イベントがアプリケーションに通知されます。

データを読取中にエラーが発生した場合は、**ErrorEvent**イベントが**DataEvent**イベントの代わりにキューに入れられます。アプリケーションが**DataEventEnabled**プロパティをTRUEに設定すると、**ErrorEvent**イベントがアプリケーションに通知されます。

アプリケーションは、**DataCount**プロパティを取得することにより、キューイングされたデータイベントの数を取得することができます。

AutoDisableプロパティがTRUEならば、**DataEvent**のキューイング時にコントロールは自動的にディセーブルとなります。

キューイングされたすべての入力データは、**ClearInput**メソッドを実行することで削除できます。

出力モデル

WriteValueメソッド、AddValueメソッド、SubtractValueメソッド、CancelValueメソッド、AccessLogメソッド、TransactionAccessメソッドはOPOS出力モデルに従い、AsyncModeプロパティにより、同期および非同期の切り替えが可能です。

AsyncModeプロパティがTRUEの場合でも、メソッド発行後直ちに次のメソッドを発行するような多重発行はできません。ただし**ClearOutput**メソッドは先行して発行された非同期メソッドを取消目的で実行されますので、例外です。

非同期処理の実行が完了すると、**OutputCompleteEvent**イベントがアプリケーションに通知されます。

サブサービス利用のサポート

ひとつの電子バリューリーダライタが複数の電子バリューサービスを提供し、それぞれのサービスに対応する電子バリューリーダライタサービスオブジェクトが存在する場合、それらをサブサービスとして利用することができます。

Openメソッドを実行すると、全てのサブサービスの**Open**メソッドが呼び出され、**ReaderWriterServiceList**プロパティにサブサービスが列挙されます。**Close**メソッド、**ClaimDevice**メソッド、**ReleaseDevice**メソッドも同様に全てのサブサービスに作用します。

アプリケーションは利用する任意のサービスを**CurrentService**プロパティに設定し、メソッド、プロパティの操作を行います。

一括処理（トランザクションモード）のサポート

一括処理（トランザクション）は、複数のメソッドの呼び出しで構成されます。一括処理に含むことが可能な操作は**WriteValue**メソッド、**AddValue**メソッド、**SubtractValue**メソッド、**CancelValue**メソッド、およびプロパティの設定です。一括処理中のこれらのメソッドの実行は、最初に有効性が確認されます。もし正当ならば、これらの操作は一括処理バッファに追加されますが、まだカードに対して実行はされません。アプリケーションが必要とする幾つかの操作を行った後、一括処理の実行メソッドが呼び出されます。

一括処理の実行も**AsyncMode**プロパティの影響を受けます。

もし一括処理が同期処理されるなら、戻り値は全ての一括処理の実行が成功したこと、あるいはエラーが起こったことを表します。もし一括処理が非同期的に実行されるならば、メソッドの非同期処理の規則に従います。もしエラーが発生し、アプリケーションがエラーイベントハンドラにリトライの要求するならば、全ての一括処理のメソッドはリトライされます。

トレーニングモードのサポート　　１．１４版で追加

CapTrainingModeプロパティ

TrainingModeStateプロパティ

追加情報のタグ値による設定と取得　　１．１４版で追加

CapPINDeviceプロパティ

PINEntryプロパティ

ClearParameterInformationメソッド

RetrieveResultInformationメソッド

SetParameterInformationメソッド

各メソッドが要求する追加情報を**SetPatameterInformation**メソッドを使ってタグ値で設定することができます。各メソッドの処理結果の追加情報を**RetrieveResultInformation**メソッドを使ってタグ値で受け取ることができます。

1.13版までは、サービスプロバイダ毎に内容が異なる追加情報は**AdditionalSecurityInformation**プロパティを利用して受け渡しされていました。**AdditionalSecurityInformation**プロパティに設定される内容はサービスプロバイダ毎の実装に委ねられているため、サービスプロバイダ毎に互換性がなく、標準化の妨げになっていました。

1.14版では追加情報の構文と意味を標準化することにより、サービスプロバイダ毎の互換性と汎用性を確保しています。

AdditionalSecurityInformationプロパティは従来どおり、サービスプロバイダ固有の追加情報の受け渡しに利用することができます。

非同期処理中のイベント通知　１．１４版で追加

TransitionEventイベント

メソッドの非同期処理中に、電子バリューリーダライタデバイスとPOSアプリケーションで双方向のコミュニケーションが必要な場合に、**TransitionEvent**イベントが通知されます。

このイベントは、電子バリューリーダライタデバイスでのみ適用されます。

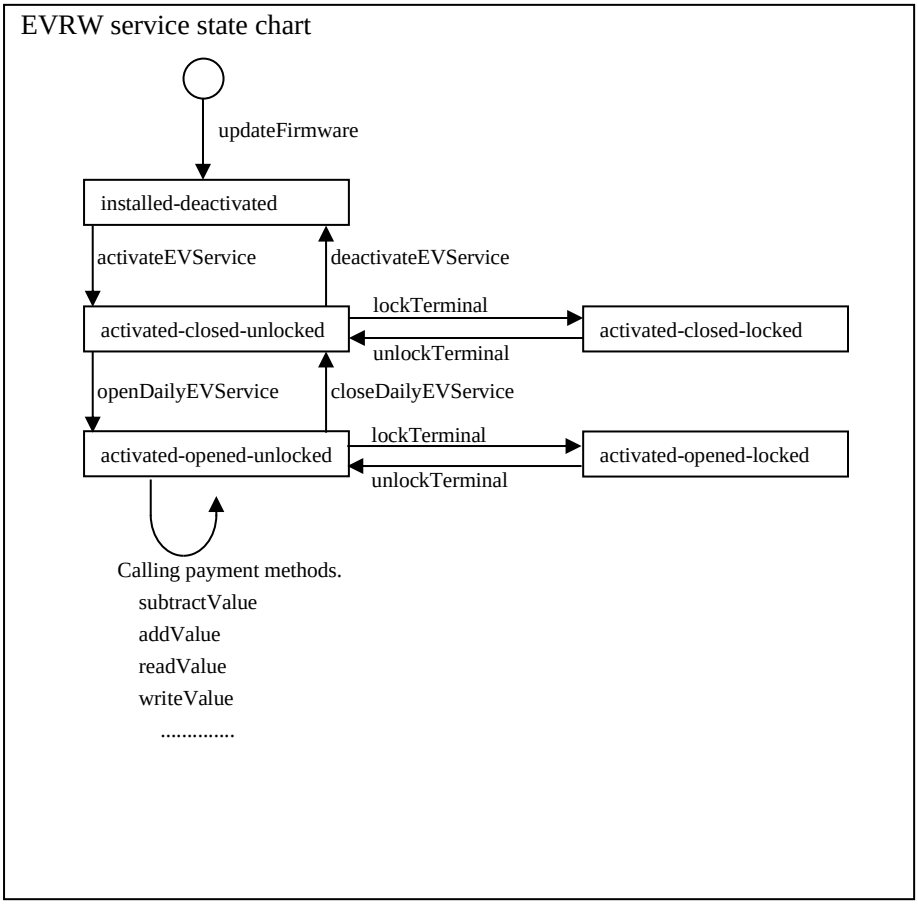
直前に成功した取引情報の取得　１．１４版で追加

QuesrLastSuccessfulTransactionResultメソッド

取引の途中で電源断などが発生して取引状態が判らなくなった場合に備えて、POSシステムが復旧後に、直前に成功した取引情報を取得することで、取引の整合性を確認するメソッドが追加されています。

サブサービスのライフサイクル　　1.14.1版で追加

サブサービスは以下のライフサイクルを持ちます。



installed-deactivated状態
UpdateFirmwareメソッドによりインストールされましたが、この時点ではアクティベートされていない状態です。

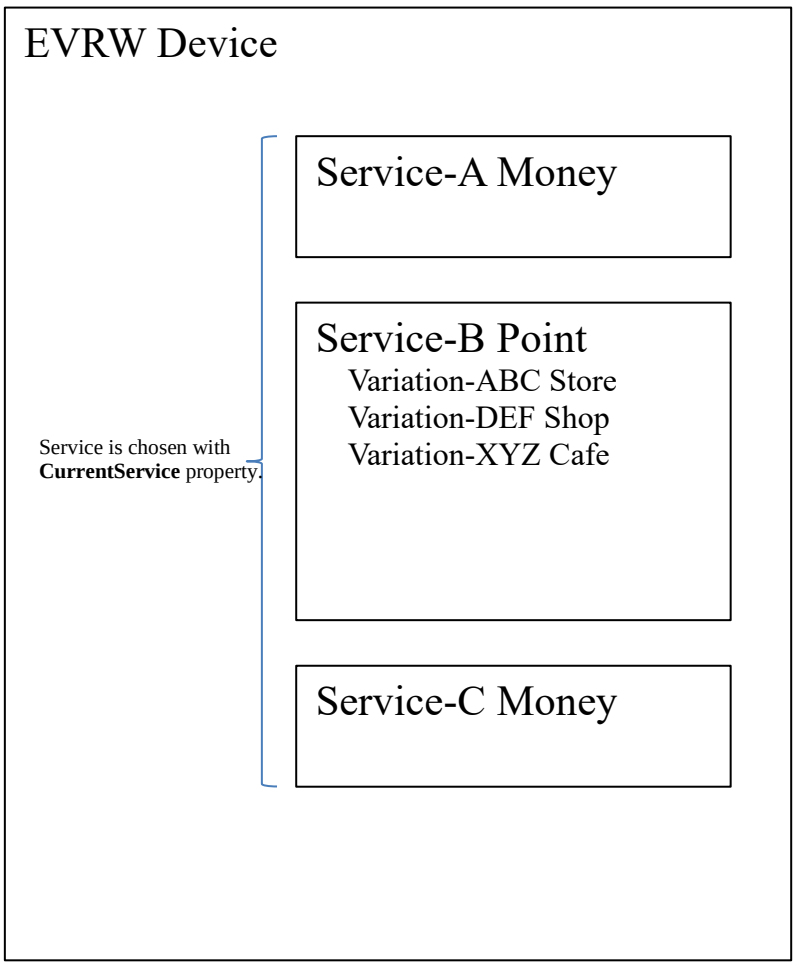
activated-closed-unlocked状態
ActivateEVServiceメソッドにより、サブサービスがアクティベートされた状態です。サービスを実行するためには、オープンする必要があります。

activated-opened-unlocked状態
OpenDailyEVServiceメソッドにより、サブサービスがオープンされ、サービスの実行が可能な状態です。

activated-closed-locked状態／activated-opened-locked状態
LockTerminalメソッドにより、ロックされた状態です。ロック状態を解除するにはUnlockTerminalメソッドを実行する必要があります。

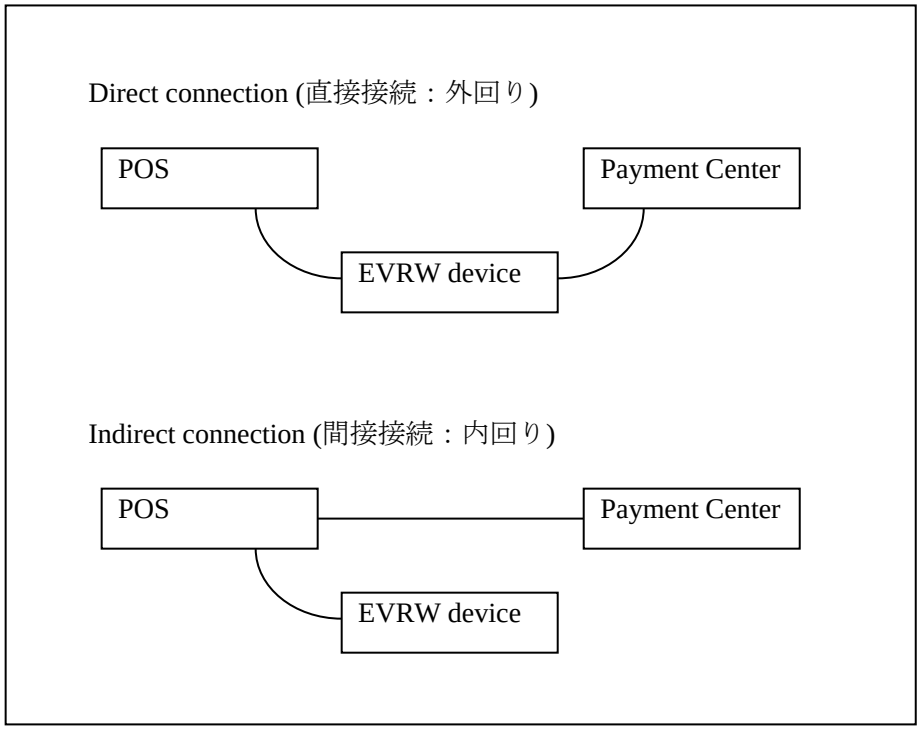
サービスがバリエーションを持つ場合　　1.14.1版で追加

サービスは、バリエーションを持つことができます。同じサービスでも、店舗や場所により振る舞いが変わる場合には、バリエーションを指定することができます。



電子バリューリーダライタデバイスと決済センタの接続形態
1.14.1版で追加

電子バリューリーダライタ仕様では、電子バリューリーダライタデバイスと決済センタの接続の方法として、電子バリューリーダライタデバイスと決済センタが直接接続され通信する「直接接続型（外回り）」と、POSシステム等を介して間接接続され通信する「間接接続型（内回り）」を想定しています。



デバイスの共有

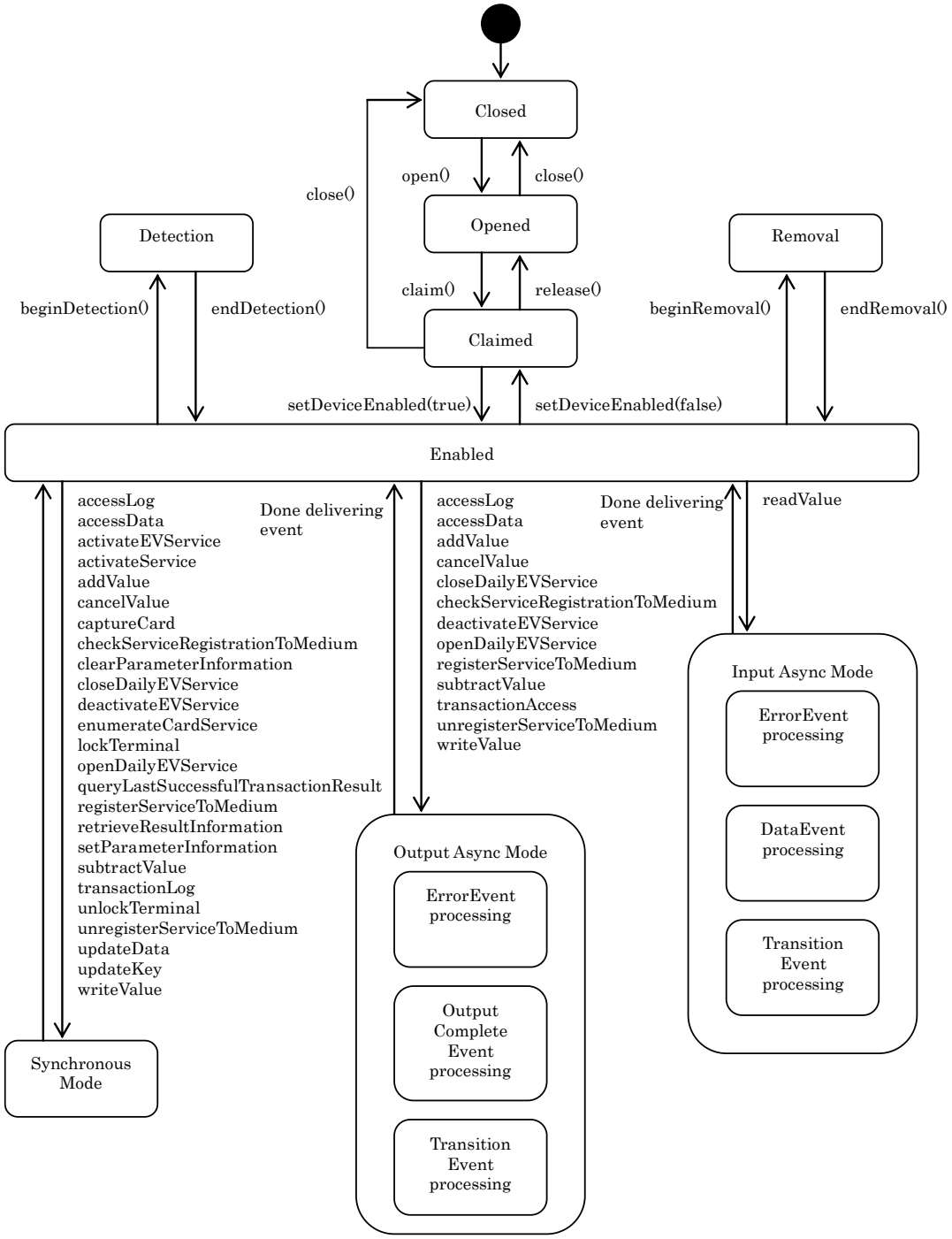
電子バリューリーダライタは、以下に示すような排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に、排他アクセス権を獲得しなければなりません。

アプリケーションは、プロパティを参照、あるいは、デバイスを更新するメソッドを呼び出す前には、デバイスの排他アクセス権を獲得し、イネーブルにしなければなりません。

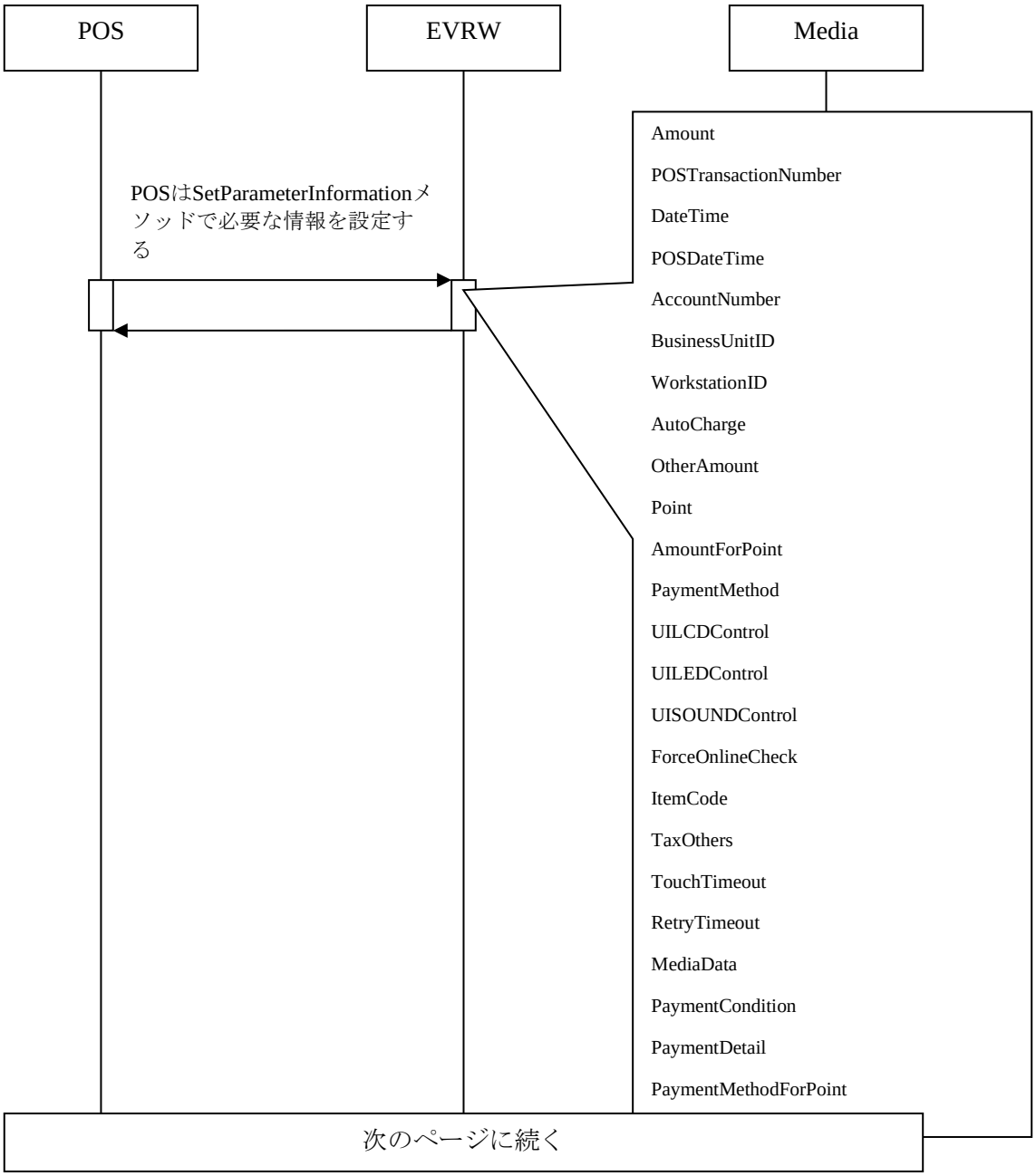
使用上の必要条件に関しては、「一覧」を参照してください。

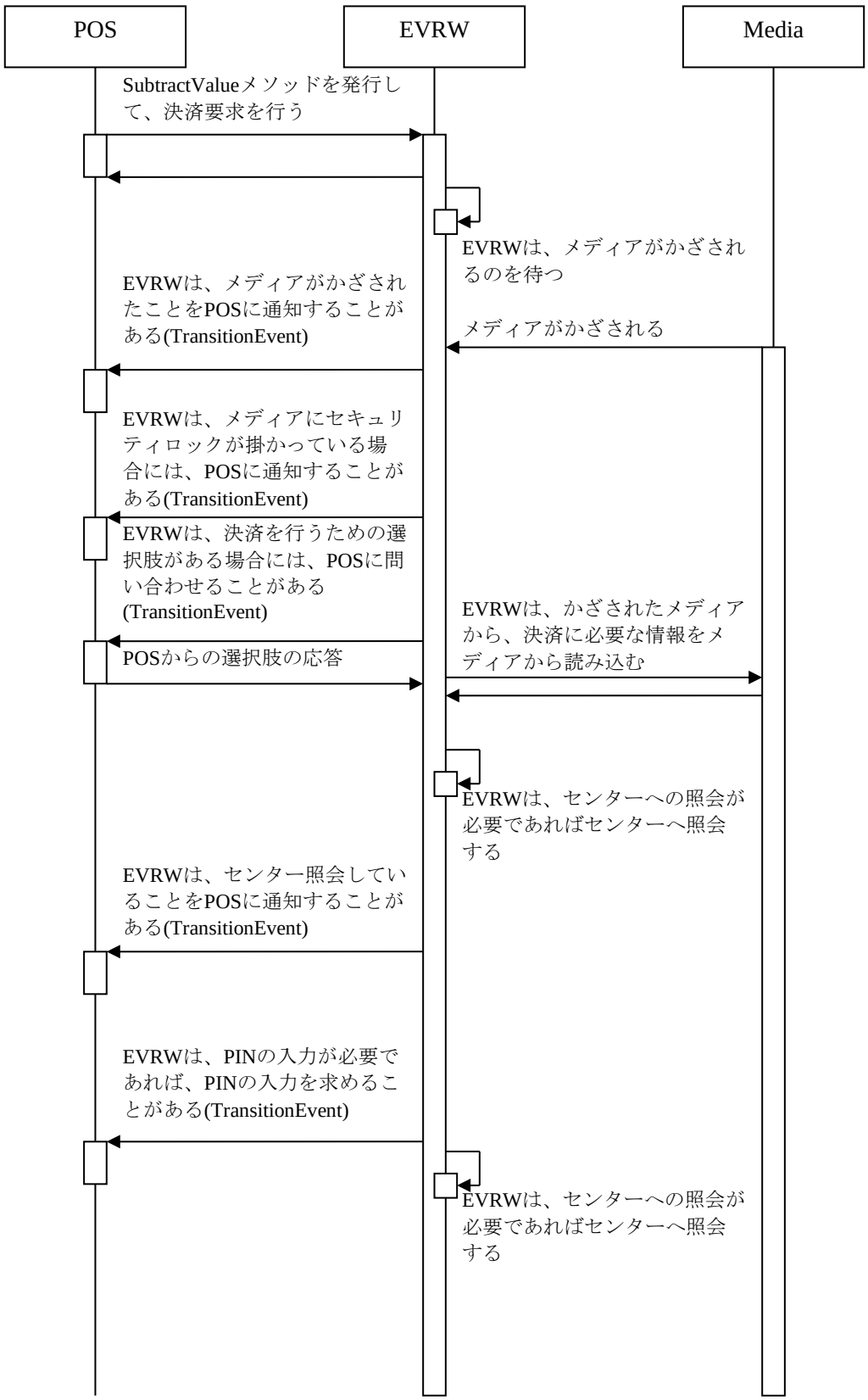
ステート図


Closed
open()
close()
close()
Opened
Detection
Removal
claim()
release()
beginDetection()
endDetection()
beginRemoval()
endRemoval()
Claimed
setDeviceEnabled(true)
setDeviceEnabled(false)
Enabled
accessLog
accessData
activateEVService
activateService
addValue
cancelValue
captureCard
checkServiceRegistrationToMedium
clearParameterInformation
closeDailyEVService
deactivateEVService
enumerateCardService
lockTerminal
openDailyEVService
queryLastSuccessfulTransactionResult
registerServiceToMedium
retrieveResultInformation
setParameterInformation
subtractValue
transactionLog
unlockTerminal
unregisterServiceToMedium
updateData
updateKey
writeValue
Done delivering event
accessLog
accessData
addValue
cancelValue
closeDailyEVService
checkServiceRegistrationToMedium
deactivateEVService
openDailyEVService
registerServiceToMedium
subtractValue
transactionAccess
unregisterServiceToMedium
writeValue
Done delivering event
readValue
Synchronous Mode
Output Async Mode
ErrorEvent processing
Output Complete Event processing
Transition Event processing
Input Async Mode
ErrorEvent processing
DataEvent processing
Transition Event processing

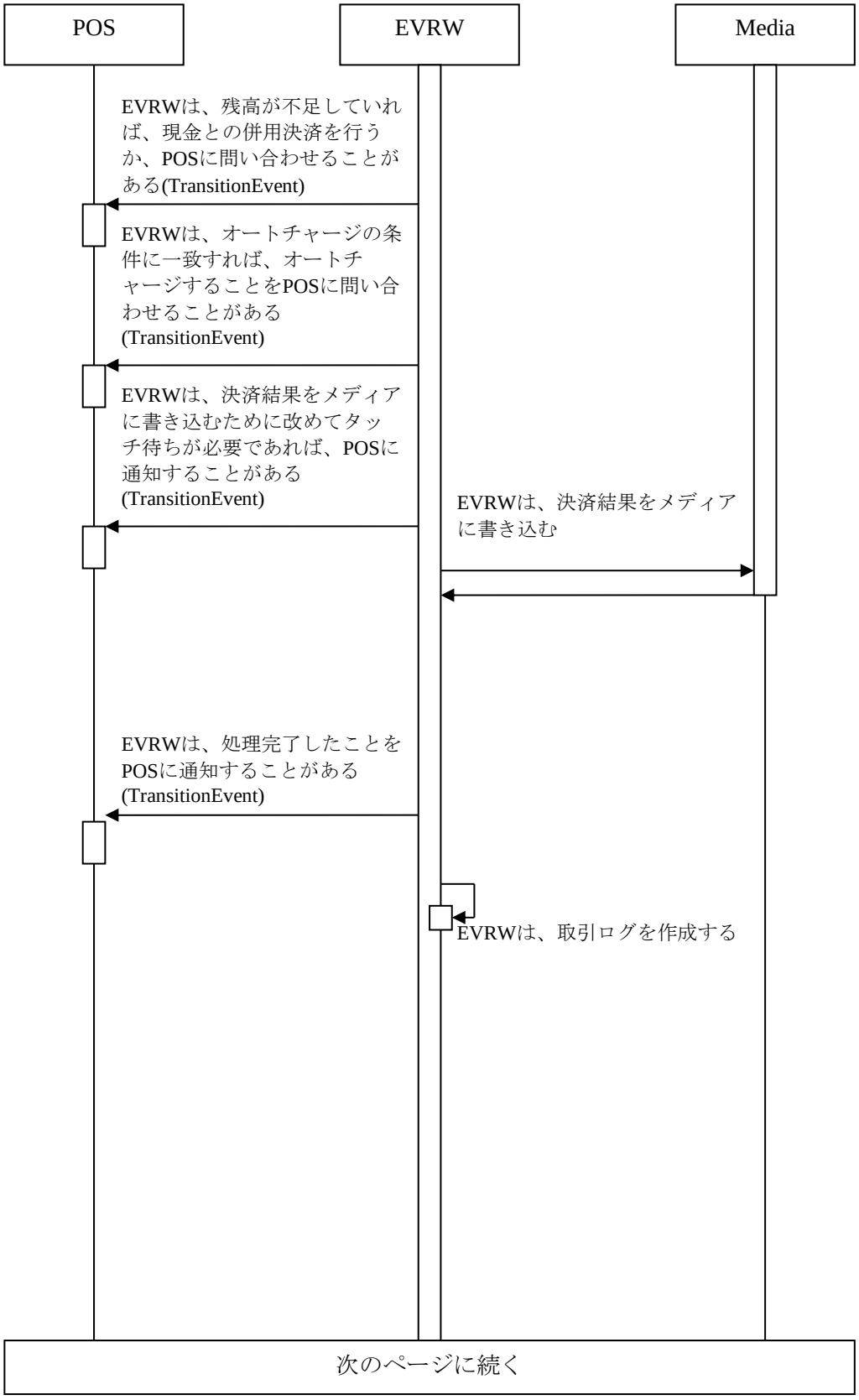
シーケンス図

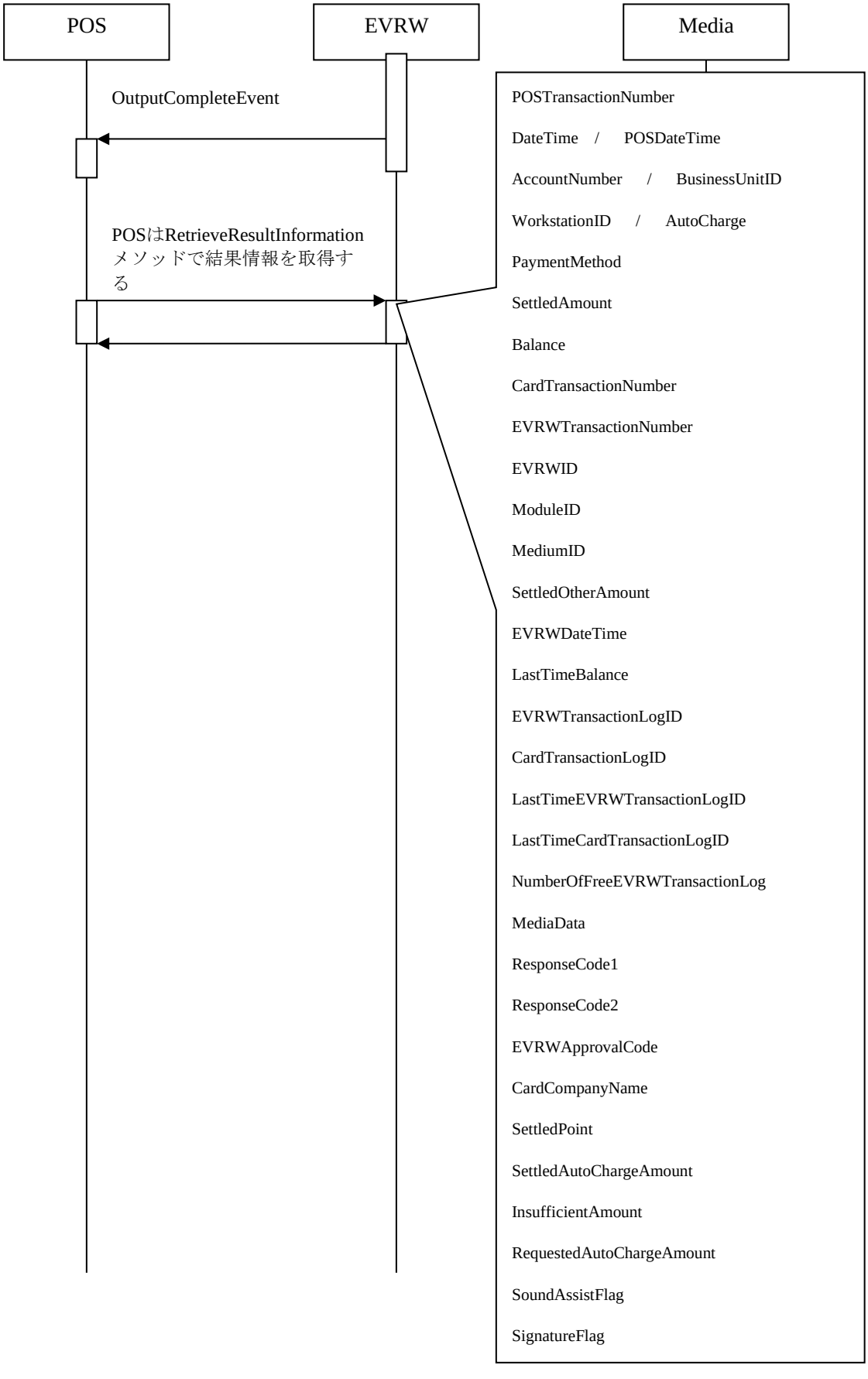
SubtractValueメソッドによる決済処理



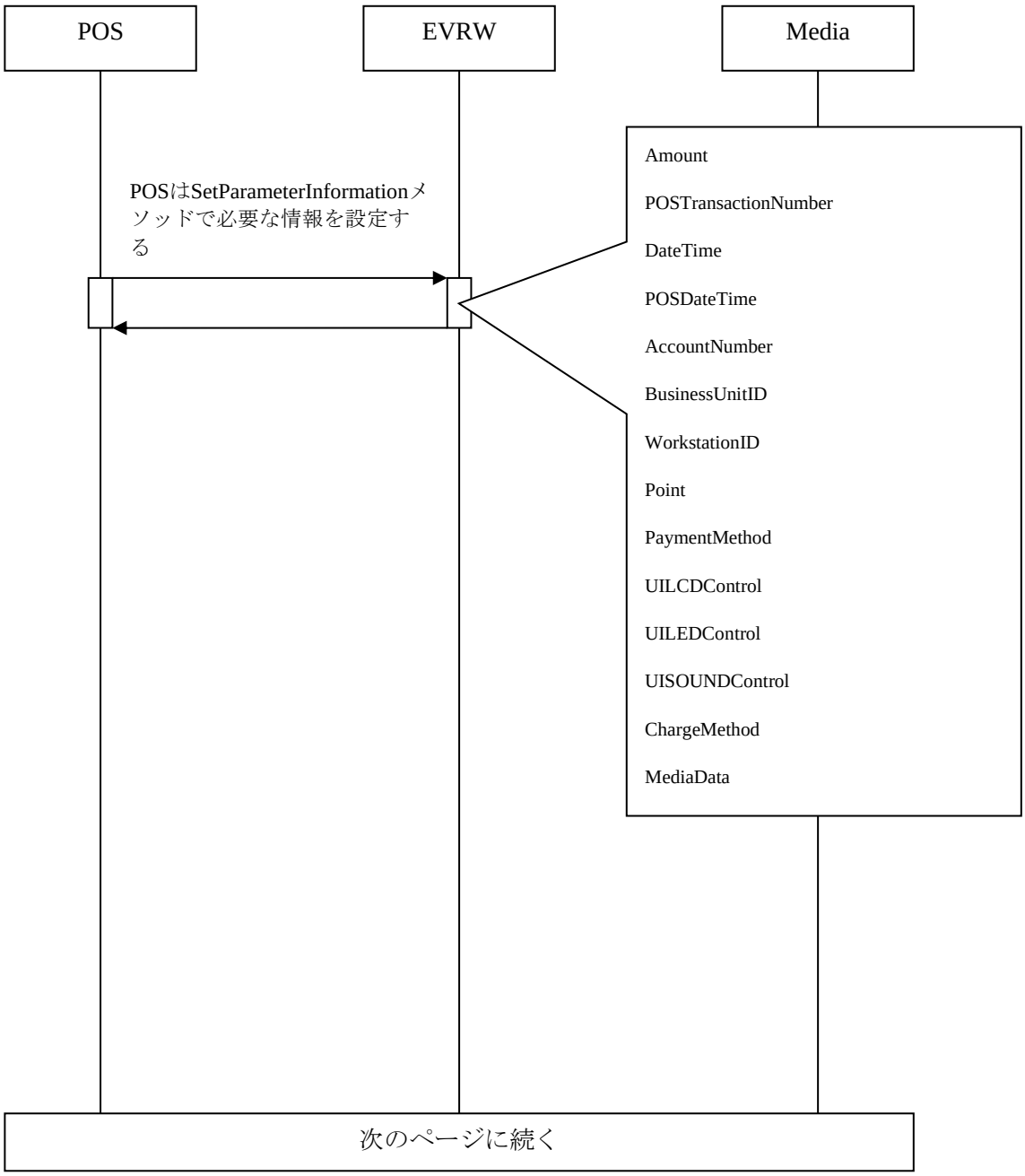

POS
EVRW
Media
SubtractValueメソッドを発行して、決済要求を行う
EVRWは、メディアがかざされるのを待つ
EVRWは、メディアがかざされたことをPOSに通知することがある(TransitionEvent)
メディアがかざされる
EVRWは、メディアにセキュリティロックが掛かっている場合には、POSに通知することがある(TransitionEvent)
EVRWは、決済を行うための選択肢がある場合には、POSに問い合わせることがある(TransitionEvent)
EVRWは、かざされたメディアから、決済に必要な情報をメディアから読み込む
POSからの選択肢の応答
EVRWは、センターへの照会が必要であればセンターへ照会する
EVRWは、センター照会していることをPOSに通知することがある(TransitionEvent)
EVRWは、PINの入力が必要であれば、PINの入力を求めることがある(TransitionEvent)
EVRWは、センターへの照会が必要であればセンターへ照会する

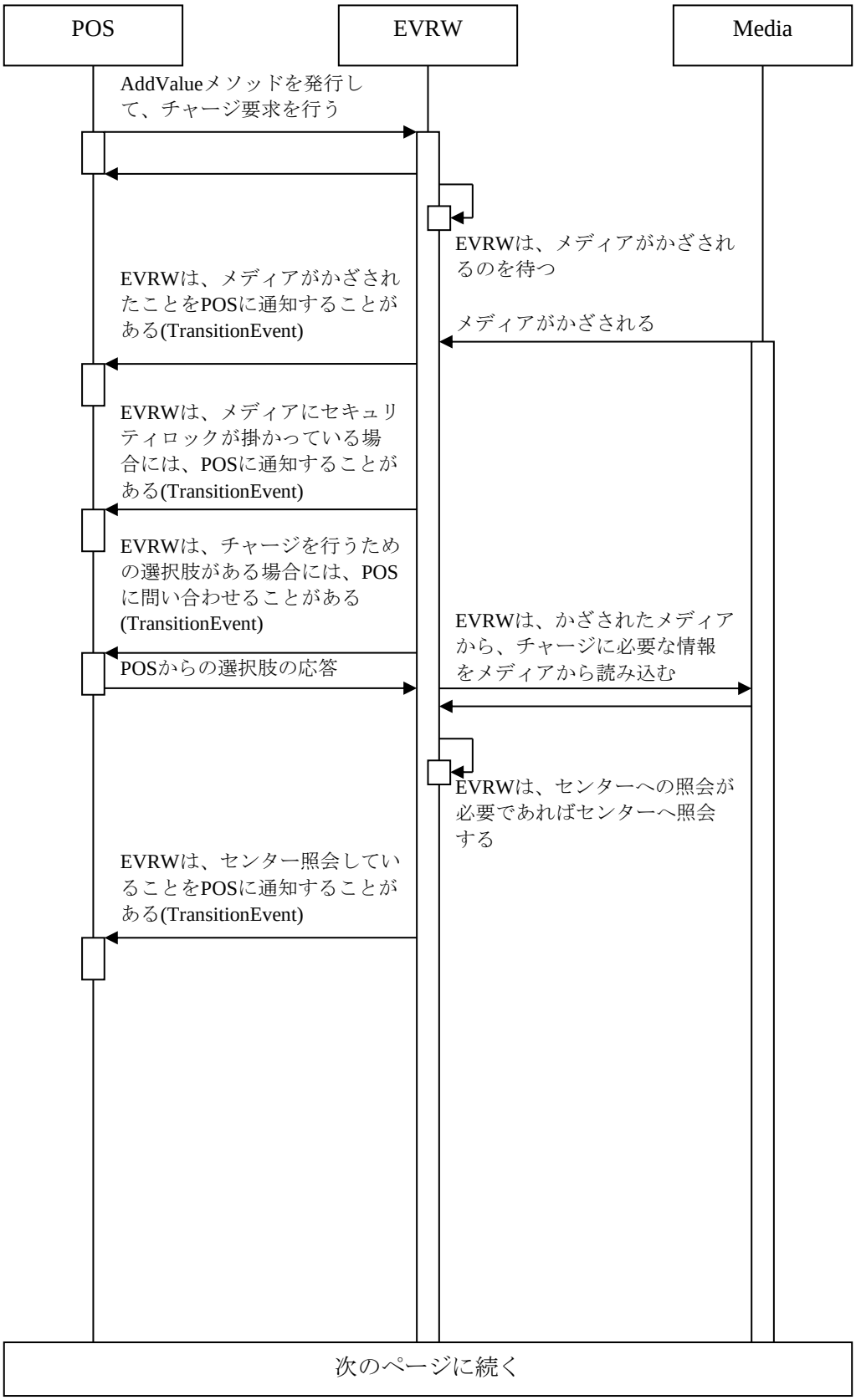
次のページに続く

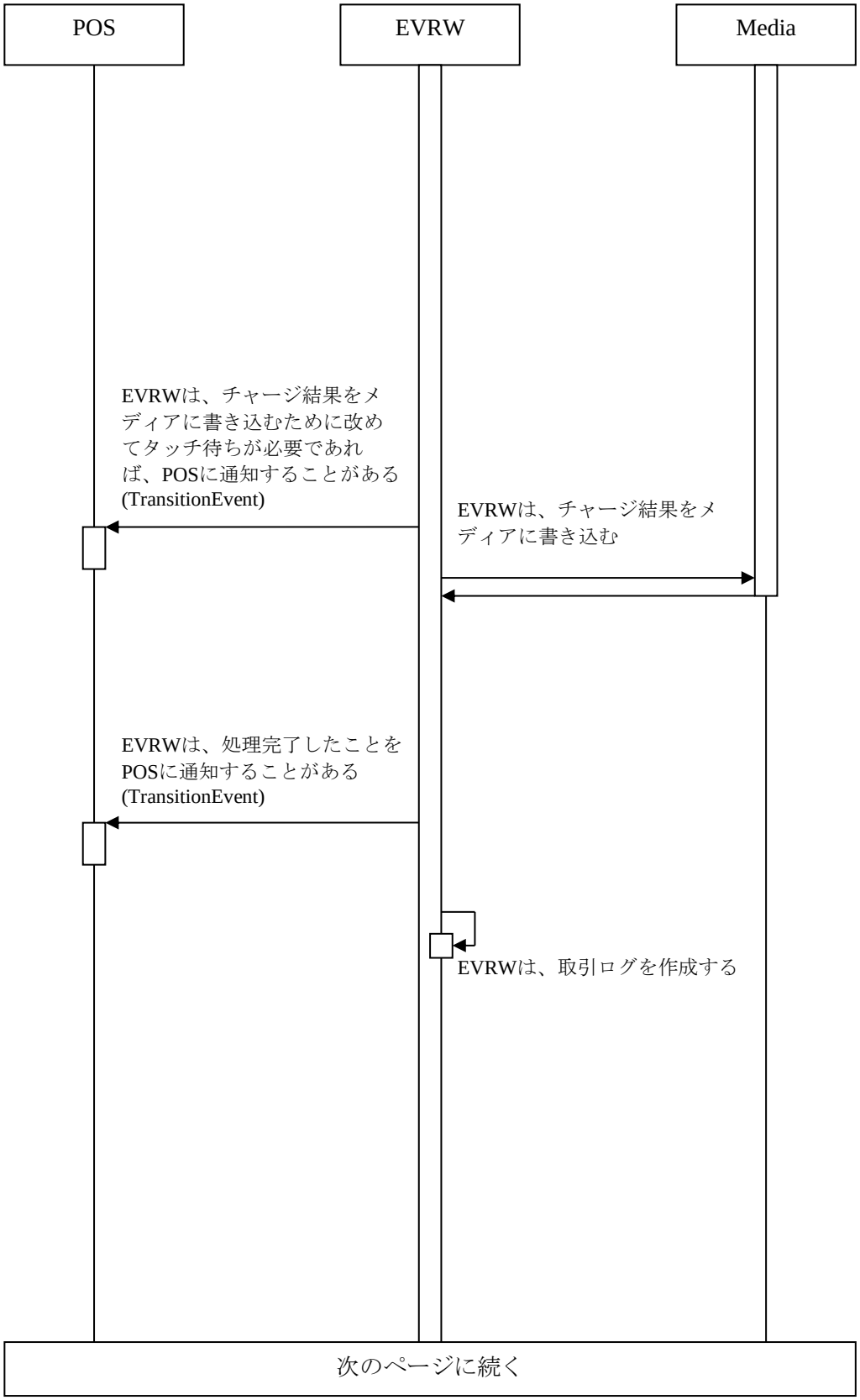

POS
EVRW
Media
EVRWは、残高が不足していれば、現金との併用決済を行うか、POSに問い合わせることがある(TransitionEvent)
EVRWは、オートチャージの条件に一致すれば、オートチャージすることをPOSに問い合わせることがある(TransitionEvent)
EVRWは、決済結果をメディアに書き込むために改めてタッチ待ちが必要であれば、POSに通知することがある(TransitionEvent)
EVRWは、決済結果をメディアに書き込む
EVRWは、処理完了したことをPOSに通知することがある(TransitionEvent)
EVRWは、取引ログを作成する
次のページに続く

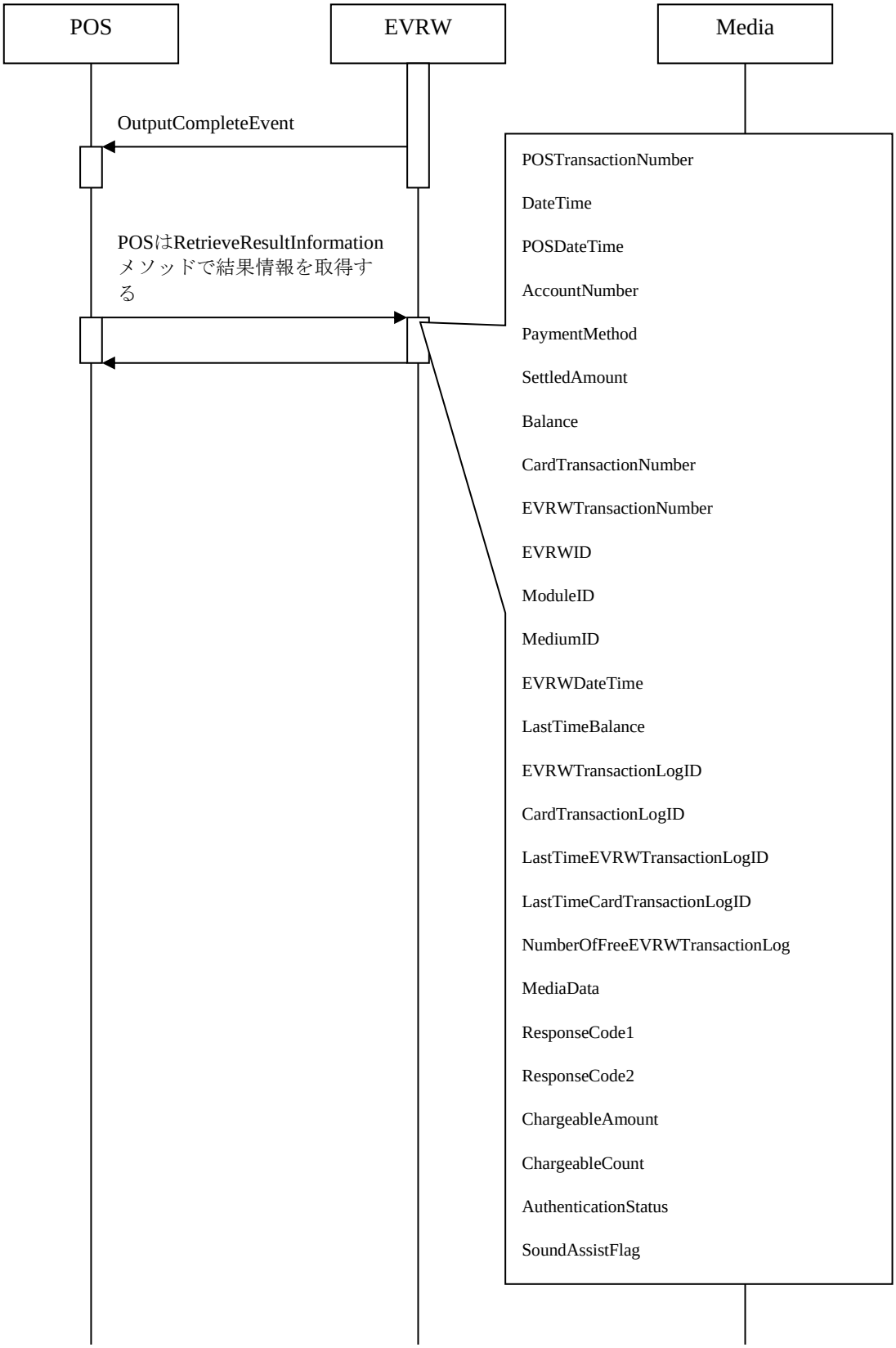

POS
EVRW
Media
OutputCompleteEvent
POSはRetrieveResultInformation
メソッドで結果情報を取得す
る
POSTransactionNumber
DateTime / POSDateTime
AccountNumber / BusinessUnitID
WorkstationID / AutoCharge
PaymentMethod
SettledAmount
Balance
CardTransactionNumber
EVRWTransactionNumber
EVRWID
ModuleID
MediumID
SettledOtherAmount
EVRWDateTime
LastTimeBalance
EVRWTransactionLogID
CardTransactionLogID
LastTimeEVRWTransactionLogID
LastTimeCardTransactionLogID
NumberOfFreeEVRWTransactionLog
MediaData
ResponseCode1
ResponseCode2
EVRWApprovalCode
CardCompanyName
SettledPoint
SettledAutoChargeAmount
InsufficientAmount
RequestedAutoChargeAmount
SoundAssistFlag
SignatureFlag

AddValueメソッドによるチャージ処理

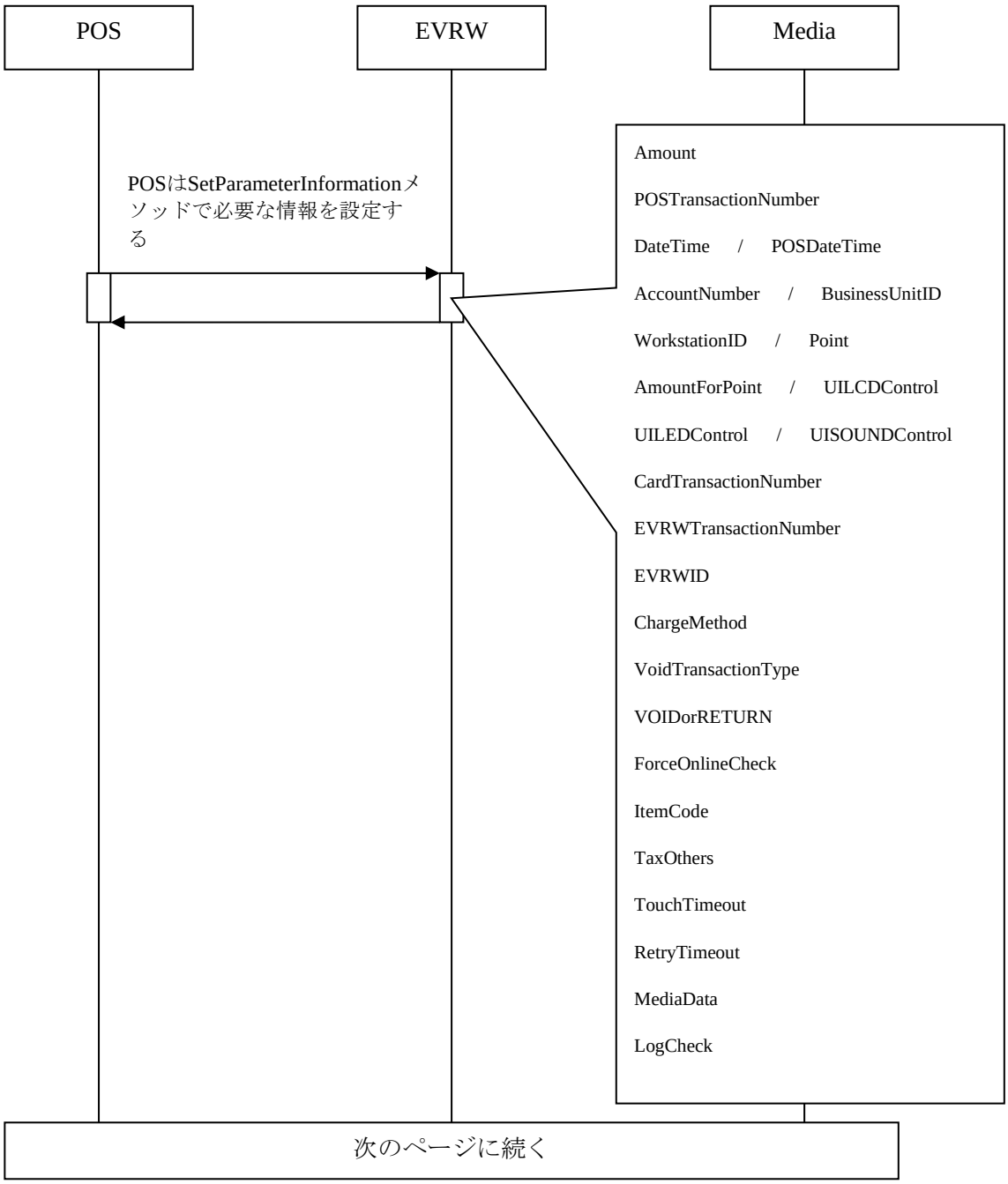


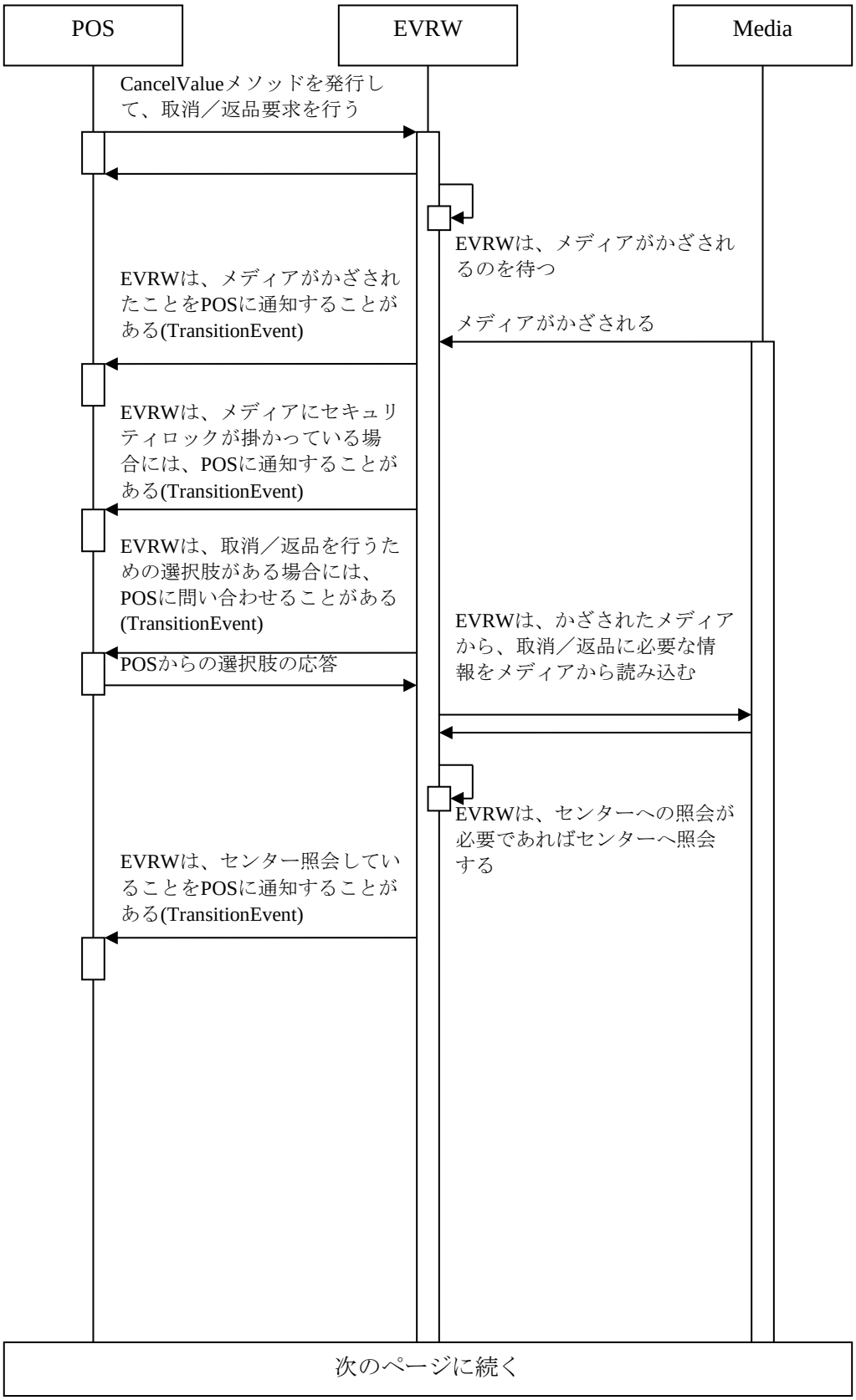

POS
EVRW
Media
AddValueメソッドを発行して、チャージ要求を行う
EVRWは、メディアがかざされるのを待つ
EVRWは、メディアがかざされたことをPOSに通知することがある(TransitionEvent)
メディアがかざされる
EVRWは、メディアにセキュリティロックが掛かっている場合には、POSに通知することがある(TransitionEvent)
EVRWは、チャージを行うための選択肢がある場合には、POSに問い合わせることがある(TransitionEvent)
EVRWは、かざされたメディアから、チャージに必要な情報をメディアから読み込む
POSからの選択肢の応答
EVRWは、センターへの照会が必要であればセンターへ照会する
EVRWは、センター照会していることをPOSに通知することがある(TransitionEvent)
次のページに続く

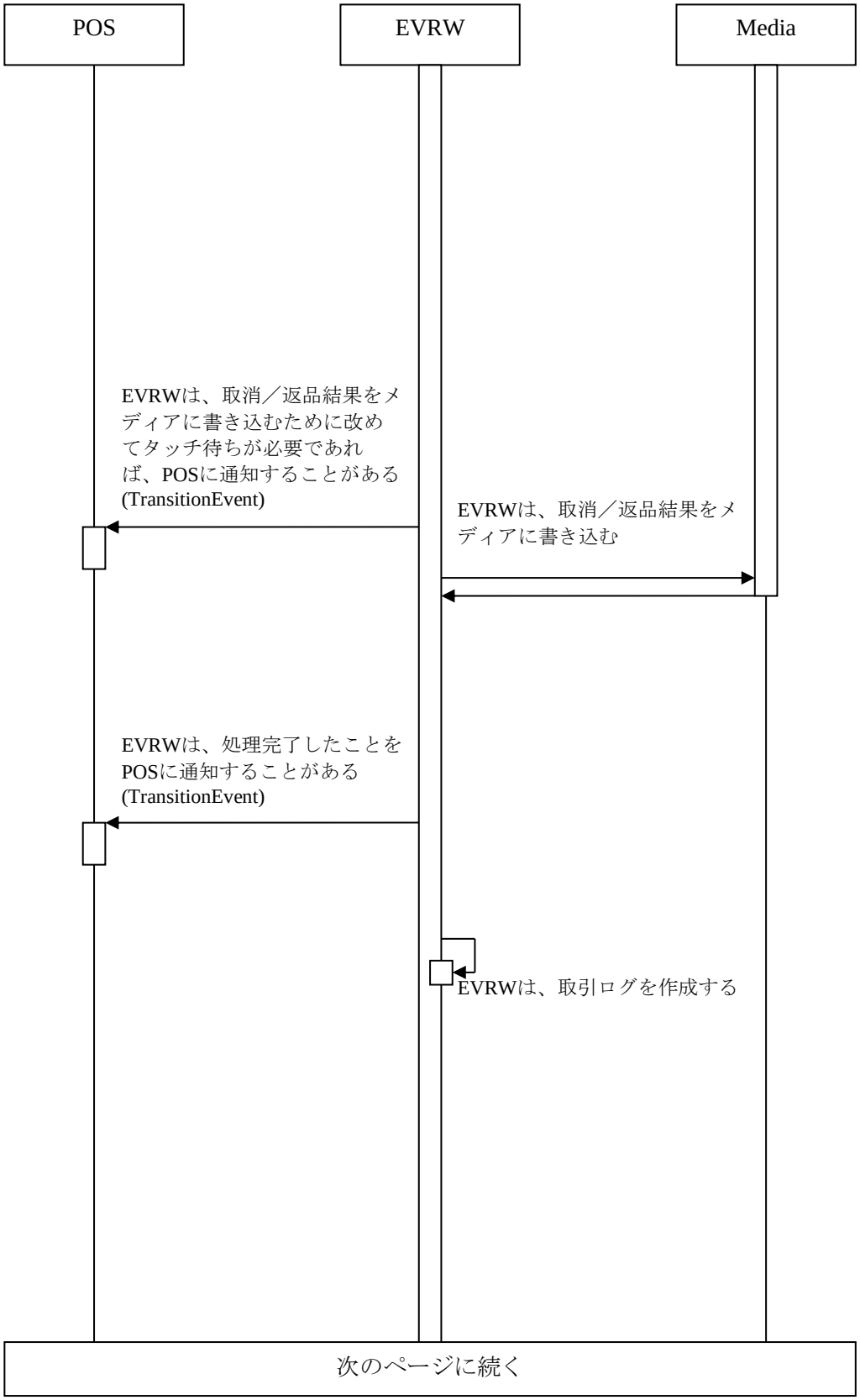

POS
EVRW
Media
EVRWは、チャージ結果をメディアに書き込むために改めてタッチ待ちが必要であれば、POSに通知することがある(TransitionEvent)
EVRWは、チャージ結果をメディアに書き込む
EVRWは、処理完了したことをPOSに通知することがある(TransitionEvent)
EVRWは、取引ログを作成する
次のページに続く

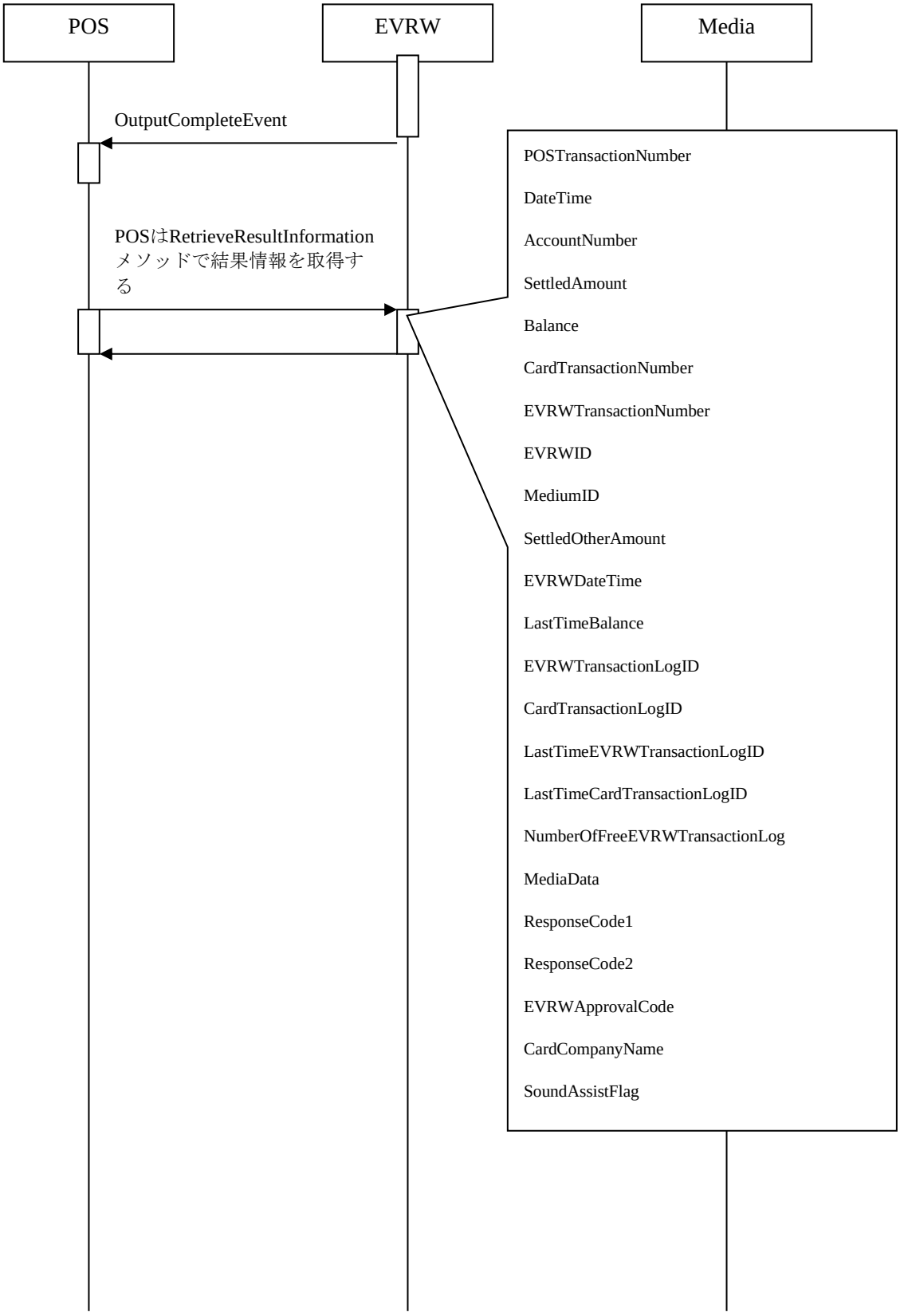

POS
EVRW
Media
OutputCompleteEvent
POSはRetrieveResultInformation
メソッドで結果情報を取得す
る
POSTransactionNumber
DateTime
POSDateTime
AccountNumber
PaymentMethod
SettledAmount
Balance
CardTransactionNumber
EVRWTransactionNumber
EVRWID
ModuleID
MediumID
EVRWDateTime
LastTimeBalance
EVRWTransactionLogID
CardTransactionLogID
LastTimeEVRWTransactionLogID
LastTimeCardTransactionLogID
NumberOfFreeEVRWTransactionLog
MediaData
ResponseCode1
ResponseCode2
ChargeableAmount
ChargeableCount
AuthenticationStatus
SoundAssistFlag

CancelValueメソッドによる取消／返品処理

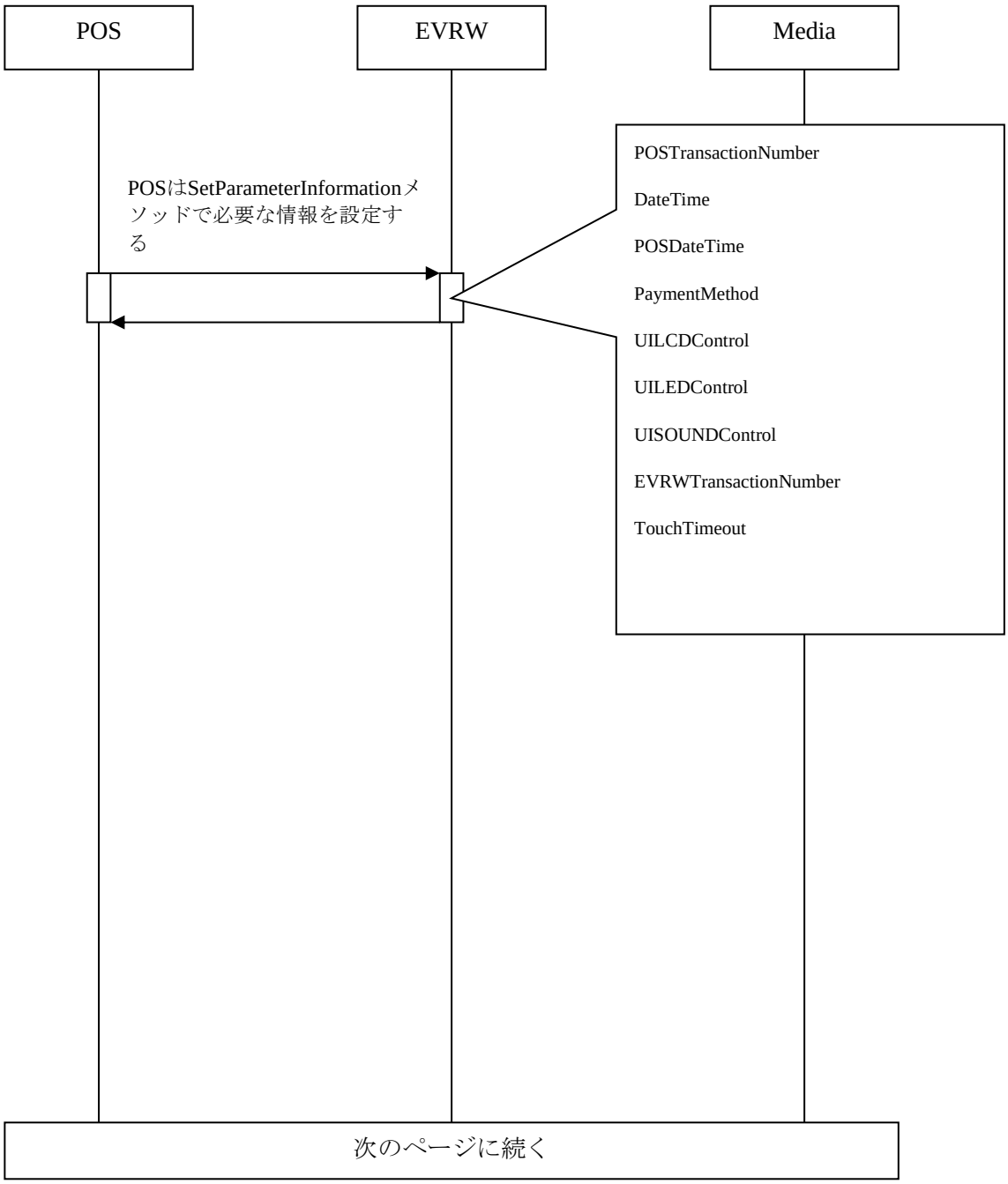


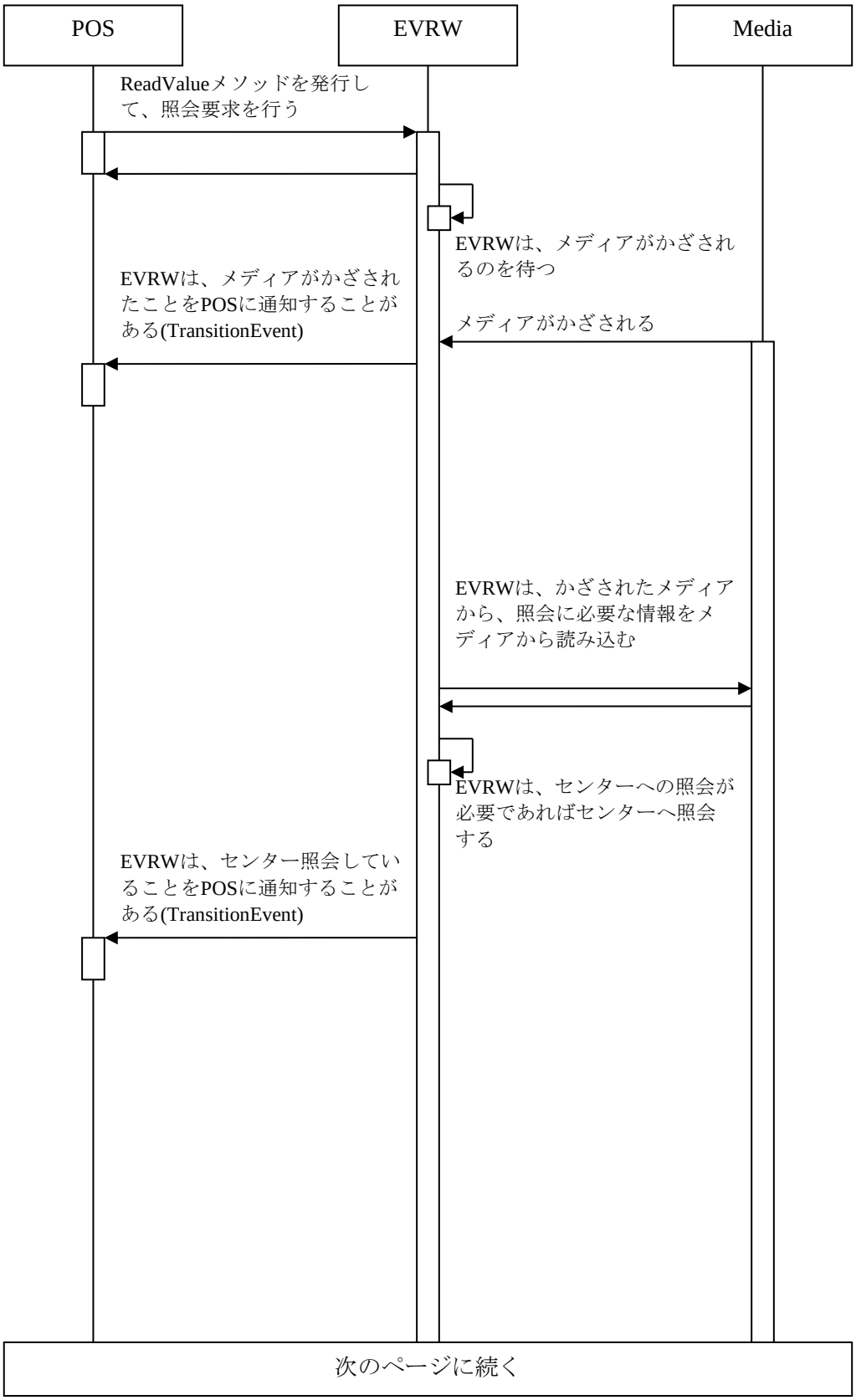

POS
EVRW
Media
CancelValueメソッドを発行して、取消／返品要求を行う
EVRWは、メディアがかざされるのを待つ
EVRWは、メディアがかざされたことをPOSに通知することがある(TransitionEvent)
メディアがかざされる
EVRWは、メディアにセキュリティロックが掛かっている場合には、POSに通知することがある(TransitionEvent)
EVRWは、取消／返品を行うための選択肢がある場合には、POSに問い合わせることがある(TransitionEvent)
EVRWは、かざされたメディアから、取消／返品に必要な情報をメディアから読み込む
POSからの選択肢の応答
EVRWは、センターへの照会が必要であればセンターへ照会する
EVRWは、センター照会していることをPOSに通知することがある(TransitionEvent)
次のページに続く

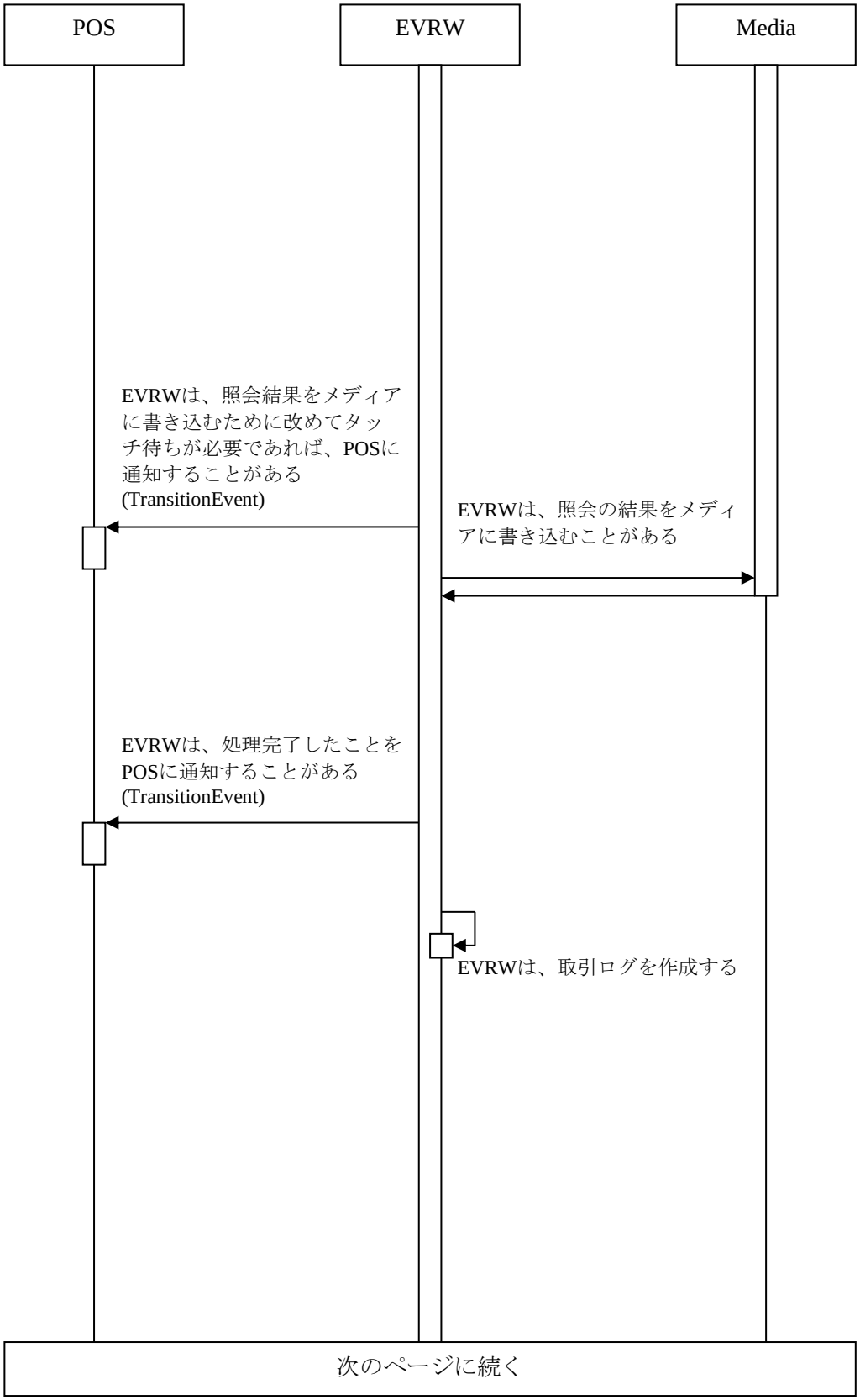

POS
EVRW
Media
EVRWは、取消／返品結果をメディアに書き込むために改めてタッチ待ちが必要であれば、POSに通知することがある(TransitionEvent)
EVRWは、取消／返品結果をメディアに書き込む
EVRWは、処理完了したことをPOSに通知することがある(TransitionEvent)
EVRWは、取引ログを作成する
次のページに続く

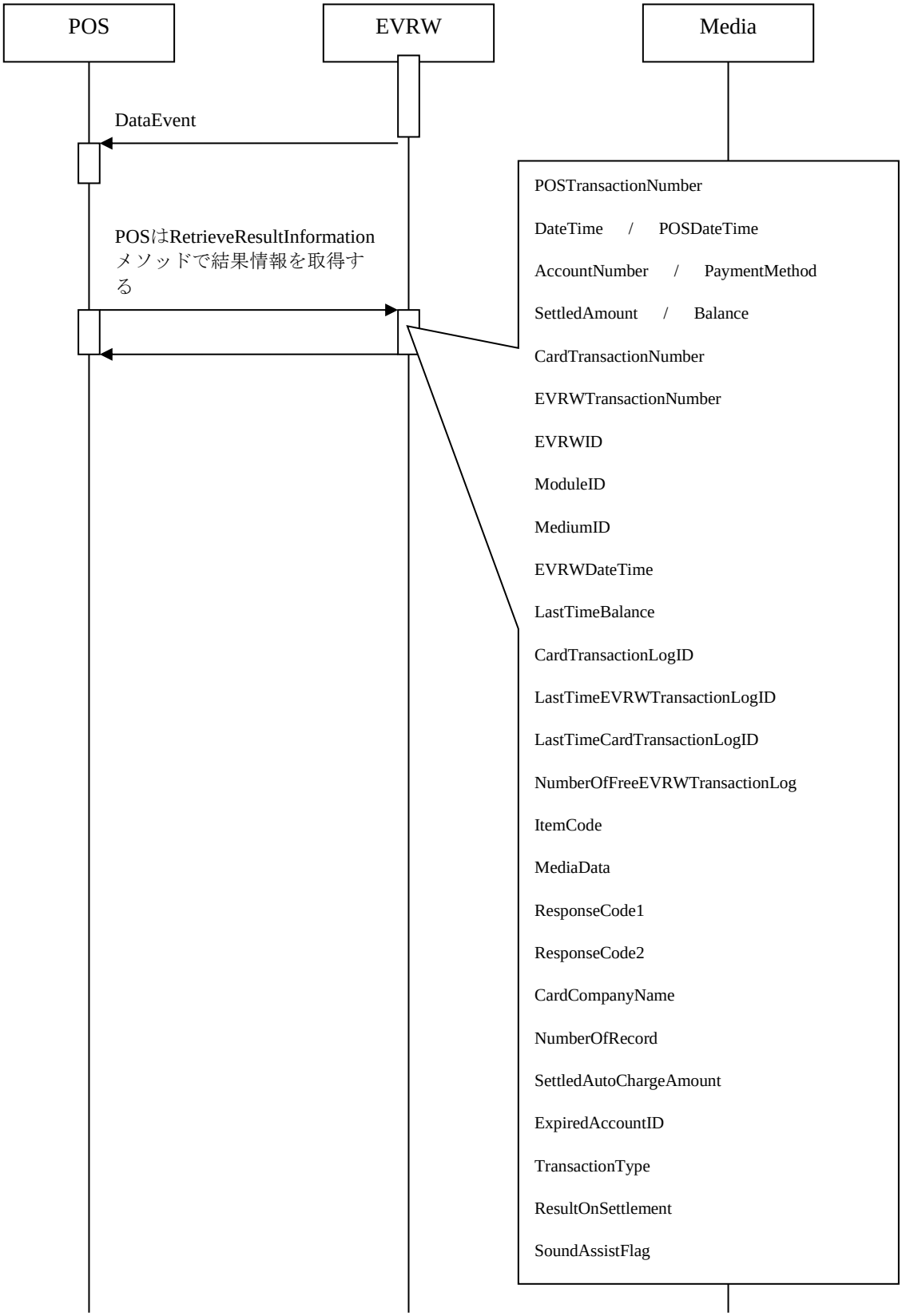

POS
EVRW
Media
OutputCompleteEvent
POSはRetrieveResultInformation
メソッドで結果情報を取得す
る
POSTransactionNumber
DateTime
AccountNumber
SettledAmount
Balance
CardTransactionNumber
EVRWTransactionNumber
EVRWID
MediumID
SettledOtherAmount
EVRWDateTime
LastTimeBalance
EVRWTransactionLogID
CardTransactionLogID
LastTimeEVRWTransactionLogID
LastTimeCardTransactionLogID
NumberOfFreeEVRWTransactionLog
MediaData
ResponseCode1
ResponseCode2
EVRWApprovalCode
CardCompanyName
SoundAssistFlag

ReadValueメソッドによる照会処理

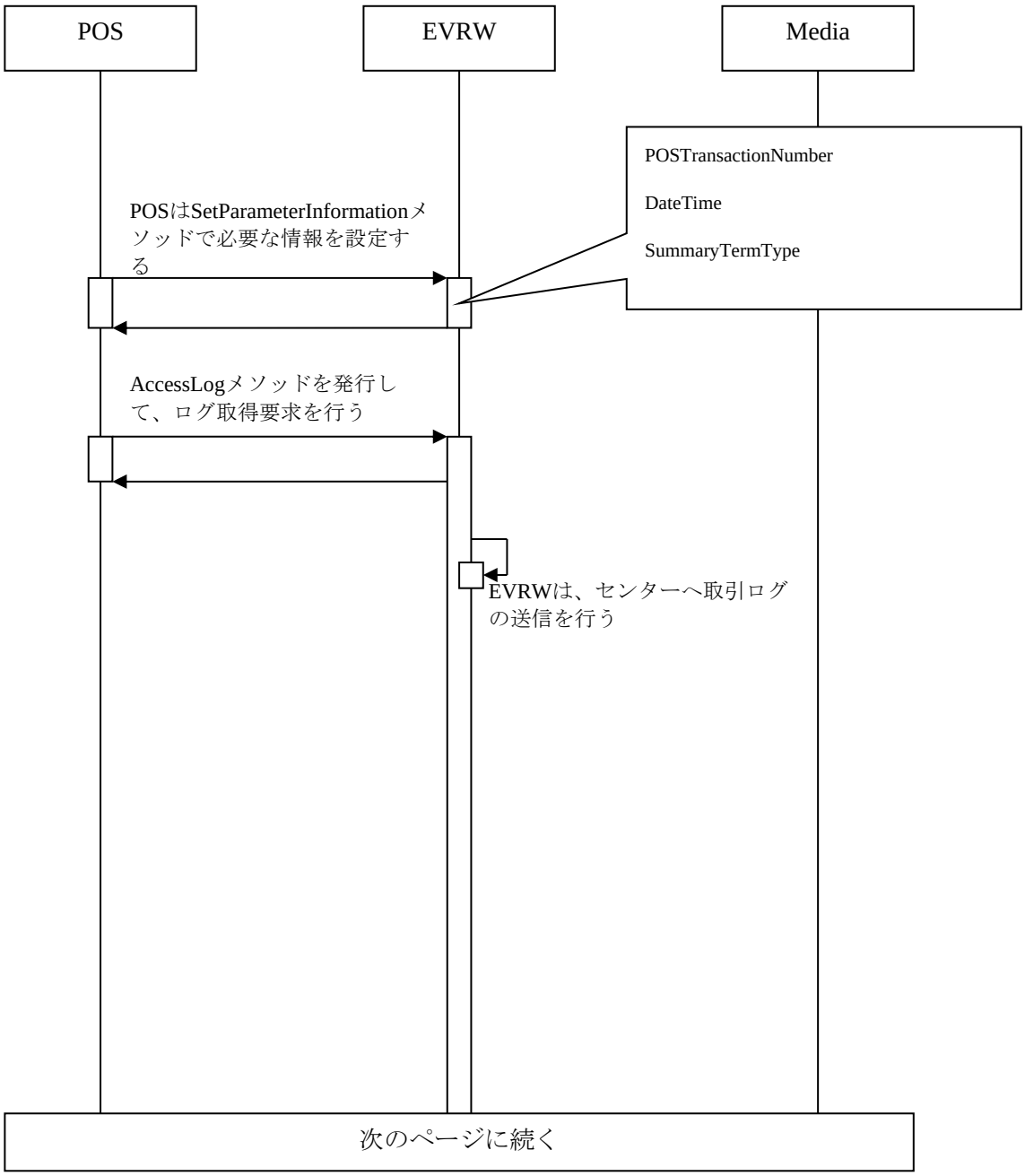


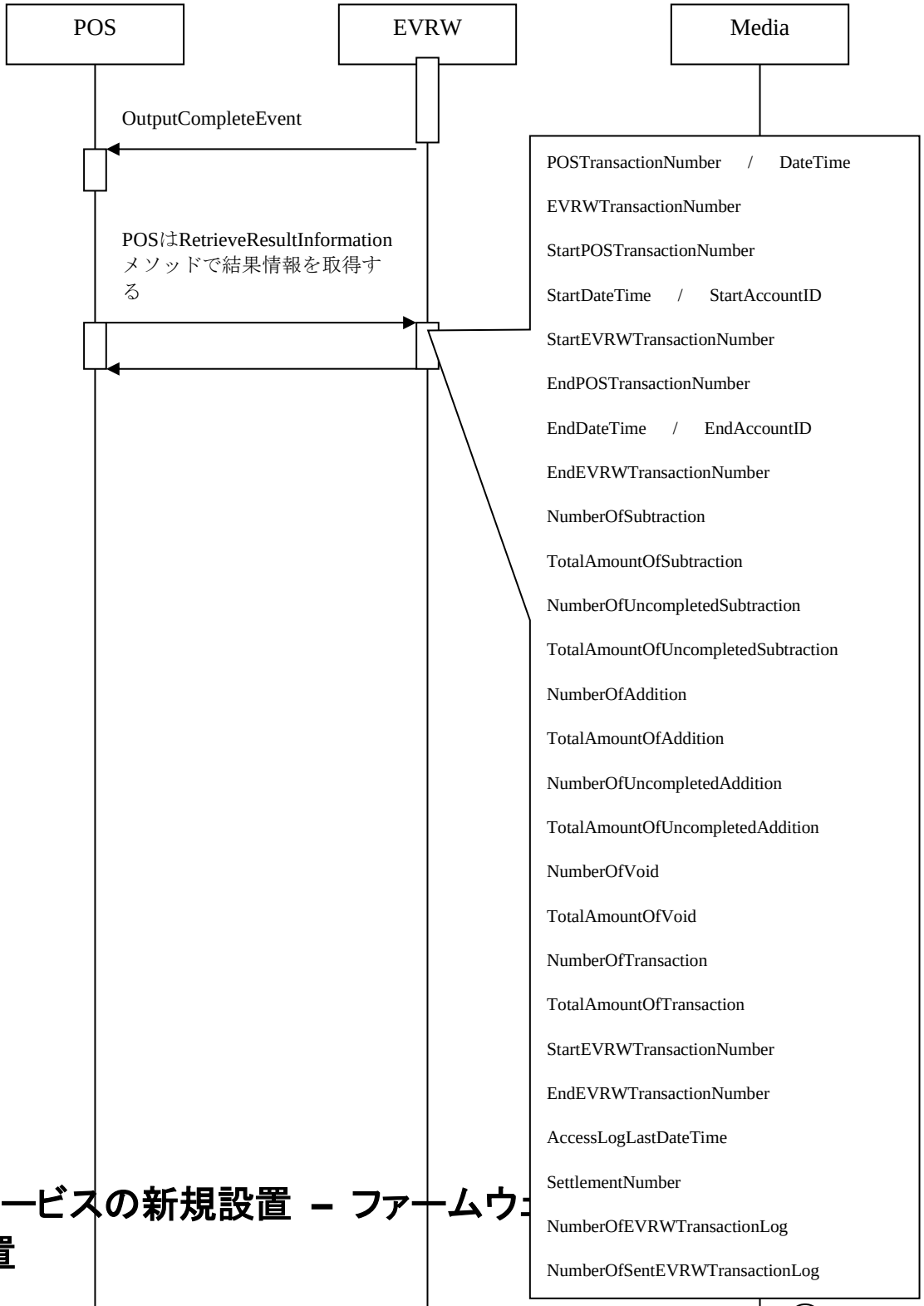

POS
EVRW
Media
ReadValueメソッドを発行して、照会要求を行う
EVRWは、メディアがかざされるのを待つ
EVRWは、メディアがかざされたことをPOSに通知することがある(TransitionEvent)
メディアがかざされる
EVRWは、かざされたメディアから、照会に必要な情報をメディアから読み込む
EVRWは、センターへの照会が必要であればセンターへ照会する
EVRWは、センター照会していることをPOSに通知することがある(TransitionEvent)
次のページに続く


POS
EVRW
Media
EVRWは、照会結果をメディアに書き込むために改めてタッチ待ちが必要であれば、POSに通知することがある
(TransitionEvent)
EVRWは、照会の結果をメディアに書き込むことがある
EVRWは、処理完了したことをPOSに通知することがある
(TransitionEvent)
EVRWは、取引ログを作成する
次のページに続く

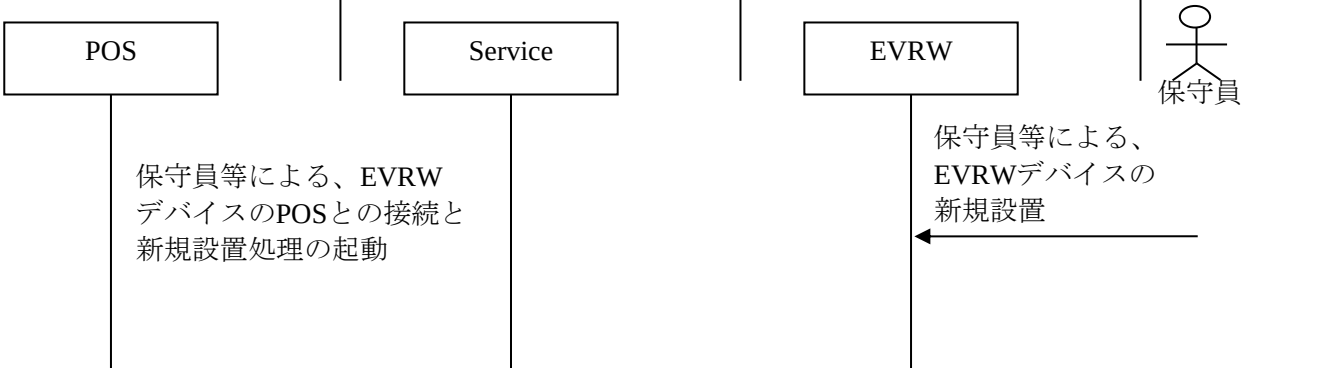

POS
EVRW
Media
DataEvent
POSはRetrieveResultInformation
メソッドで結果情報を取得す
る
POSTransactionNumber
DateTime / POSDateTime
AccountNumber / PaymentMethod
SettledAmount / Balance
CardTransactionNumber
EVRWTransactionNumber
EVRWID
ModuleID
MediumID
EVRWDateTime
LastTimeBalance
CardTransactionLogID
LastTimeEVRWTransactionLogID
LastTimeCardTransactionLogID
NumberOfFreeEVRWTransactionLog
ItemCode
MediaData
ResponseCode1
ResponseCode2
CardCompanyName
NumberOfRecord
SettledAutoChargeAmount
ExpiredAccountID
TransactionType
ResultOnSettlement
SoundAssistFlag

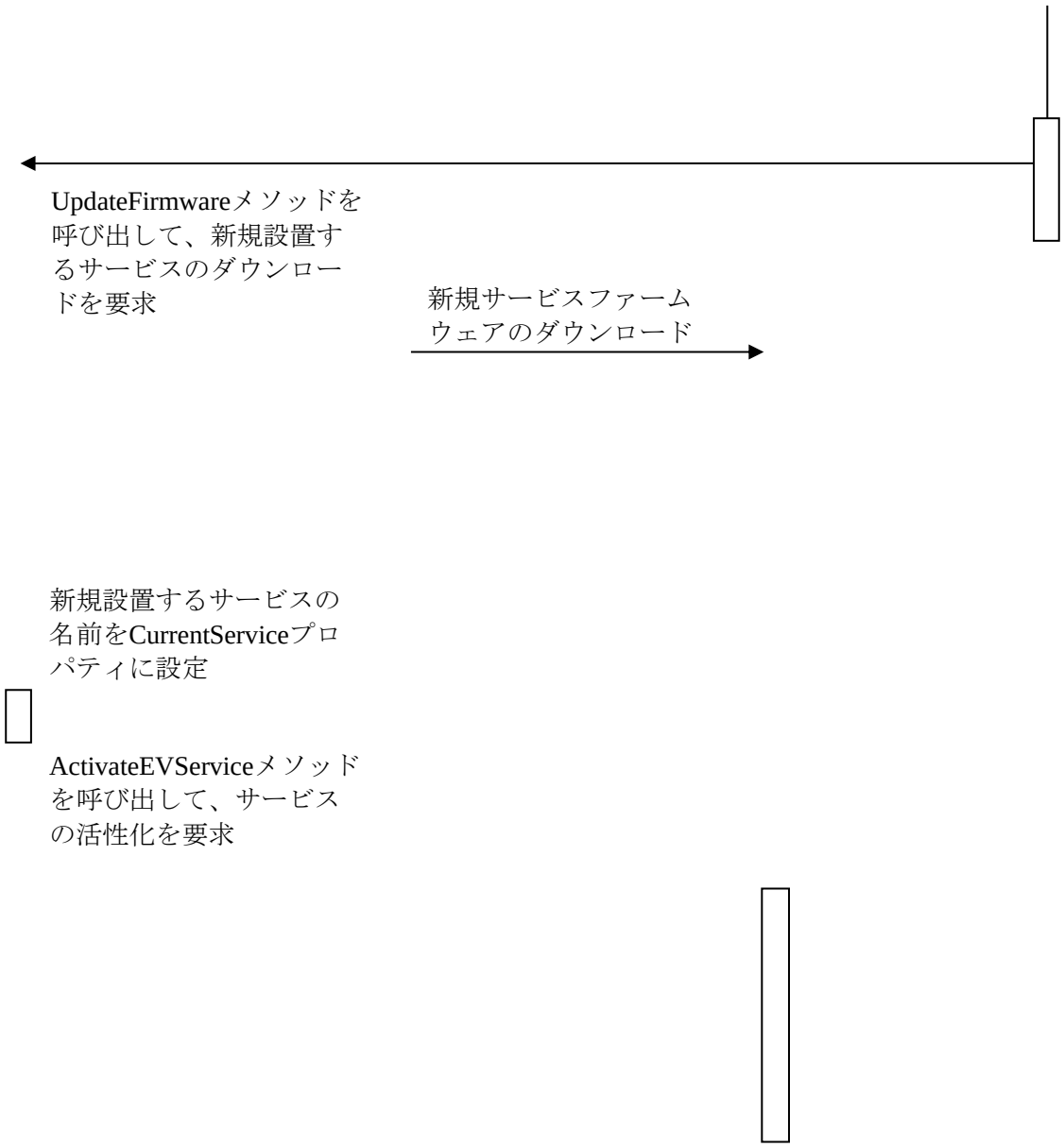
AccessLogメソッドによるログ取得処理

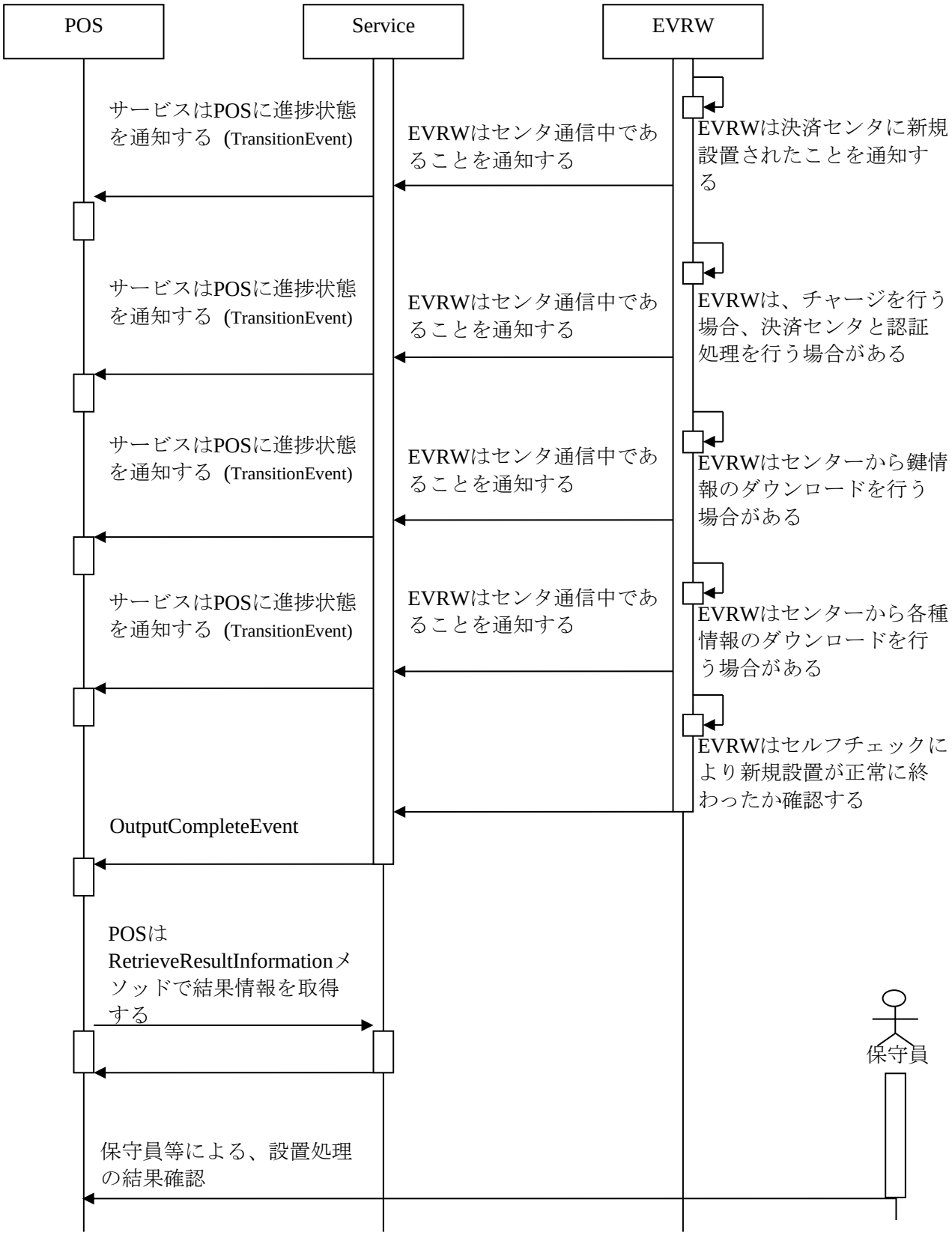




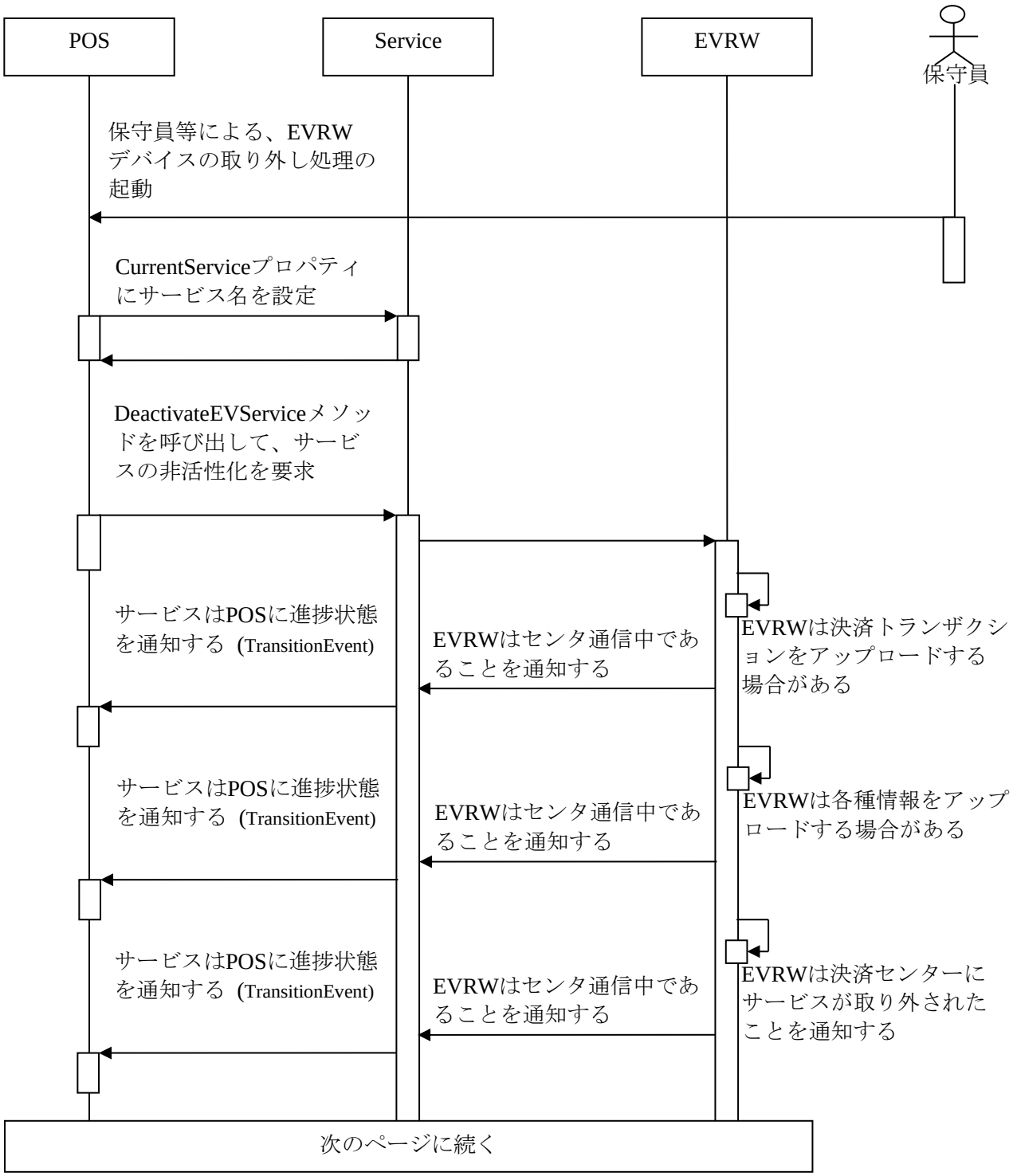
EVRWサービスの新規設置 – ファームウ
新規設置

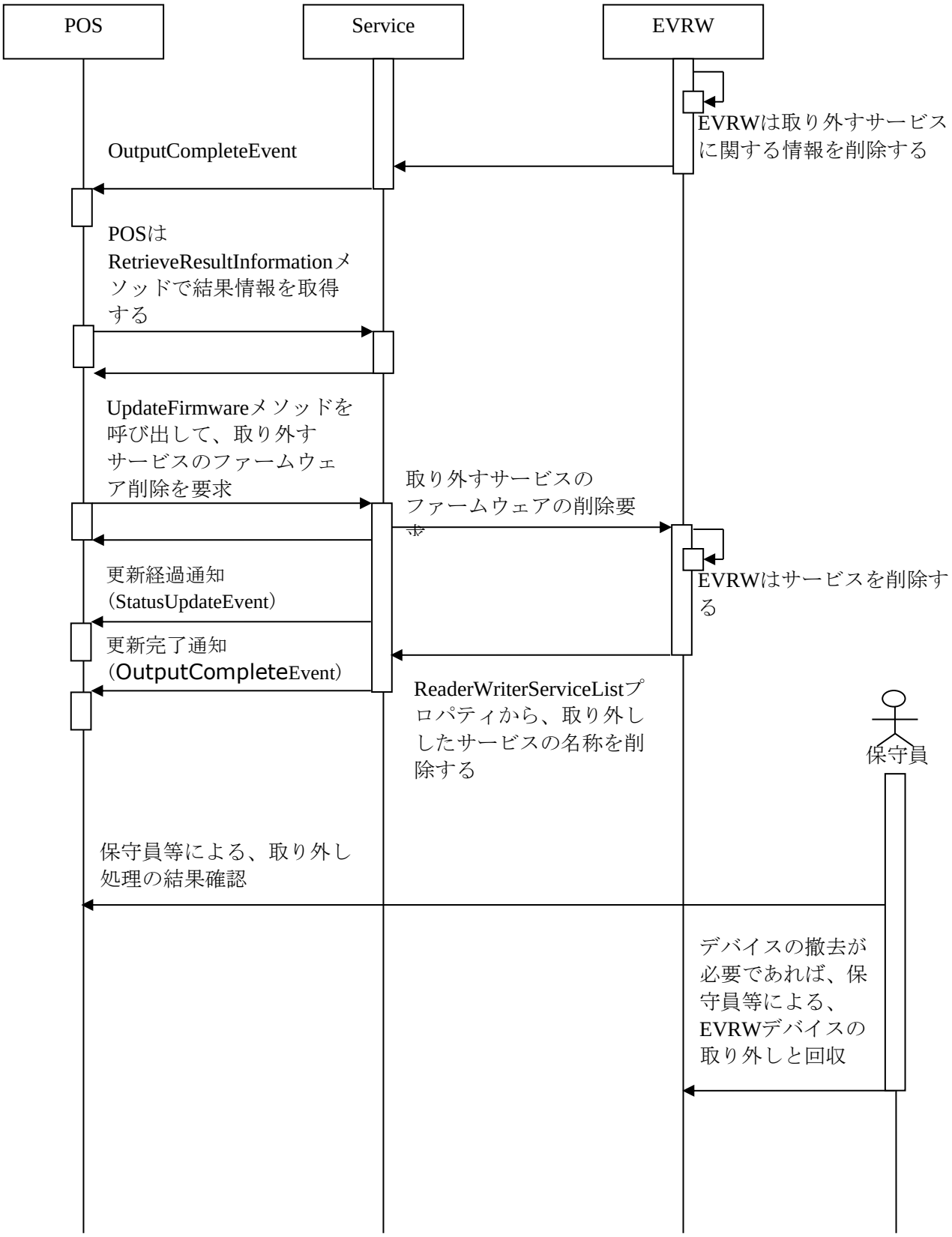



UpdateFirmwareメソッドを
呼び出して、新規設置す
るサービスのダウンロー
ドを要求
新規サービスファーム
ウェアのダウンロード
新規設置するサービスの
名前をCurrentServiceプロ
パティに設定
ActivateEVServiceメソッド
を呼び出して、サービス
の活性化を要求

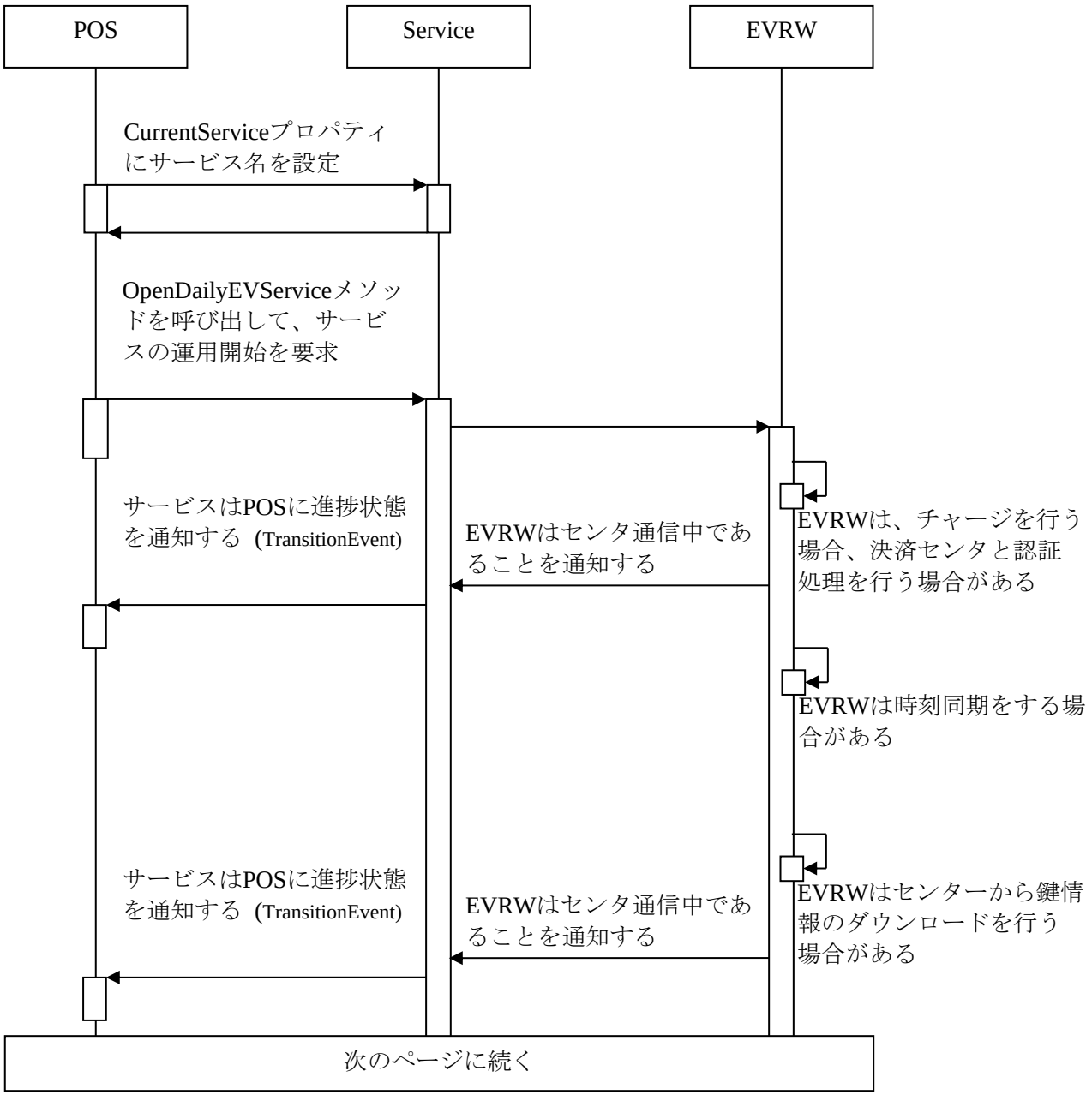

POS
Service
EVRW
サービスはPOSに進捗状態を通知する（TransitionEvent）
EVRWはセンタ通信中であることを通知する
EVRWは決済センタに新規設置されたことを通知する
サービスはPOSに進捗状態を通知する（TransitionEvent）
EVRWはセンタ通信中であることを通知する
EVRWは、チャージを行う場合、決済センタと認証処理を行う場合がある
サービスはPOSに進捗状態を通知する（TransitionEvent）
EVRWはセンタ通信中であることを通知する
EVRWはセンターから鍵情報のダウンロードを行う場合がある
サービスはPOSに進捗状態を通知する（TransitionEvent）
EVRWはセンタ通信中であることを通知する
EVRWはセンターから各種情報のダウンロードを行う場合がある
EVRWはセルフチェックにより新規設置が正常に終わったか確認する
OutputCompleteEvent
POSはRetrieveResultInformationメソッドで結果情報を取得する
保守員
保守員等による、設置処理の結果確認

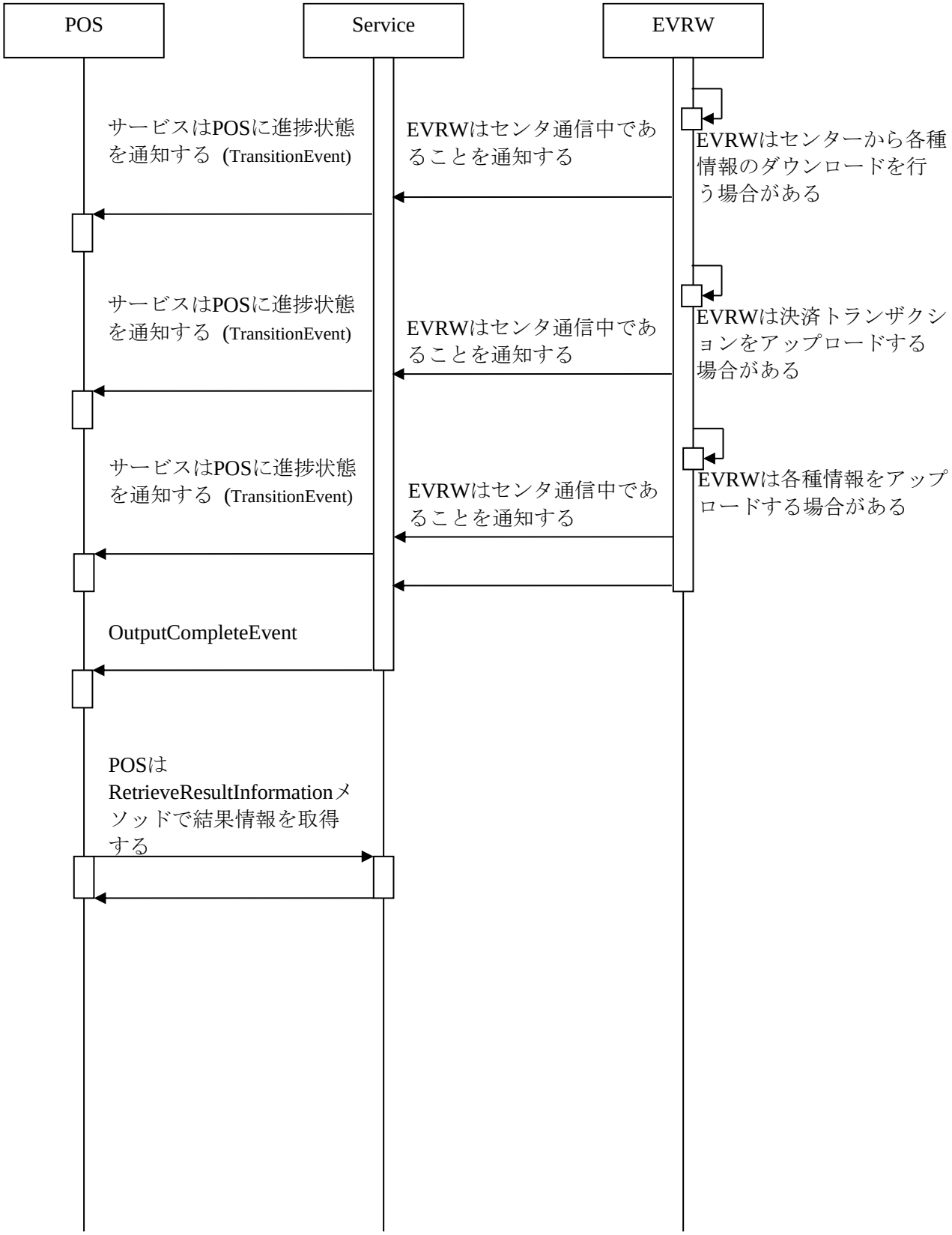
EVRW取り外し


POS
Service
EVRW
保守員
保守員等による、EVRW
デバイスの取り外し処理の
起動
CurrentServiceプロパティ
にサービス名を設定
DeactivateEVServiceメソッ
ドを呼び出して、サービ
スの非活性化を要求
サービスはPOSに進捗状態
を通知する (TransitionEvent)
EVRWはセンタ通信中であ
ることを通知する
EVRWは決済トランザクシ
ョンをアップロードする
場合がある
サービスはPOSに進捗状態
を通知する (TransitionEvent)
EVRWはセンタ通信中であ
ることを通知する
EVRWは各種情報をアップ
ロードする場合がある
サービスはPOSに進捗状態
を通知する (TransitionEvent)
EVRWはセンタ通信中であ
ることを通知する
EVRWは決済センターに
サービスが取り外された
ことを通知する
次のページに続く

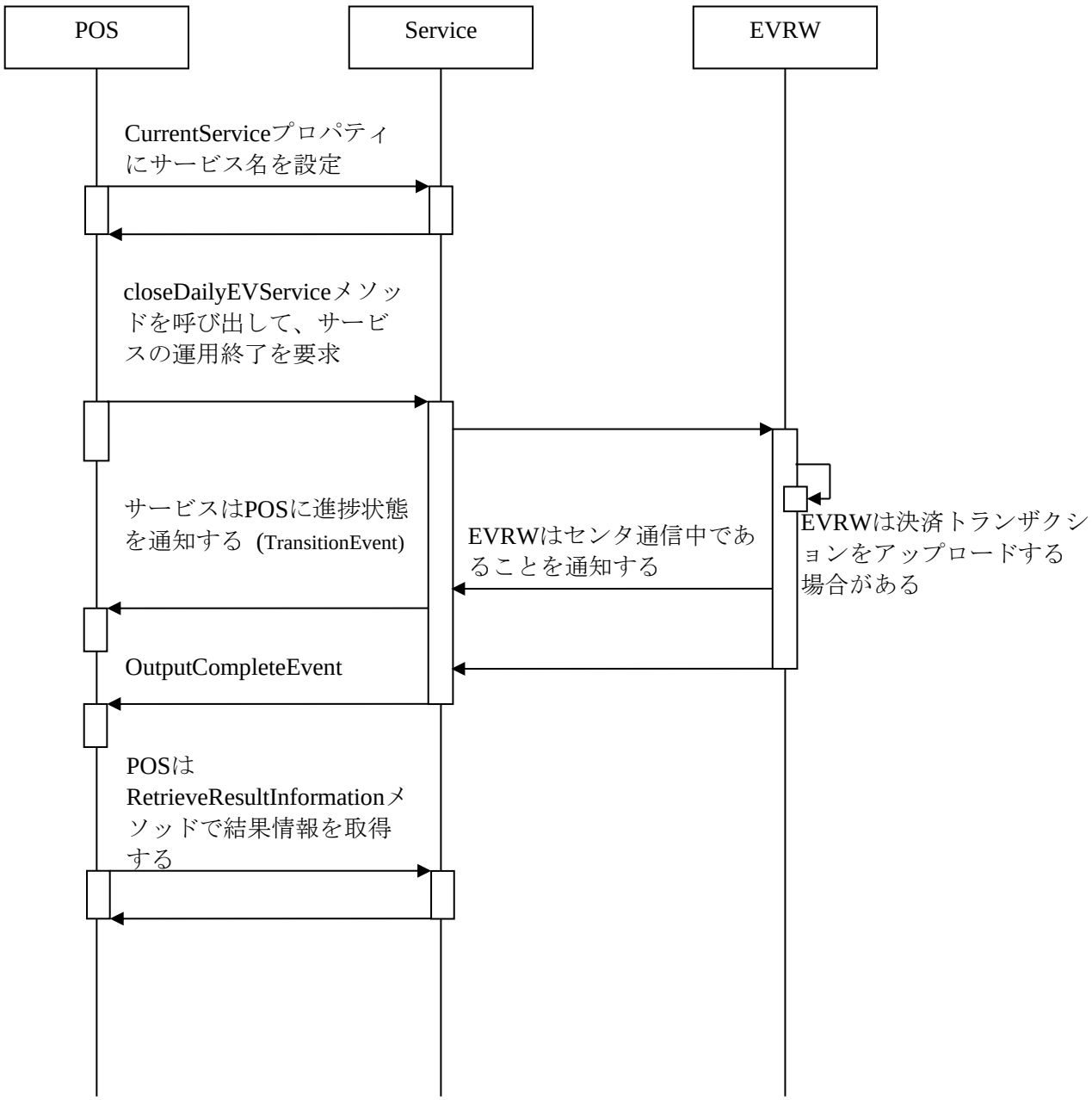

POS
Service
EVRW
EVRWは取り外すサービスに関する情報を削除する
OutputCompleteEvent
POSはRetrieveResultInformationメソッドで結果情報を取得する
UpdateFirmwareメソッドを呼び出して、取り外すサービスのファームウェア削除を要求
取り外すサービスのファームウェアの削除要求
EVRWはサービスを削除する
更新経過通知（StatusUpdateEvent）
更新完了通知（OutputCompleteEvent）
ReaderWriterServiceListプロパティから、取り外ししたサービスの名称を削除する
保守員
保守員等による、取り外し処理の結果確認
デバイスの撤去が必要であれば、保守員等による、EVRWデバイスの取り外しと回収

EVRW運用開始

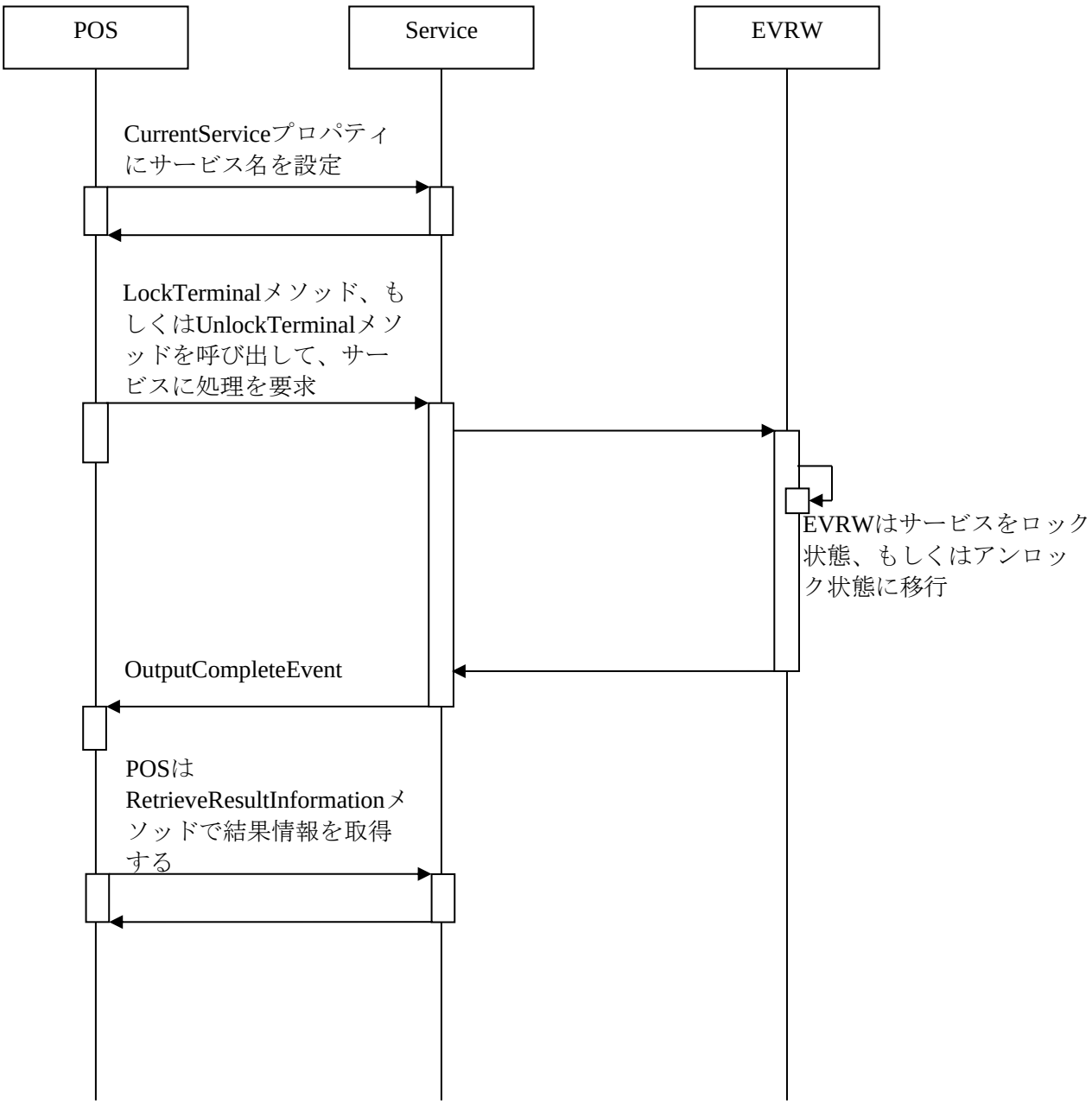

POS
Service
EVRW
CurrentServiceプロパティにサービス名を設定
OpenDailyEVServiceメソッドを呼び出して、サービスの運用開始を要求
サービスはPOSに進捗状態を通知する (TransitionEvent)
EVRWはセンタ通信中であることを通知する
EVRWは、チャージを行う場合、決済センタと認証処理を行う場合がある
EVRWは時刻同期をする場合がある
サービスはPOSに進捗状態を通知する (TransitionEvent)
EVRWはセンタ通信中であることを通知する
EVRWはセンターから鍵情報のダウンロードを行う場合がある
次のページに続く


POS
Service
EVRW
サービスはPOSに進捗状態を通知する (TransitionEvent)
EVRWはセンタ通信中であることを通知する
EVRWはセンターから各種情報のダウンロードを行う場合がある
サービスはPOSに進捗状態を通知する (TransitionEvent)
EVRWはセンタ通信中であることを通知する
EVRWは決済トランザクションをアップロードする場合がある
サービスはPOSに進捗状態を通知する (TransitionEvent)
EVRWはセンタ通信中であることを通知する
EVRWは各種情報をアップロードする場合がある
OutputCompleteEvent
POSはRetrieveResultInformationメソッドで結果情報を取得する

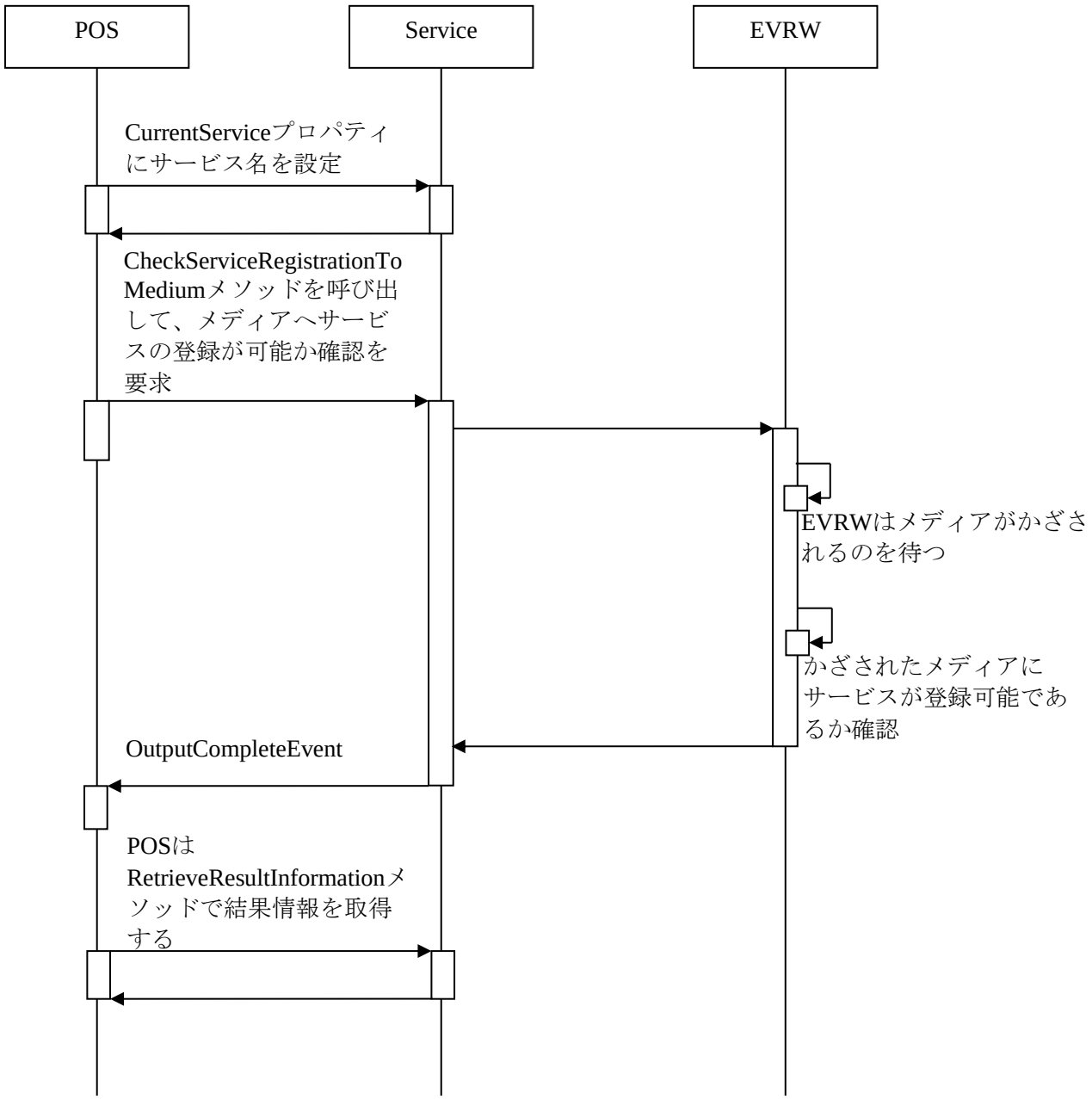
EVRW運用終了


POS
Service
EVRW
CurrentServiceプロパティにサービス名を設定
closeDailyEVServiceメソッドを呼び出して、サービスの運用終了を要求
サービスはPOSに進捗状態を通知する (TransitionEvent)
EVRWはセンタ通信中であることを通知する
EVRWは決済トランザクションをアップロードする場合がある
OutputCompleteEvent
POSはRetrieveResultInformationメソッドで結果情報を取得する

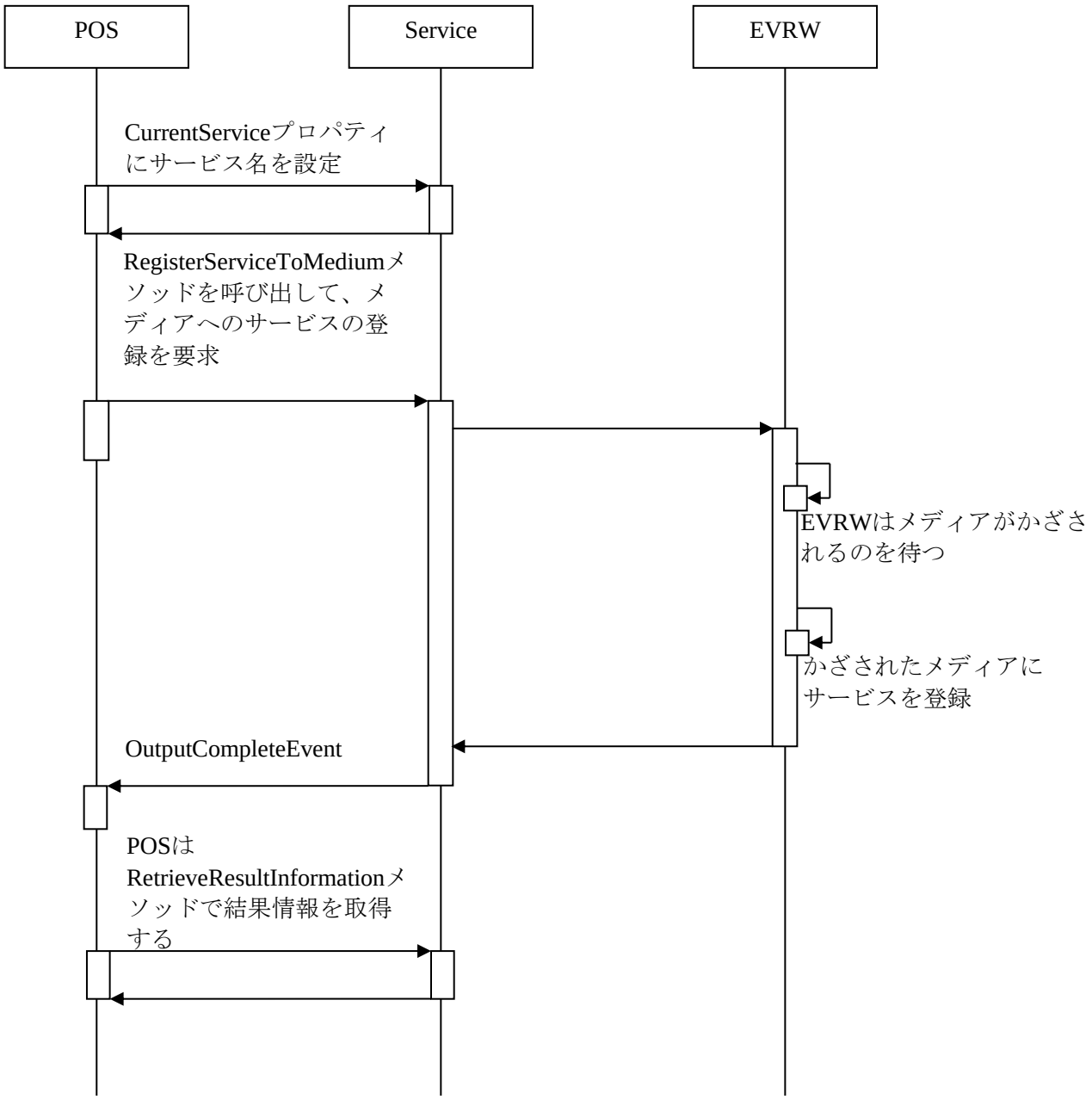
EVRWロック／アンロック



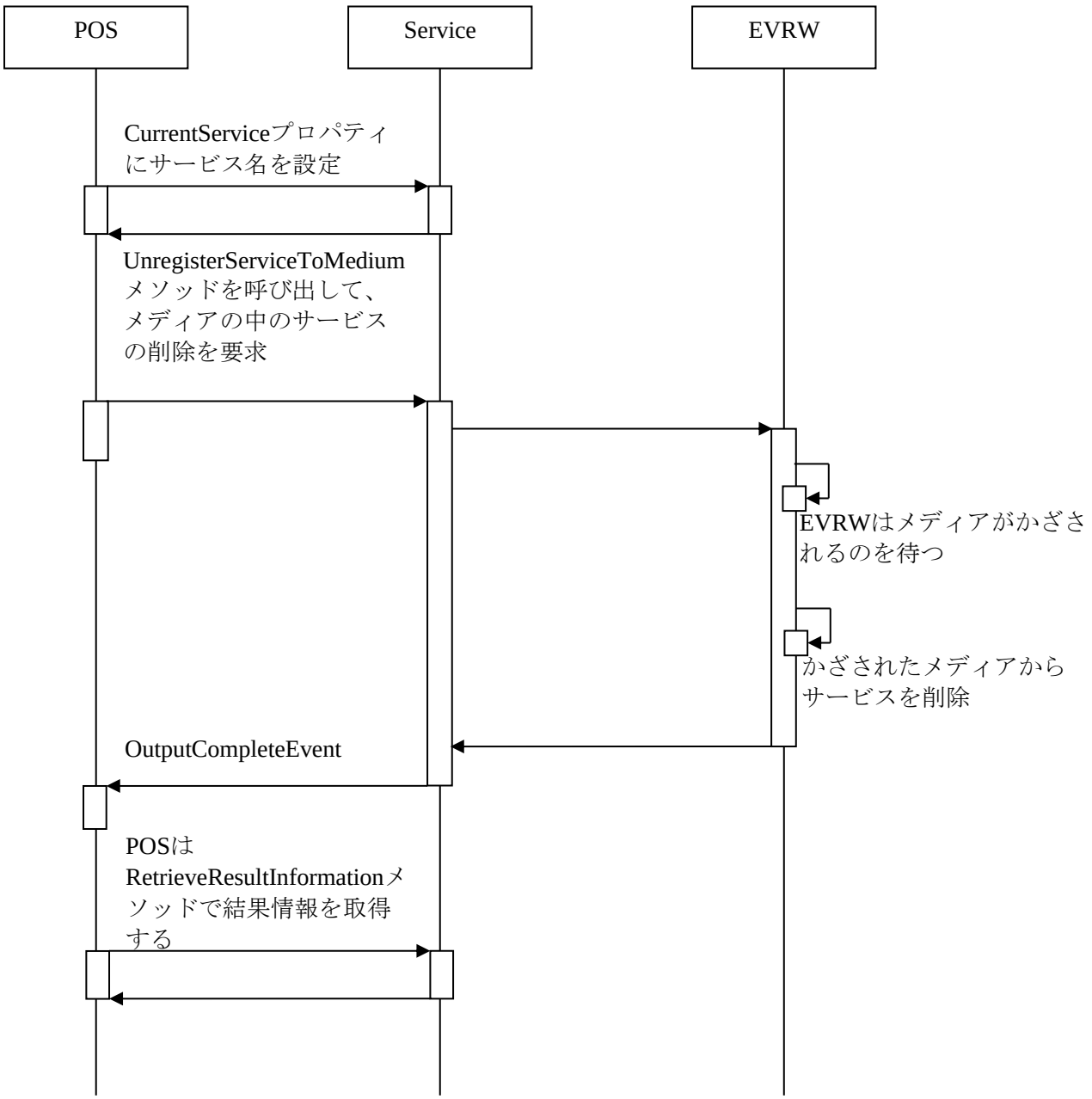
メディアへサービス登録可能かの確認



メディアへのサービスの登録

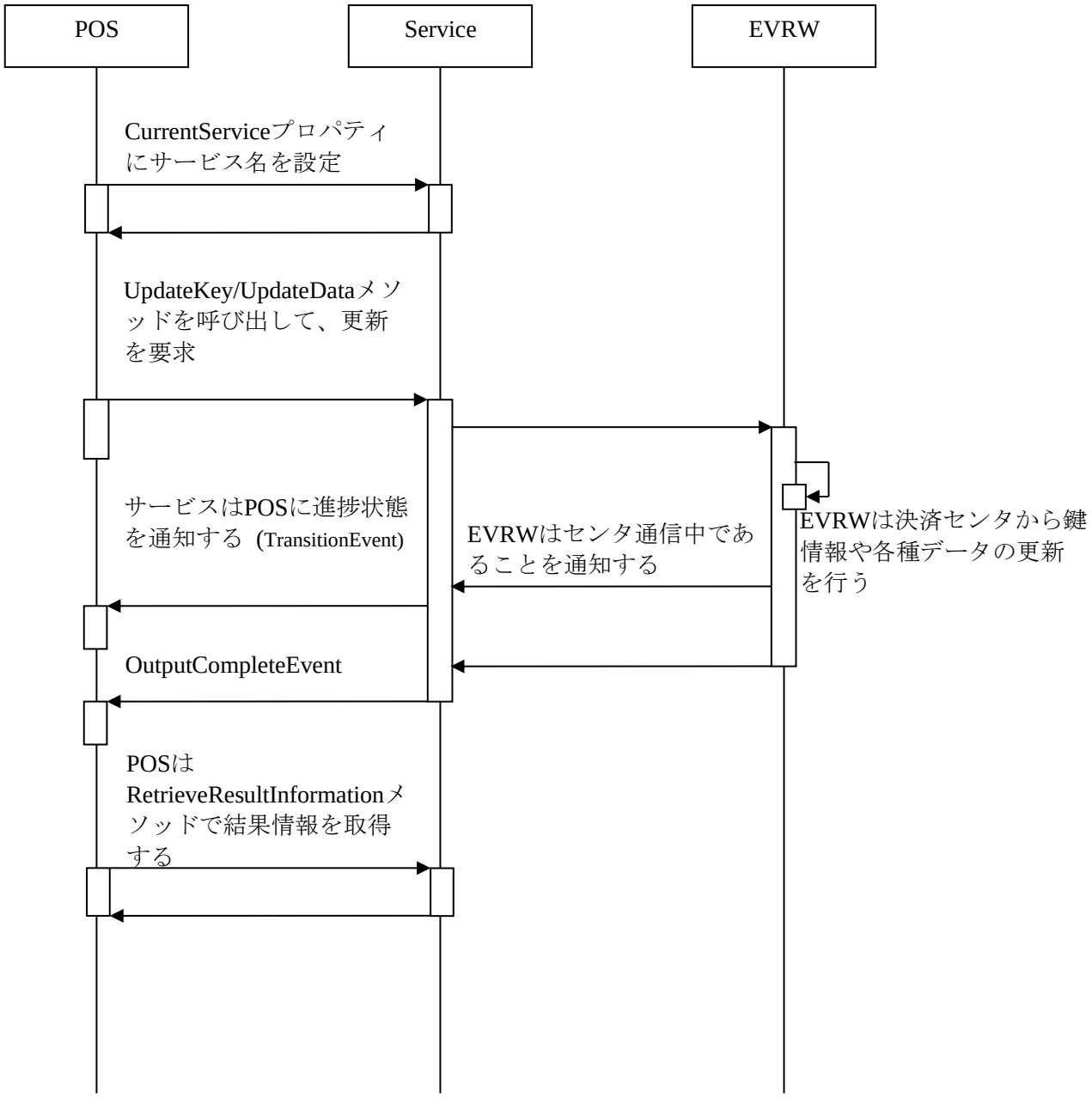


メディアの中のサービスの削除



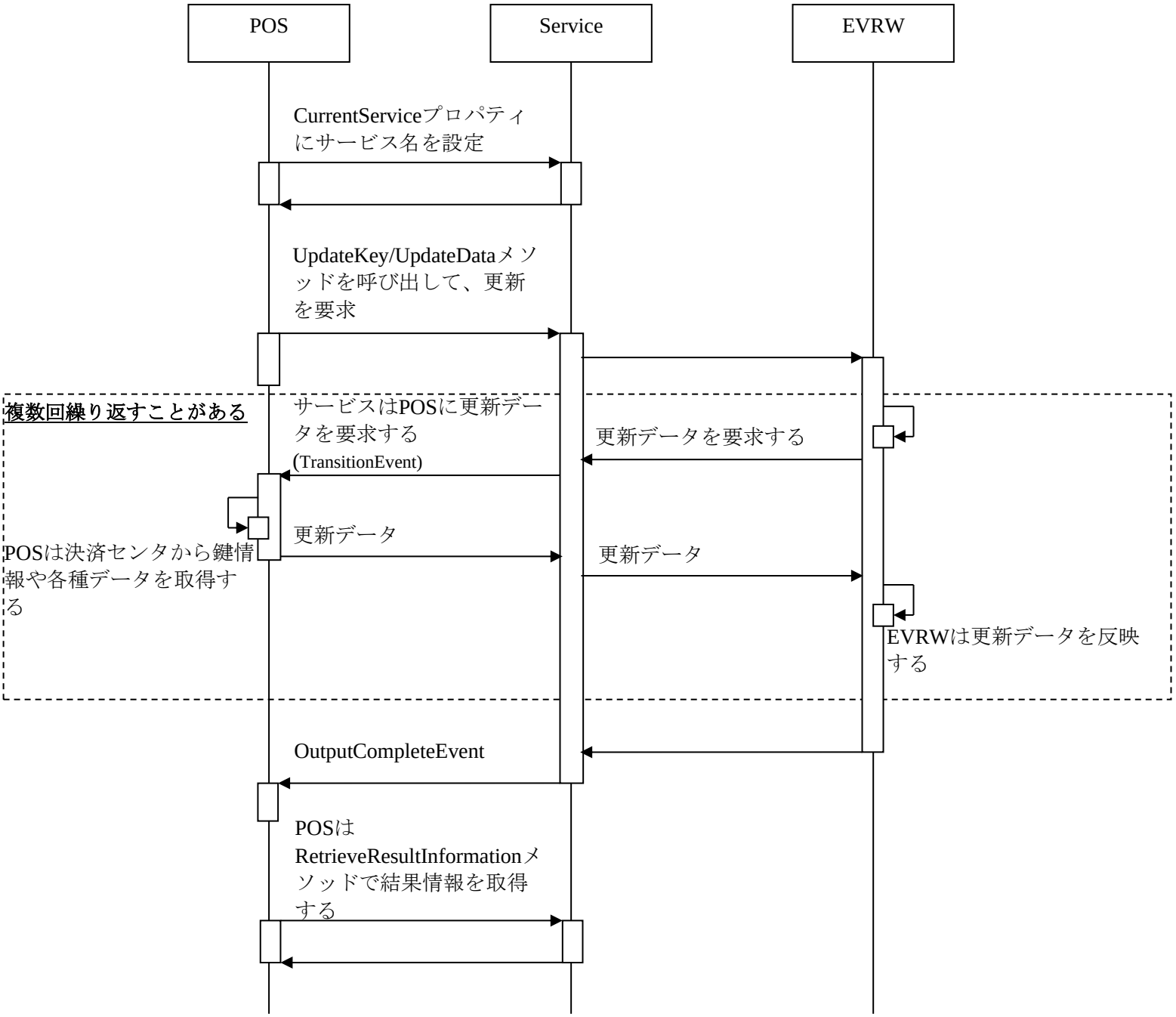
更新（外回り）

外回り：EVRW デバイスが決済センタと直接接続された方式



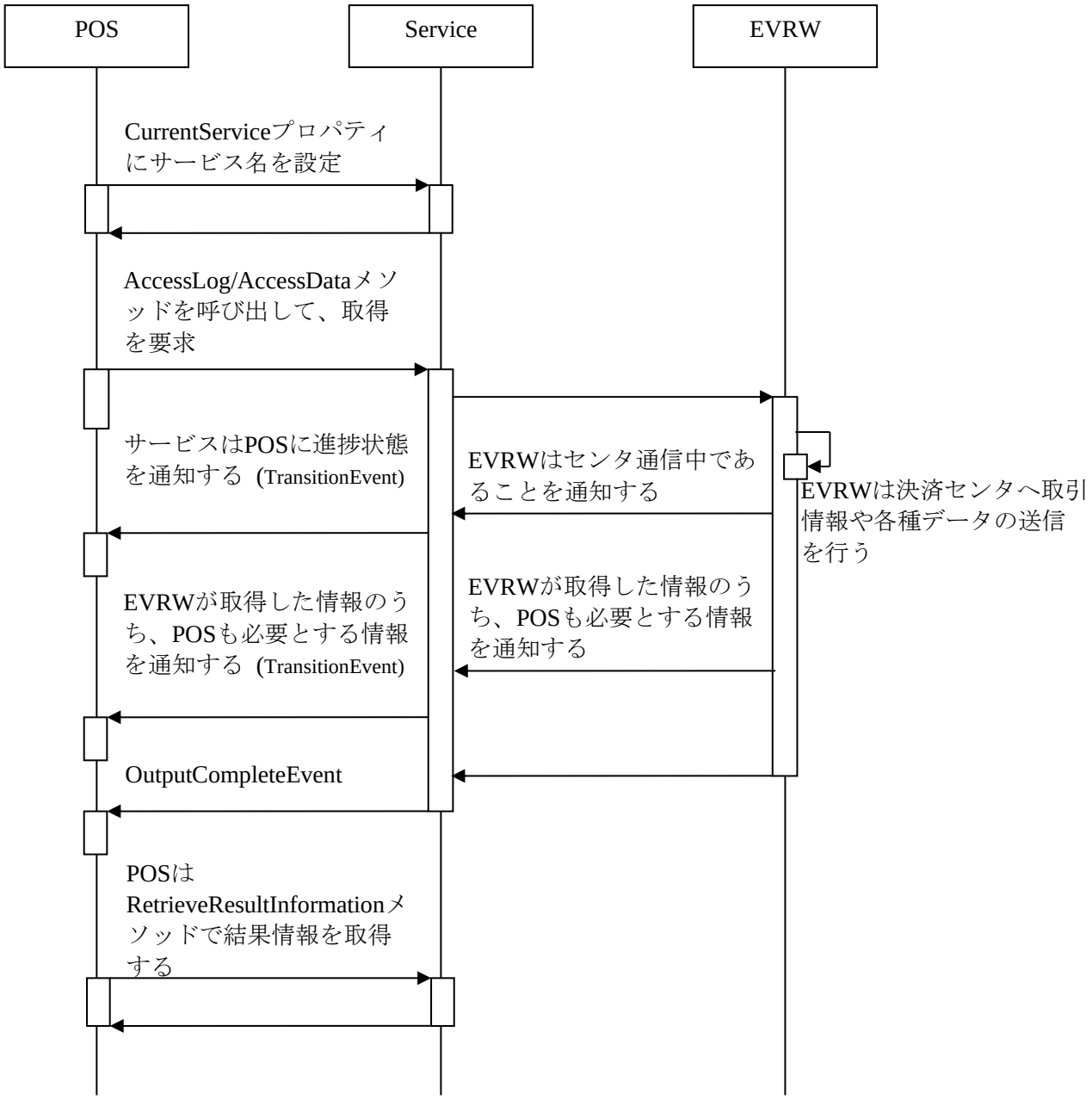
更新（内回り）

内回り：EVRW デバイスが POS を通じて決済センタと接続された方式



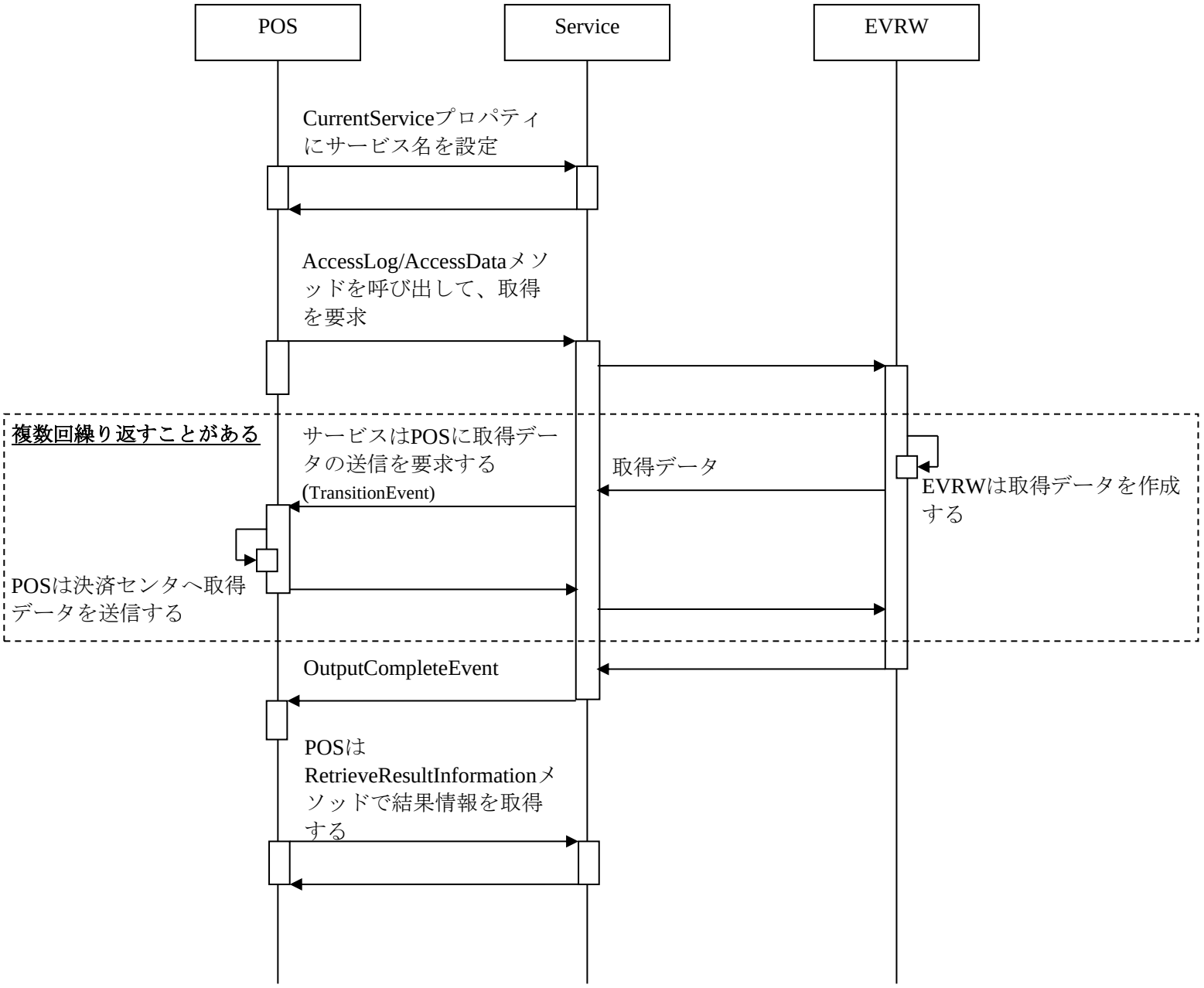
取得（外回り）

外回り：EVRW デバイスが決済センタと直接接続された方式



取得（内回り）

内回り：EVRW デバイスが POS を通じて決済センタと接続された方式



プロパティ

AccountNumber プロパティ　第1.14版で変更

形式　**BSTR AccountNumber;**

説明　カードが持つ情報で、カード番号、会員番号など、サービス提供者がユーザー（所有者）を特定するための情報を設定します。

SetParameterInfomationメソッドおよび**RetrieveResultInformation**メソッドのタグに同様の情報を持つため、このプロパティを使用せずにタグの情報を利用することを推奨します。

AdditionalSecurityInformation プロパティ　R/W

形式　**BSTR AdditionalSecurityInformation;**

説明　メソッド発行前に、このプロパティに任意のデータをセットすることにより、任意のデータをデバイスへ送信できます。また処理結果として、デバイスから送られたデータで他のプロパティに格納されないデータをこのプロパティに格納することにより、アプリケーションへ渡すことが可能です。格納されるデータはデバイスの実装に依存します。

設定されるデータのフォーマットは**BinaryConversion**プロパティの値に依存します。詳細は**BinaryConversion**プロパティを参照してください。

Amount プロパティ　R/W　第1.14版で変更

形式　**CURRENCY Amount;**

説明　電子マネーサービスにおいて、決済金額を設定します。

SetParameterInfomationメソッドおよび**RetrieveResultInformation**メソッドのタグに同様の情報を持つため、このプロパティを使用せずにタグの情報を利用することを推奨します。

ApprovalCode プロパティ　R/W

形式　**BSTR ApprovalCode;**

説明　決済結果としての「決済番号」、承認結果としての「承認番号」を設定します。

決済番号の内容はデバイスの実装に依存します。デバイスで行われた処理に対して一意な番号が発行される場合、その情報が設定されます。

デバイスが決済の取消をサポートする場合、取消する決済を指定するために、以下のメソッドの発行前に、プロパティを設定します。

CancelValue

AsyncMode プロパティ R/W

形式　**BOOL AsyncMode;**

説明　TRUE: 専用メソッドが非同期的に実行されます。

FALSE: 同期的に実行されます。

このプロパティは、**Open**メソッドによりFALSEに初期化されます。

戻り値　次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティが正常にセットされました。

Balance プロパティ　第1.14版で変更

形式　**CURRENCY Balance;**

説明　電子マネーサービスにおいて、残高を設定します。

SetParameterInfomationメソッドおよび**RetrieveResultInformation**メソッドのタグに同様の情報を持つため、このプロパティを使用せずにタグの情報を利用することを推奨します。

BalanceOfPoint プロパティ

形式　　**CURRENCY BalanceOfPoint;**

説明　　ポイントサービスにおいて、残ポイントを設定します。

SetParameterInfomationメソッドおよび**RetrieveResultInformation**メソッドのタグに同様の情報を持つため、このプロパティを使用せずにタグの情報を利用することを推奨します。

CapActivateService プロパティ

形式　　**BOOL CapActivateService;**

説明　　TRUE：活性化処理が有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapAdditionalSecurityInformationプロパティ　　第1.15版で追加

形式　　BOOL CapAdditionalSecurityInformation;

説明　　TRUE: AdditionalSecurityInformationプロパティは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　AdditionalSecurityInformationプロパティ

CapAddValue プロパティ

形式　　**BOOL CapAddValue;**

説明　　TRUE：電子バリューの追加が有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapAuthorizeCompletionプロパティ　第1.15版で追加

形式　BOOL CapAuthorizeCompletion;

説明　TRUE: AuthorizeCompletionメソッドは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　AuthorizeCompletionメソッド

CapAuthorizePreSalesプロパティ　第1.15版で追加

形式　BOOL CapAuthorizePreSales;

説明　TRUE: AuthorizePreSalesメソッドは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　AuthorizePreSalesメソッド

CapAuthorizeRefundプロパティ　　第1.15版で追加

形式　　BOOL CapAuthorizeRefund;

説明　　TRUE: AuthorizeRefundメソッドは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　AuthorizeRefundメソッド

CapAuthorizeVoidプロパティ　　第1.15版で追加

形式　　BOOL CapAuthorizeVoid;

説明　　TRUE: AuthorizeVoidメソッドは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　AuthorizeVoidメソッド

CapAuthorizeVoidPreSalesプロパティ　　第1.15版で追加

形式　　BOOL CapAuthorizeVoidPreSales;

説明　　TRUE: AuthorizeVoidPreSalesメソッドは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　AuthorizeVoidPreSalesメソッド

CapCancelValue プロパティ

形式　**BOOL CapCancelValue;**

説明　TRUE：電子バリューへの操作の取消が有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapCardSensor プロパティ

形式　**LONG CapCardSensor;**

説明　カード検出センサの有無を示します。センサが存在する場合、検出状態が**DetectionStatus**プロパティに設定されます。

このプロパティは以下の値の論理和で設定されます。

値	意味
EVRW_CCS_ENTRY	挿入口センサあり
EVRW_CCS_DETECT	カード検出センサあり
EVRW_CCS_CAPTURE	保管庫センサあり

このプロパティは、**Open**メソッドにより初期化されます。

CapCashDepositプロパティ　　第1.15版で追加

形式　BOOL CapCashDeposit;

説明　デバイスがCashDepositメソッドによるチャージ機能を有するかどうかを示します。
TRUE: CashDepositメソッドは実装されています。
FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　CashDepositメソッド

CapCenterResultCodeプロパティ　　第1.15版で追加

形式　　BOOL CapCenterResultCode;

説明　　TRUE:CenterResultCodeプロパティは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　CenterResultCodeプロパティ

CapCheckCardプロパティ　　第1.15版で追加

形式　　BOOL CapCheckCard;

説明　　TRUE:CheckCardメソッドは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　CheckCardメソッド

CapDailyLogプロパティ　　第1.15版で追加

形式　LONG CapDailyLog;

説明　デバイスの認証ログ能力を示します

値	意味
EVRW_DL_NONE	このデバイスは認証ログ機能を一切持ちません。
EVRW_DL_REPORTING	このデバイスは中間計の機能のみ持ちます。中間計とは認証ログの読み出しは行いますが、認証ログの消去（精算）は行いません。
EVRW_DL_SETTLEMENT	このデバイスは精算機能のみ持ちます。 精算とは認証ログの読み出しと、認証ログの消去を行います。
EVRW_DL_ REPORTING_ SETTLEMENT	このデバイスは中間計と精算機能の両方を持ちます。

このプロパティは**Open**メソッドにより初期化されます。

参照　**DailyLog**プロパティ、**AccessDailyLog**メソッド

CapDetectionControl プロパティ

形式　**LONG CapDetectionControl;**

説明　カードの検出処理、カードの排出処理、カードの保管処理および、これらの処理をリーダライタから、もしくはアプリケーションから制御することが可能かどうかを示します。

サービスによっては、「決済要求」や「残高照会要求」など要求の粒度が大きく、カードの検出や排出といった細かい部分をアプリケーションから制御することが出来ない場合があります。そのような場合、検出、排出処理はリーダライタで制御されます。

このプロパティは以下の値の論理和で設定されます。

値	意味
EVRW_CDC_RWCONTROL	リーダライタで制御可能
EVRW_CDC_APPLICATIONCONTROL	アプリケーションで制御可能

このプロパティは、**Open**メソッドにより初期化されます。

CapElectronicMoney プロパティ

形式　**BOOL CapElectronicMoney;**

説明　TRUE：電子マネーサービスが有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapEnumerateCardServices プロパティ

形式　**BOOL CapEnumerateCardServices;**

説明　TRUE：カード内のサービスの列挙が有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapIndirectTransactionLog プロパティ

形式　**BOOL CapIndirectTransactionLog;**

説明　TRUE：取引ログは間接的なファイルとして取得されます。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapInstallmentsプロパティ　第1.15版で追加

形式　BOOL CapInstallments;

説明　TRUE: **AccessDailyLog**の結果として**DailyLog**プロパティに格納される項目のInstallmentsは実装されています。

FALSE: サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**DailyLog**プロパティ

CapLockTerminal プロパティ

形式　**BOOL CapLockTerminal;**

説明　TRUE：セキュリティロック設定が有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapLogStatus プロパティ

形式　**BOOL CapLogStatus;**

説明　TRUE：取引ログの状態確認が有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapMediumID プロパティ

形式　**BOOL CapMediumID;**

説明　TRUE：媒体識別子の指定が有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapMembershipCertificate プロパティ　*第1.14.1版で追加*

形式　**BOOL CapMembershipCertificate;**

説明　TRUE：会員証サービスが有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapPaymentDetailプロパティ　第1.15版で追加

形式　BOOL CapPaymentDetail;

説明　TRUE: PaymentDetailプロパティは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照　**PaymentDetail**プロパティ

CapPINDevice プロパティ　　第1.14版で追加

形式　**BOOL CapPINDevice;**

説明　PINの入力についてのデバイスの能力を示します。

値	説明
TRUE	電子バリューリーダライタはPINの入力を行うためのデバイスを持ちます。
FALSE	電子バリューリーダライタはPINの入力を行うためのデバイスを持ちません。

このプロパティは**Open**メソッド実行時に初期化されます。

CapPoint プロパティ

形式　**BOOL CapPoint;**

説明　TRUE：ポイントサービスが有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapSubtractValue プロパティ

形式　**BOOL CapSubtractValue;**

説明　TRUE：電子バリューの減算が有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapTaxOthersプロパティ　　第1.15版で追加

形式　　BOOL CapTaxOthers;

説明　　TRUE: AccessDailyLogの結果としてDailyLogプロパティに格納される項目のTaxOthersは実装されています。 このプロパティがFALSEの場合でも承認系メソッドの引数"*TaxOthers*"にはダミーで“０”をセットする必要があります。

FALSE: サポートしません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**DailyLog**プロパティ

CapTrainingMode プロパティ　　第1.14版で追加

形式　　**BOOL CapTrainingMode;**

説明　　トレーニングモードをサポートするかを示します。

値	説明
TRUE	練習モードはサポートされています。
FALSE	練習モードはサポートされていません。

このプロパティは**Open**メソッド実行時に初期化されます。

CapTransaction プロパティ

形式　　**BOOL CapTransaction;**

説明　　TRUE：一括処理が有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapTransactionLog プロパティ

形式　**BOOL CapTransactionLog;**

説明　TRUE：取引ログが有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapTransactionNumberプロパティ　第1.15版で追加

形式　BOOL CapTransactionNumber;

説明　TRUE: TransactionNumberプロパティは実装されています。

FALSE:サポートしません。

このプロパティは、Openメソッドにより初期化されます。

参照 TransactionNumberプロパティ

CapUnlockTerminal プロパティ

形式　**BOOL CapUnlockTerminal;**

説明　TRUE：セキュリティロック解除が有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapUpdateKey プロパティ

形式　**BOOL CapUpdateKey;**

説明　TRUE：鍵情報の更新が有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapVoucher プロパティ

形式　**BOOL CapVoucher;**

説明　TRUE：クーポンチケットサービスが有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CapWriteValue プロパティ

形式　**BOOL CapWriteValue;**

説明　TRUE：電子バリューの書き込みが有効です。
FALSE：無効です。

このプロパティは、**Open**メソッドにより初期化されます。

CardCompanyID プロパティ　　第1.15版で追加

形式　**BSTR CardCompanyID;**

説明　承認結果としての「カード会社ID」または「金融機関コード（支店コード含む）」を通知します。 桁数はデバイスに依存します。

このプロパティは、**Open**メソッドにより空の文字列に初期化されます。

CardServiceList プロパティ　*第1.14.1版で変更*

形式　**BSTR CardServiceList;**

説明　**EnumerateCardServices**メソッドで認識された、カードで利用可能なサービスが列挙されます。

サービスを列挙する文字列は、それぞれサービスを識別する任意の文字列をカンマによって連結した文字列（ＣＳＶ文字列）です。

たとえばA電子マネーサービスを識別する文字列が「MoneyA」、Ｂ電子ポイントサービスを識別する文字列が「PointB」であるとき、**CardServiceList**プロパティは”MoneyA, PointB”となります。

サービスがバリエーションを持つ場合の指定（第1.14.1版で追加）
サービスがいくつかのバリエーションを持つ場合、以下のルールでサービス識別文字列が設定されます。
“サービス名[:バリエーション名[:付加情報]]”
サービス名は省略されませんが、サービス名とバリエーション名の間のセパレータとなる “:”と、それ以降のバリエーション名、付加情報、バリエーション名とパラメータの間のセパレータとなる “:” は、省略されることがあります。また、サービス名、バリエーション名、付加情報には”,”と”:”を使うことはできません。

例）サービス”XYZCustomerPoint”はバリエーションとして” ABCStore”と”DEFShop”の２つのバリエーションを提供しており、カードにどちらのサービスでも有効であれば、**CardServiceList**プロパティには”XYZCustomerPoint:ABCStore, XYZCustomerPoint:DEFShop”が設定される。

CenterResultCodeプロパティ　第1.15版で追加

形式　BSTR CenterResultCode;

説明　各承認メソッドの結果として「承認機関からのコード」を格納します。実際格納されるコードは承認機関に確認してください。

このプロパティは、**Open**メソッドにより空の文字列に初期化されます。

CurrentService プロパティ　R/W　　*第1.14.1版で変更*

形式　**BSTR CurrentService;**

説明　現在選択され、有効であるサービスを識別する文字列が設定されます。

ReaderWriterServiceListプロパティに列挙されている文字列が設定可能です。

空文字列を設定するといずれのサービスも選択されていない状態となります。この状態は、実装により、デバイスであるリーダライタに対して処理を行うことが出来る状態を示します。

CurrentServiceプロパティに有効な文字列が設定されると、電子バリューリーダライタで該当のサービスが選択されます。

サービスオブジェクトがサブサービスをサポートしているとき、メソッドの実行やプロパティの設定は、プロパティが示すサービスのサブサービスに対して行われます。また、プロパティが示すサービスのサブサービスから発生したイベントだけがアプリケーションに通知されます。

サービスがバリエーションを持つ場合の指定（第1.14.1版で追加）
サービスがいくつかのバリエーションを持つ場合、以下のルールでサービス識別文字列を指定することができます。
“サービス名[:バリエーション名[:パラメータ]]”
サービス名の指定は必須ですが、サービス名とバリエーション名の間のセパレータとなる “:”と、それ以降のバリエーション名、パラメータ、バリエーション名とパラメータの間のセパレータとなる “:” は、省略可能です。また、サービス名、バリエーション名、パラメータには”,”と”:”を使うことはできません。

例）サービス”XYZCustomerPoint”はバリエーションとして”ABCStore”と”DEFShop”の２つのバリエーションを提供している。このとき、**ReaderWriterServiceList**プロパティには”XYZCustomerPoint: ABCStore, XYZCustomerPoint:DEFShop”が設定されている。
XYZCustomerPointサービスの”ABCStore”バリエーションにポイントを加算する場合には”XYZCustomerPoint:ABCStore”と**CurrentService**プロパティへ指定する。

DailyLogプロパティ　　第1.16.1版で更新

形式　　BSTR DailyLog;

説明　　AccessDailyLogメソッドの結果を格納します。データは１取引ごとにCR（13）＋LF（１０）で区切られて、各値はASCIIコードで格納されます。 各取引の詳細なデータは“,”（４４）で区切られています。

このプロパティは、Openメソッドにより空の文字列に初期化されます。

１取引データの詳細は以下順番の通りです。 また対応するプロパティとCapプロパティも併記します。

順番	項目	対応するプロパティ	対応する**Cap**プロパティ
1	カード会社ID/ 金融機関コード（支店コード含む）	**CardCompanyID**	なし
2	取引種別	**TransactionType**	なし
3	センター処理日時 注1)	なし	なし
4	センター処理通番 注３）	**TransactionNumber**	**CapTransactionNumber**
5	支払種別	**PaymentCondition**	なし
6	伝票番号	**SlipNumber**	なし
7	センター承認番号	**ApprovalCode**	なし
8	利用日 注１、注５）	なし	なし
9	会員番号/口座番号	**AccountNumber**	なし
１０	金額 注４)	**Authorize**系メソッドの引数の*Amount*または実際に承認された金額	なし
１１	税･その他 注３）	**Authorize**系メソッドの引数の*TaxOthers*	**CapTaxOthers**
１２	分割回数 注３）	なし	**CapInstallments**
１３	追加データ 注２）	**AdditionalSecurityInformation**	**CapAdditionalSecurityInformation**

注

1）フォーマット

項目	フォーマット
センター処理日時	YYYYMMDDHHMMSS
利用日	MMDD

デバイスによっては秒まで内部時計で持っていない場合があります。 その場合、センター処理日時の秒には“００”が書き込まれます。

２）追加データ

デバイスがベンダー固有に持っているデータを格納する領域です。 これらを使用することにより項目にないデータをアプリケーションが受取ることが可能になります。 ここに格納されるデータはデバイスやカードに依存するため、このデータは移植性を考慮した開発においては極力使用を避けるべきです。

３）その該当する**Cap**プロパティがFALSEの場合

デバイスによって該当するデータが無く、**Cap**プロパティがFALSEになります。 その場合、その項目が無く次の区切りのコンマがすぐにきます。 例えば「金額」が１２３４円で、「税・その他」が無く、「分割回数」が２の場合「1234,,２」となります。　これにより、**Cap**プロパティに依存せずに、各データの位置が一義的に決まります。

４）金額

項目１１の「税・その他」がある場合(**CapTaxOthers**プロパティがTRUEの場合)でも、ない場合(FALSEの場合)でも常に「税・その他」を含みます。

もし通貨がJPY（日本円）で、実際の金額が12,345円である場合、64ビット整数値は「123450000」となり、文字列表現は「12345.00」または「12345.0000」あるいは「12345」または「123450000」となります。もし通貨がUSD（米ドル）で、実際の金額が1,234.56ドルである場合、64ビット整数値は「12345600」となり、文字列表現は「1234.56」または「1234.5600」あるいは「12345600」となります。

５）利用日

マニュアル入力による承認後売上の利用日です。

例　　以下に例を示します。

項目	内容	意味
カード会社ID/ 金融機関コード（支店コード含む）	102	JCB
取引種別	**EVRW_PC_TRANSACTION_SALES**	売上
センター処理日時	19980116134530	1998年1月16日 13時45分30秒
センター処理通番	123456	123456
支払種別	**EVRW_PC_PAYMENT_INSTALLMENT_1**	分割1
伝票番号	12345	12345
センター承認番号	0123456	0123456
利用日	なし	なし
会員番号/口座番号	1234123412341234	1234-1234-1234-1234
金額	JPY（日本円） 12345.00 or 12345.0000 or 12345 or 123450000	12345円
	USD（米ドル） 12345.00 or 12345.0000 or 123450000	12345.00 USD
税・その他	なし	なし
分割回数	2	2回
追加データ	12345678	固有情報

上の例で実際に**DailyLog**に格納されるデータは以下の通りです。

```
102,10,19980116134530,123456,61,12345,0123456,,1234123412341234,12345,,2,12345678[CR][LF]
```

電子マネー決済デバイスでは、締め処理用通信装置を備えていない電子マネー決済デバイスが締め処理を行った結果の取引ログが設定されます。

取引ログの内容がセンター以外に対して暗号化されているデバイスがあるかもしれません。その場合、プロパティにはバイナリデータが設定されるため、**BinaryConversion**プロパティをOPOS_BC_NONE以外の値に設定してアクセスする必要があります。

参照　**CapDailyLog**プロパティ、**BinaryConversion**プロパティ、**AccessDailyLog**メソッド

DetectionControl プロパティ　R/W

形式　**BOOL DetectionControl;**

説明　TRUE：**BeginDetection/EndDetection**メソッドによるカードの検出処理、**BeginRemoval/EndRemoval**メソッドによるカード排出処理をアプリケーションで制御します。
CapDetectionControlプロパティに、EVRW_CDC_APPLICATIONCONTROLが指定されているとき、設定可能です。

FALSE：カードの検出、排出処理をアプリケーションから制御しません。アプリケーションからの**BeginDetection/EndDetection**メソッド、**BeginRemoval/EndRemoval**メソッドの呼び出しは無効です。
CapDetectionControlプロパティに、EVRW_CDC_RWCONTROLが指定されているとき、設定可能です。

DetectionStatus プロパティ

形式　**LONG DetectionStatus;**

説明　電子バリューリーダライタが認識しているカードの状態が設定されます。

値	意味
EVRW_DS_NOCARD	カードなし カード検出センサがカードを未検出
EVRW_DS_DETECTED	カードあり カード検出センサがカードを検出
EVRW_DS_ENTERED	挿入口にカードあり 挿入口センサがカードを検出
EVRW_DS_CAPTURED	保管庫にカードあり 保管庫センサがカードを検出

ExpirationDate プロパティ

形式　**BSTR ExpirationDate;**

説明　カードの満期日、有効期限を設定します。

YYYYMMDDの形式で設定されます。

SetParameterInfomationメソッドおよび**RetrieveResultInformation**メソッドのタグに同様の情報を持つため、このプロパティを使用せずにタグの情報を利用することを推奨します。

LastUsedDate プロパティ

形式　**BSTR LastUsedDate;**

説明　カードの最終利用日時を設定します。

YYYYMMDDHHMMSSの形式で設定されます。

SetParameterInfomationメソッドおよび**RetrieveResultInformation**メソッドのタグに同様の情報を持つため、このプロパティを使用せずにタグの情報を利用することを推奨します。

LogStatus プロパティ

形式　**LONG LogStatus;**

説明　取引ログの状態を示します。

値	意味
EVRW_LS_OK	取引ログは充分な空き容量があります。
EVRW_LS_NEARFULL	取引ログはフルに近い状態です。
EVRW_LS_FULL	取引ログは一杯です。

このプロパティは初期化された後、デバイスがイネーブルの間は最新の状態となります。

取引ログが一杯になると、デバイスによっては処理を行うことが出来なくなるかもしれません。

MediumID プロパティ　R/W　　第1.14版で変更

形式　**BSTR MediumID;**

説明　カードの媒体識別子が設定されます。

媒体識別子はカードを一意に特定するための情報（製造番号など）であり、どのような内容が設定されるのかはカードの実装に依存します。

以下のメソッドは、プロパティに指定された媒体識別子を持つカードに対して処理を行います。

BeginDetection
ReadValue
WriteValue
AddValue
SubtractValue
CancelValue

メソッド発行前に、プロパティに媒体識別子を設定することで、操作対象となるカードをアプリケーションが明示的に指定することができます。プロパティを空文字列に指定することは、不特定のカードに対する操作を意味します。

メソッドが成功すると、操作対象となったカードの媒体識別子が設定されます。

SetParameterInfomationメソッドおよび**RetrieveResultInformation**メソッドのタグに同様の情報を持つため、このプロパティを使用せずにタグの情報を利用することを推奨します。

PaymentConditionプロパティ　　第1.15版で追加

形式　LONG PaymentCondition;

説明　各承認メソッドの結果として「支払種別」を格納します。

このプロパティには、次に示す値の内の一つが設定されます。詳細な支払方法は**PaymentDetail**を参照してください。

値	意味
EVRW_PC_PAYMENT_LUMP	一括
EVRW_PC_PAYMENT_BONUS_1	ボーナス１
EVRW_PC_PAYMENT_BONUS_2	ボーナス２
EVRW_PC_PAYMENT_BONUS_3	ボーナス３
EVRW_PC_PAYMENT_BONUS_4	ボーナス４
EVRW_PC_PAYMENT_BONUS_5	ボーナス５
EVRW_PC_PAYMENT_INSTALLMENT_1	分割１
EVRW_PC_PAYMENT_INSTALLMENT_2	分割２
EVRW_PC_PAYMENT_INSTALLMENT_3	分割３
EVRW_PC_PAYMENT_BONUS_COMBINATION_1	ボーナス併用１
EVRW_PC_PAYMENT_BONUS_COMBINATION_2	ボーナス併用２
EVRW_PC_PAYMENT_BONUS_COMBINATION_3	ボーナス併用３
EVRW_PC_PAYMENT_BONUS_COMBINATION_4	ボーナス併用４
EVRW_PC_PAYMENT_REVOLVING	リボルビング
EVRW_PC_PAYMENT_DEBIT	デビットカード
EVRW_PC_PAYMENT_ELECTRONIC_MONEY	電子マネー

このプロパティは、**Open**メソッドによりゼロに初期化されます。

参照　**PaymentDetail**プロパティ

PaymentDetailプロパティ　　第1.16.1版で更新

形式　BSTR PaymentDetail;

説明　各承認メソッドの結果として「支払種別の詳細」を格納します。支払の詳細は各支払種別（ PaymentCondition ）により異なります。 それぞれのデータはASCIIコードで格納され、値は“,”（コンマ）で区切られます。 「ボーナス４」、「ボーナス５」、「分割２」、「ボーナス併用３」および「ボーナス併用４」は最大で６回までで、もし６回以下ならばそれまでのデータのみが格納されます。 “NULL”とはデータが何も格納されないという意味であり、長さゼロの**BSTR**となります。

一括

NULL

ボーナス１

NULL

ボーナス２

ボーナス回数

ボーナス３

ボーナス月

ボーナス４

ボーナス回数、ボーナス１回目月、ボーナス２回目月、ボーナス３回目月、ボーナス4回目月、ボーナス５回目月、ボーナス６回目月　　（最大６回)

ボーナス５

ボーナス回数、ボーナス１回目月、１回目金額、ボーナス２回目月、２回目金額、ボーナス３回目月、３回目金額、ボーナス４回目月、４回目金額、ボーナス５回目月、５回目金額、ボーナス６回目月、６回目金額、　(最大６回)

分割１

開始月、分割回数

分割２

開始月、分割回数、１回目金額、２回目金額、３回目金額、４回目金額、５回目金額、６回目金額(最大６回)

分割３

開始月、分割回数、初回金額

ボーナス併用１

開始月、分割回数

ボーナス併用２

開始月、分割回数、ボーナス金額

ボーナス併用３

開始月、分割回数、ボーナス回数、ボーナス１回目月、ボーナス２回目月、ボーナス３回目月、ボーナス４回目月、ボーナス５回目月、ボーナス６回目月 (最大６回)

ボーナス併用４

開始月、分割回数、ボーナス回数、ボーナス１回目月、ボーナス１回目金額、ボーナス２回目月、ボーナス２回目金額、ボーナス３回目月、ボーナス３回目金額、ボーナス４回目月、ボーナス４回目金額、ボーナス５回目月、ボーナス５回目金額、ボーナス６回目月、ボーナス６回目金額(最大６回)

リボルビング払い

NULL

デビットカード払い

NULL

電子マネー

NULL

これらの支払区分、名称は各CATデバイスによって異なります。 次ページからそのCATデバイス別可能支払区分と用語の一覧を示します。 一部にはOPOS用語と各CATでの用語が異なるため注意が必要です。

このプロパティは、**Open**メソッドにより空の文字列に初期化されます。

CAT別支払区分名称、方法一覧

支払方法	入力項目	コード	CAT名称	CAT(旧CAT)	G-CAT	JET-S	SG-CAT	Master-T
			カード	不特定	不特定	ＪＣＢ	ＶＩＳＡ	ＭＡＳＴＥＲ
			OPOS用語	各社の用語				
一括	（なし）	10	一括	同左	同左	同左	同左	同左
ボーナス	（なし）	21	ボーナス１	同左	同左	同左	同左	同左
	ボーナス回数	22	ボーナス２	同左	同左	同左	同左	同左
	ボーナス月	23	ボーナス３	同左	－	－	同左	同左
	ボーナス回数 ボーナス月(1) ボーナス月(2) ボーナス月(3) ボーナス月(4) ボーナス月(5) ボーナス月(6)	24	ボーナス４	同左	ボーナス３	ボーナス３	同左 （但し、ボーナス月入力は２回まで）	同左
	ボーナス回数 ボーナス月(1) ボーナス金額(1) ボーナス月(2) ボーナス金額(2) ボーナス月(3) ボーナス金額(3) ボーナス月(4) ボーナス金額(4) ボーナス月(5) ボーナス金額(5) ボーナス月(6) ボーナス金額(6)	25	ボーナス５	同左	－	－	－	同左
分割	支払開始月 分割回数	61	分割１	同左	同左	同左	同左	同左

	支払開始月 分割回数 分割金額(1) 分割金額(2) 分割金額(3) 分割金額(4) 分割金額(5) 分割金額(6)	62	分割２	同左	－	－	－	－
	支払開始月 分割回数 初回金額	63	分割３	同左	分割２	分割２	－	分割２
ボ併	支払開始月 分割回数	31	ボーナス 併用１	同左	同左	同左	同左	同左
	支払開始月 分割回数 ボーナス金額	32	ボーナス 併用２	同左	－	－	同左	同左
	支払開始月 分割回数 ボーナス回数 ボーナス月(1) ボーナス月(2) ボーナス月(3) ボーナス月(4) ボーナス月(5) ボーナス月(6)	33	ボーナス 併用３	同左	－	－	同左 （但し、ボーナス月入力は２回まで）	同左

	支払開始月 分割回数 ボーナス回数 ボーナス月(1) ボーナス金額(1) ボーナス月(2) ボーナス金額(2) ボーナス月(3) ボーナス金額(3) ボーナス月(4) ボーナス金額(4) ボーナス月(5) ボーナス金額(5) ボーナス月(6) ボーナス金額(6)	34	ボーナス 併用４	同左	ボーナス 併用２	ボーナス 併用２	同左 （但し、ボーナス月、金額入力は２回まで）	同左
リボ	（なし）	80	リボルビング	同左	同左	同左	同左	同左
デビット	（なし）	110	デビットカード	対応機種のみ	対応機種のみ	対応機種のみ	対応機種のみ	対応機種のみ

凡例：　同左：ＯＰＯＳ用語に同じ

―　：その支払区分が存在しない

参照　　**CapPaymentDetail**プロパティ、**DailyLog**プロパティ

PaymentMediaプロパティR/W　　第1.15版で追加

形式　LONG PaymentMedia;

説明　各承認メソッドの承認すべきメディアを指定します。

このプロパティには、各承認メソッドを発行する前に次に示す値の内の一つを設定します。指定無しとはデバイス側で指定することを意味します。

値	意味
EVRW_PM_MEDIA_UNSPECIFIED	指定無し
EVRW_PM_MEDIA_NONDEFINE	同上（第1.4版以前との互換性確保のため残すがEVRW_PM_MEDIA_UNSPECIFIEDを使用すること）
EVRW_PM_MEDIA_CREDIT	クレジット
EVRW_PM_MEDIA_DEBIT	デビット
EVRW_PM_MEDIA_ELECTRONIC_MONEY	電子マネー

このプロパティは、**Open**メソッドにより EVRW_PM_MEDIA_NONDEFINEに初期化されます。

PINEntry プロパティ R/W　　第1.14版で追加

形式　**LONG PINEntry;**

説明　CurrentService選択されているサービスについての、PINの入力についての能力を示します。

値	説明
EVRW_PIN_ENTRY_UNKNOWN	サービスが選択されていないため、PINの入力について判断できません。 **Open**メソッド実行直後や、**CurrentService**プロパティで有効なサービスが選択されていないときにこの値になります。
EVRW_PIN_ENTRY_NONE	PINの入力はサポートされません。
EVRW_PIN_ENTRY_INTERNAL	PINの入力が必要な場合、電子バリューリーダライタが持っているPIN入力デバイスでPINの入力が処理されます。
EVRW_PIN_ENTRY_EXTERNAL	電子バリューリーダライタはPIN入力デバイスを持っていません。PINの入力が必要な場合、外部のPINパッドデバイスでPINの入力を行い、PIN情報を連携する必要があります。

このプロパティは**Open**メソッド実行時、および**CurrentService**プロパティの変更時に初期化されます。

Point プロパティ　R/W　　第1.14版で変更

形式　　**CURRENCY Point;**

説明　　ポイントサービスにおいて、決済ポイントを設定します。

SetParameterInfomationメソッドおよび**RetrieveResultInformation**メソッドのタグに同様の情報を持つため、このプロパティを使用せずにタグの情報を利用することを推奨します。

ReaderWriterServiceList プロパティ　*第1.14.1版で変更*

形式　**BSTR ReaderWriterServiceList;**

説明　電子バリューリーダライタで利用可能なサービスが列挙されます。

サービスを列挙する文字列は、それぞれサービスを識別する任意の文字列をカンマによって連結した文字列（ＣＳＶ文字列）です。

たとえばA電子マネーサービスを識別する文字列が「MoneyA」、Ｂ電子ポイントサービスを識別する文字列が「PointB」であるとき、**ReaderWriterServiceList**プロパティは”MoneyA, PointB”となります。

サービスオブジェクトがサブサービスをサポートする場合、**Open**メソッドが成功すると、全てのサブサービスが提供するサービスが列挙されます。

サービスオブジェクトがサブサービスをサポートしない場合でも、１つの電子バリューリーダライタサービスオブジェクトが複数のサービスに対応するときには、**ReaderWriterServiceList**プロパティには複数のサービスが列挙されます。

サービスがバリエーションを持つ場合の指定（第1.14.1版で追加）
サービスがいくつかのバリエーションを持つ場合、以下のルールでサービス識別文字列が設定されます。
“サービス名[:バリエーション名[:付加情報]]”
サービス名は省略されませんが、サービス名とバリエーション名の間のセパレータとなる “:”と、それ以降のバリエーション名、付加情報、バリエーション名とパラメータの間のセパレータとなる “:” は、省略されることがあります。また、サービス名、バリエーション名、付加情報には”,”と”:”を使うことはできません。

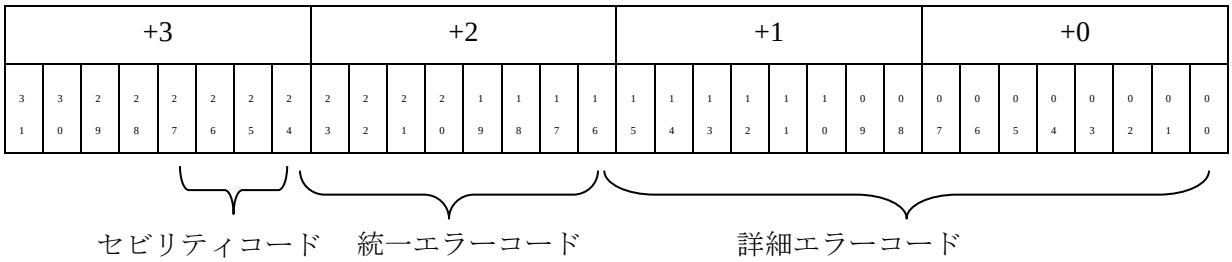
例）サービス”XYZCustomerPoint”はバリエーションとして” ABCStore”と”DEFShop”の２つのバリエーションを提供している。このとき、**ReaderWriterServiceList**プロパティには”XYZCustomerPoint:ABCStore, XYZCustomerPoint:DEFShop”が設定される。

ResultCodeExtended プロパティ　　*第1.14.1版で変更*

形式　　**LONG ResultCodeExtended;**

説明　　**ResultCode**がOPOS_E_EXTENDEDの場合、このプロパティには電子バリューリーダライタデバイス共通エラーコードが設定されます。

電子バリューリーダライタデバイス共通エラーコードは、ResultCodeExtendedプロパティの32ビット整数型に対して、以下のように割り当てます。



ビット位置	サイズ（ビット）	項目	説明
31 – 27	5	（未使用）	
26 – 24	3	セビリティコード	エラーの深刻度と、復旧の方法を示す
23 – 16	8	統一エラーコード	
15-0	16	詳細エラーコード	サービス、ベンダーに依存

セビリティコードは、デバイスをエラーからの復旧するために必要な行為を示します。

値	項目	内容	備考
0	NORMAL	復旧の必要なし	
1	BLOCKED	保守員による作業で復旧可能	場合によって交換が必要

2	RECOVERABLE	POSなどアプリケーションプログラムの判断で、必要に応じて条件を変えてリトライすることで復旧可能	カード待ちタイムアウト
3	RECOVERABLE_ASK_CARDHOLDER	お客様による確認・判断・是正で、条件を変えてリトライすることで復旧可能	残高不足、処理未了、金額超過
4	RECOVERABLE_ASK_OPERATOR	店員による確認・判断・是正で、条件を変えてリトライすることで復旧可能	端末のログフル、モード違い

統一エラーコードはエラーの分類を示します。

統一エラーコード	項目	内容
OPOS_EEVRW_ABORTED	POSからの取引の中止	POSからの取引中止要求がされ、処理を中止した
OPOS_EEVRW_DEFICIENT	残高不足エラー	
OPOS_EEVRW_DETECTION_TIMEOUT	メディア検出タイムアウトエラー	規定の時間にメディアを検出することができず、処理が実行できなかった
OPOS_EEVRW_HOST_CANNOT_CLOSE	決済センタ異常終了（閉局処理）	閉局処理で異常が発生して処理が正常終了しなかった。
OPOS_EEVRW_HOST_CANNOT_OPEN	決済センタ異常終了（開局処理）	開局処理で異常が発生して処理が正常終了しなかった。
OPOS_EEVRW_HOST_CANNOT_OPERATE	決済センタ異常終了（営業中処理）	営業中処理で異常が発生して処理が正常終了しなかった。
OPOS_EEVRW_HOST_REFUSAL	決済センタ拒否エラー	処理に決済センタによる承認が必要な場合、決済センタから承認拒否され、処理が実行できなかった
OPOS_EEVRW_IN_PROGRESS	取引中エラー	既に取引が処理中であり、処理が実行できなかった
OPOS_EEVRW_INVALID_MEDIUM	無効メディア検出エラー	ネガなどで無効になったメディアを検出し、処理が実行できなかった

OPOS_EEVRW_INVALID_MEDIUM_ABORTED	メディア異常	メディアで異常が発生して、処理が実行できなかった。
OPOS_EEVRW_INVALID_MEDIUM_ABORTED_EXISTS	メディア異常（重複）	メディアにサービスが既に存在するため、処理が実行できなかった。
OPOS_EEVRW_INVALID_MEDIUM_ABORTED_NOSERVICE	メディア異常(未登録)	メディアにサービスが登録されておらず、処理が実行できなかった。
OPOS_EEVRW_INVALID_MEDIUM_ABORTED_NOSPACE	メディア異常(空き容量なし)	メディアに空き容量がなく、処理が実行できなかった。
OPOS_EEVRW_INVALID_MEDIUM_EXPIRED	有効期限切れエラー	有効期限切れのメディアを検出し、処理が実行できなかった
OPOS_EEVRW_LOG_OVERFLOW	取引ログ超過エラー	取引ログが上限を超過するため、処理が実行できなかった
OPOS_EEVRW_MEDIUM_CANNOT_AUTHORIZE	メディア認証エラー	検出したメディアと認証を行ったが、認証に失敗し、処理が実行できなかった
OPOS_EEVRW_MESSAGE_FORMAT	電文エラー	電文の構成に不備があり、処理が実行できなかった。通信異常の場合を想定。
OPOS_EEVRW_OVERDEPOSIT	チャージ金額超過エラー	チャージ金額加算後の残高が上限を超過するため、処理が実行できなかった
OPOS_EEVRW_OVERDEPOSIT_TO_POINT	ポイント超過エラー	ポイント加算後のポイント数が上限ポイントを超過するため、処理が実行できなかった

OPOS_EEVRW_PAYMENT_RESTRICTION	決済制限エラー	決済取引に制限される要素（決済が制限された商品が含まれるなど）があるため、処理が実行できなかった
OPOS_EEVRW_RW_LOCKED	R／Wロック	R／Wはロックされており、処理が実行できなかった。
OPOS_EEVRW_RW_OUT	R／W障害エラー	R／Wに障害が発生しており、処理が実行できなかった
OPOS_EEVRW_RW_OUT_TEMPORARY_OUT	R／W障害エラー（復旧可能）	R／Wに障害が発生しており、処理が実行できなかった。ただし、開局処理等を実施することで復旧が可能。
OPOS_EEVRW_RW_OUT_TEMPORARY_OUT_NEED_TO_RESET	R／Wリセット要求	
OPOS_EEVRW_TRANSACTION_INCOMPLETE	処理未了エラー	決済処理中にメディアが離され、決済処理が成立したか不明の状態で処理が終了した
OPOS_EEVRW_UNREACHABLE_HOST	決済センタ接続エラー	処理に決済センタによる承認が必要な場合、決済センタに接続できなかったため、処理が実行できなかった

OPOS_EEVRW_UPOS114COMPATIBLE	UPOS1.14までの仕様でのエラーコードとの互換用	詳細エラーコードには、UPOS1.14までのResultCodeExtendedプロパティの仕様で定義されるエラーコードが設定される（OPOS_EEVRW_UPOS114COMPATIBLEの値は0であり、統一エラーコード0の場合にはセビリティコードは使われず、詳細エラーコードにUPOS1.14までで定義されているエラーコードが設定される）

詳細エラーコードは、デバイスやサービスオブジェクトにより定義が異なる、詳細なエラー状態を示すコードです。詳細エラーコードの内容は実装に依存します。

参照　**ResultCodeプロパティ**

SequenceNumber プロパティ

形式　**LONG SequenceNumber;**

説明　各メソッドの結果として「受信シーケンス番号」を格納します。 この番号はアプリケーションでメソッドの引数*SequenceNumber*との一致を確認する必要があります。

ServiceType プロパティ　*第1.16.1版で更新*

形式　**LONG ServiceType;**

説明　**CurrentService**プロパティで指定されているサービスのサービス種別が設定されます。**Cap**プロパティで示されるデバイスの能力とは異なり、本プロパティは選択されているサービスの能力を示します。

値	意味
EVRW_ST_ELECTRONIC_MONEY	電子マネーサービス
EVRW_ST_POINT	ポイントサービス
EVRW_ST_VOUCHER	クーポンチケットサービス
EVRW_ST_MEMBERSHIP	会員証
EVRW_ST_UNSPECIFIED	CurrentServiceプロパティでサービスが指定されていない
EVRW_ST_CAT	CATサービス

SettledAmount プロパティ　**第1.14版で変更**

形式　**CURRENCY SettledAmount;**

説明　電子マネーサービスにおいて、実決済金額を設定します。

SetParameterInfomationメソッドおよび**RetrieveResultInformation**メソッドのタグに同様の情報を持つため、このプロパティを使用せずにタグの情報を利用することを推奨します。

SettledPoint プロパティ

形式　**CURRENCY SettledPoint;**

説明　ポイントサービスにおいて、実決済ポイントを設定します。

SlipNumberプロパティ　第1.15版で追加

形式　BSTR SlipNumber;

説明　各承認メソッドの結果として「伝票番号」を格納します。

このプロパティは、**Open**メソッドで空の文字列に初期化されます。

TransactionNumberプロパティ　第1.15版で追加

形式　BSTR TransactionNumber;

説明　各承認メソッドの結果として「センター処理通番」を格納します。

このプロパティは、**Open**メソッドで空の文字列に初期化されます。

TrainingModeStateプロパティ　R/W　第1.14版で追加

形式　**LONG TrainingModeState;**

説明　EVRW_TM_TRUE:　メソッドが練習モードで実行されます。

EVRW_TM_FALSE: メソッドが通常モードで実行されます。

EVRW_TM_UNKNOWN
CurrentServiceプロパティでサービスが選択されていないため、状態が不定です。

戻り値　本プロパティ設定時、次の値のいずれかが**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	このプログラムは正常に設定されました。
OPOS_E_ILLEGAL	正常に設定されていません。練習モードはサポートされていません。**(CapTrainingMode**プロパティを参照**)**

TransactionLog プロパティ

形式　**BSTR TransactionLog;**

説明　**AccessLog**メソッドの結果取得された取引ログを格納します。

CapIndirectTransactionLogプロパティがTRUEの場合、**TransactionLog**プロパティは取引ログが格納されたファイルなどの位置を示すURLを示します。

取引ログの内容は、デバイスやサービスに依存します。

TransactionTypeプロパティ　第1.15版で追加

形式　LONG TransactionType;

説明　各承認メソッドの結果として「取引種別」を格納します。

このプロパティには、次に示す値の内の一つが設定されます。

値	意味
EVRW_TT_TRANSACTION_SALES	売上
EVRW_TT_TRANSACTION_VOID	取消
EVRW_TT_TRANSACTION_REFUND	返品
EVRW_TT_TRANSACTION_COMPLETION	承認後売上
EVRW_TT_TRANSACTION_PRESALES	事前予約
EVRW_TT_TRANSACTION_CHECKCARD	カードチェック
EVRW_TT_TRANSACTION_VOIDPRESALES	事前予約の取消
EVRW_TT_TRANSACTION_CASHDEPOSIT	チャージ

このプロパティは、**Open**メソッドでゼロに初期化されます。

VoucherID プロパティ　R/W

形式　**BSTR VoucherID;**

説明　クーポン／チケットサービスにおいて、クーポン／チケットのＩＤを設定します。

クーポン／チケットは、権利の種類を示す識別子と個数の対で表現します。

識別子 001 のチケット６枚は、「001:6」という文字列で表されます。「:」は、識別子と枚数のセパレータです。

SetParameterInfomationメソッドおよび**RetrieveResultInformation**メソッドのタグに同様の情報を持つため、このプロパティを使用せずにタグの情報を利用することを推奨します。

VoucherIDList プロパティ　R/W

形式　**BSTR VoucherIDList;**

説明　クーポン／チケットサービスにおいて、列挙されたクーポン／チケットのＩＤを設定します。

識別子 001 のチケット６枚、識別子 002 のチケット１枚、識別子 034 のチケット２枚を保持している場合、「001:6, 002:1, 034:2」という形式のＣＳＶ文字列で表現します。「,」は、複数の権利を保持している場合のセパレータです。

SetParameterInfomationメソッドおよび**RetrieveResultInformation**メソッドのタグに同様の情報を持つため、このプロパティを使用せずにタグの情報を利用することを推奨します。

メソッド

AccessDailyLog メソッド　　第1.15版で追加

形式　LONG AccessDailyLog (LONG *SequenceNumber*, LONG *Type*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	認証ログ取得のためのシーケンス番号
Type	認証ログが中間計か精算かを指定します。
Timeout	デバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　認証ログをデバイスから取得します。

*SequenceNumber*によって指定された処理として認証ログを取得し**DailyLog**に格納します。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

*type*は次に示す値のいずれかを設定し、認証ログのタイプ（中間計か精算のどちらか）を指定します。 指定できるタイプは**CapDailyLog**に従います。

電子マネー決済デバイスでは、認証ログをデバイスから取得します。

電子マネー決済デバイスの取引ログを締めます。電子マネー決済デバイスが締め処理用通信装置をもっている場合には、取引ログは電子マネー決済端末から直接センターに送信されるでしょう。電子マネー決済デバイスが締め処理用通信装置をもたず、取引ログをPOSアプリケーションによってセンターに送信する必要がある場合には、取引ログは**DailyLog**プロパティに格納されます。

値	意味
EVRW_ADL_DL_REPORTING	認証ログとして中間計を実行します。
EVRW_ADL_DL_SETTLEMENT	認証ログとして精算を実行します。 電子マネー決済デバイスの取引ログを締めます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に認証ログ取得処理が行われました。
OPOS_E_ILLEGAL	無効な*Type*あるいは*Timeout*パラメータが指定されました。又は認証ログ取得はサポートされていません。(**CapDailyLog**プロパティを参照)
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**CapDailyLog**プロパティ、**DailyLog**プロパティ、**ErrorEvent**イベント

AccessData メソッド　*第1.14.1版で追加*

形式　**LONG AccessData (LONG** *DataType*, **LONG*** *pData*, **BSTR*** *pString*);

パラメータ	説明
DataType	更新を行うデータのタイプを指定します。値は以下の通りです。

値	意味
EVRW_AD_KEY	鍵情報
EVRW_AD_NEGATIVE_LIST	ネガティブリスト情報
EVRW_AD_OTHERS	そのほかの情報。

pData	数値データのポインタ。値は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。値はサービスオブジェクトによって変化します。

説明　電子バリューリーダライタから、取引ログ以外のログ情報を取得します。

電子バリューリーダライタが**AccessLog**メソッドで取得可能な取引ログのほかに、取得可能なログ情報を持つ場合にサポートされます。

取引ログ以外のログ情報の内容は、実装に依存します。

非同期処理でこのメソッドが起動されると、サービスオブジェクトはEVRW_TE_NOTIFY_PROGRESSに1から100を加えた進捗のパーセント値で**TransitionEvent**を通知し、処理の状態をアプリケーションに報告します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
その他	**ResultCode**の項目を参照してください。

AccessLog メソッド　　*第1.14.1版で変更*

形式　**LONG AccessLog (LONG** *SequenceNumber***, LONG** *Type*, **LONG** *Timeout***);**

パラメータ	説明
SequenceNumber	取引ログ取得のためのシーケンス番号
Type	中間計か精算かを指定します。
Timeout	デバイスから応答がくるまでの最大待ち時間 (ミリ秒単位) です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　取引ログをデバイスから取得します。

*SequenceNumber*によって指定された要求として取引ログを取得し**TransactionLog**プロパティに格納します。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

サービスが取引の精算処理を必要とする場合には、このメソッドで行います。

実装により、精算処理を行うことで取引ログがクリアされることがあります。一方、中間計は、取引ログ取得を行っても、デバイス内の取引ログがクリアされないことを保障します。

実装により、デバイスが通信装置を持ち、取引ログがアプリケーションに渡されず、直接決済センタに送信される場合もあります。

*type*は次に示す値のいずれかを設定し、取引ログのタイプ（中間計か精算のどちらか）を指定します。

値	意味
EVRW_DL_REPORTING	取引ログを、中間計として取得します。
EVRW_DL_SETTLEMENT	取引ログを、精算処理として取得します。 実装によりますが、デバイスの取引ログを確定、消去します。

進捗の通知（第1.14.1版で追加）
非同期処理でこのメソッドが起動されると、サービスオブジェクトはEVRW_TE_NOTIFY_PROGRESSに1から100を加えた進捗のパーセント値で**TransitionEvent**を通知し、処理の状態をアプリケーションに報告します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に取引ログ取得処理が行われました。
OPOS_E_ILLEGAL	無効な*Type*あるいは*Timeout*パラメータが指定されました。又は取引ログ取得はサポートされていません。
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ResultCodeExtended**の項を参照してください。
OPOS_E_BUSY	非同期処理中でコマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

ActivateEVService メソッド　　*第1.14.1版で追加*

形式　**LONG ActivateEVService (LONG*** *pData*, **BSTR*** *pString*);

パラメータ	説明
pData	数値データのポインタ。値は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。値はサービスオブジェクトによって変化します。

説明　電子バリューリーダライタの活性化処理を行います。

デバイスもしくはサービスが活性化処理機能を持つ場合にサポートされます。

活性化処理は、デバイスやサービスを新たにインストールするときやデバイスを初期設置したとき、工場出荷時に無効化した機能を有効化するときに行われる、利用可能にするための初期処理です。処理の内容やパラメータの内容は実装に依存します。

進捗の通知（第1.14.1版で追加）
非同期処理でこのメソッドが起動されると、サービスオブジェクトはEVRW_TE_NOTIFY_PROGRESSに1から100を加えた進捗のパーセント値で**TransitionEvent**を通知し、処理の状態をアプリケーションに報告します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	デバイスは活性化機能を持ちません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ResultCodeExtended**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

ActivateService メソッド

形式　**LONG ActivateService (LONG*** *pData*, **BSTR*** *pString*);

説明　電子バリューリーダライタの活性化処理を行います。

デバイスが活性化処理機能を持つ場合にサポートされます。

活性化処理は、デバイスやサービスをインストールするときや初期設置したときに行われる、利用可能にするための初期処理です。処理の内容やパラメータの内容は実装に依存します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	デバイスは活性化機能を持ちません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

AddValue メソッド

形式　**LONG AddValue (LONG** *SequenceNumber***, LONG** *Timeout***);**

パラメータ	説明
SequenceNumber	シーケンス番号
Timeout	デバイスから応答がくるまでの最大待ち時間 (ミリ秒単位) です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　カードへ電子バリューの追加を行います。

*SequenceNumber*によって指定された要求としてカードへ電子バリューの追加を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又はサポートされていません。
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

AuthorizeCompletion メソッド　　第1.15版で追加

形式　LONG AuthorizeCompletion (LONG *SequenceNumber*, CURRENCY *Amount*, CURRENCY *TaxOthers*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	承認のためのシーケンス番号
Amount	承認のための売上金額（税、その他を含まない金額）
TaxOthers	承認のための税、その他金額
Timeout	デバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　承認後売上を行います。

*SequenceNumber*によって指定された承認として、 *Amount*、*TaxOthers*、の承認後売上を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に承認処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又は承認後売上はサポートされていません。(**CapAuthorizeCompletion**プロパティ参照)
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**CapAuthorizeCompletion**プロパティ、**ErrorEvent**イベント

AuthorizePreSales メソッド　　第1.15版で追加

形式　LONG AuthorizePreSales (LONG *SequenceNumber*, CURRENCY *Amount*, CURRENCY *TaxOthers*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	承認のためのシーケンス番号
Amount	承認のための売上金額（税、その他を含まない金額）
TaxOthers	承認のための税、その他金額
Timeout	デバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　事前予約を行います。

*SequenceNumber*によって指定された承認として、 *Amount*、*TaxOthers*、の事前予約を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に承認処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又は事前予約はサポートされていません。(**CapAuthorizePreSales**プロパティ参照)
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**CapAuthorizePreSales**プロパティ、**ErrorEvent**イベント

AuthorizeRefund メソッド　　第1.15版で追加

形式　LONG AuthorizeRefund (LONG *SequenceNumber*, CURRENCY *Amount*, CURRENCY *TaxOthers*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	承認のためのシーケンス番号
Amount	承認のための売上金額（税、その他を含まない金額）
TaxOthers	承認のための税、その他金額
Timeout	デバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　返品承認を行います。

*SequenceNumber*によって指定された承認として、 *Amount*、*TaxOthers*、の返品承認を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に承認処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又は返品承認はサポートされていません。(**CapAuthorizeRefund**プロパティ参照)
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**CapAuthorize**Refundプロパティ、**ErrorEvent**イベント

AuthorizeSales メソッド　　第1.15版で追加

形式　LONG AuthorizeSales (LONG *SequenceNumber*, CURRENCY *Amount*, CURRENCY *TaxOthers*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	承認のためのシーケンス番号
Amount	承認のための売上金額（税、その他を含まない金額）
TaxOthers	承認のための税、その他金額
Timeout	デバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　通常売上承認を行います。

*SequenceNumber*によって指定された承認として、 *Amount*、*TaxOthers*、の通常売上承認を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に承認処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**ErrorEvent**イベント

AuthorizeVoid メソッド　　第1.15版で追加

形式　LONG AuthorizeVoid (LONG *SequenceNumber*, CURRENCY *Amount*, CURRENCY *TaxOthers*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	承認のためのシーケンス番号
Amount	承認のための売上金額（税、その他を含まない金額）
TaxOthers	承認のための税、その他金額
Timeout	デバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　売上取消の承認を行います。

*SequenceNumber*によって指定された承認として、 *Amount*、*TaxOthers*、の取消承認を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に承認処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又は売上取消の承認はサポートされていません。(**CapAuthorizeVoid**プロパティ参照)
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**CapAuthorize**Voidプロパティ、ErrorEventイベント

AuthorizeVoidPreSales メソッド　　第1.15版で追加

形式　LONG AuthorizeVoidPreSales (LONG *SequenceNumber*, CURRENCY *Amount*, CURRENCY *TaxOthers*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	承認のためのシーケンス番号
Amount	承認のための売上金額（税、その他を含まない金額）
TaxOthers	承認のための税、その他金額
Timeout	デバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　事前予約の取消承認を行います。

*SequenceNumber*によって指定された承認として、 *Amount*、*TaxOthers*、の事前予約の取消承認を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

事前予約の取消承認が実装されていないOPOSコントロール／デバイスでは通常の取消で事前予約の取消承認ができる場合があります。 各OPOSコントロール／デバイスのマニュアルを参照してください。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に承認処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又は事前予約の取消承認はサポートされていません。(**CapAuthorizeVoidPreSales**プロパティ参照)
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

参照　**CapAuthorizeVoidPreSales**プロパティ、**ErrorEvent**イベント

BeginDetection メソッド

形式　**LONG BeginDetection (LONG** *Type***, LONG** *Timeout***);**

*Timeout*パラメータにミリ秒単位で設定された値の間、デバイスの処理を待ちます。*Timeout*パラメータ値が0の場合、デバイスをカード検出モードにし、即時に終了します。OPOS_FOREVER（-1）の場合、カードが検出されるかエラーが発生するまで待ち続けます。

*Type*パラメータは、検出するカードの種別を指定します。指定可能な値は以下の通りです。

値	内容
EVRW_BD_ANY	検出するカードの内容は問いません
EVRW_BD_SPECIFIC	メソッド呼び出しの時点で**MediumID**プロパティに設定されている識別子に該当するカードのみ検出します

説明　カードの検出、取り込みを開始する時に呼び出します。

接触型ＩＣカード、非接触ＩＣカードの双方をサポートします。

呼び出されると、デバイスをカード検出モードにし、カード検出の準備をします。このメソッドは、カード検出モードを終了させる**EndDetection**メソッドとセットで呼び出します。

カードの検出に成功すると、OPOS_SUCCESSを返します。

カードが検出されずタイムアウトすると、OPOS_E_TIMEOUTを返しますが、カード検出モードは維持されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	カードが正常に検出されました。
OPOS_E_BUSY	非同期処理中で実行できません。
OPOS_E_ILLEGAL	無効な*Timeout*値が指定されました。
OPOS_E_TIMEOUT	カードが検出されずに、設定されたタイムアウト値が経過しました。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

BeginRemoval メソッド

形式　**LONG BeginRemoval (LONG** *Timeout***);**

*Timeout*パラメータにミリ秒単位で設定された値の間、デバイスの処理を待ちます。パラメータ値が0の場合、カードの排出を実行し、即時に終了します。OPOS_FOREVER（-1）の場合、カードを排出し、カードが挿入口から抜き取られるかエラーが発生するまで待ち続けます。

説明　カードの排出処理をする時に呼び出します。

接触型ＩＣカードでは、このメソッドが呼び出されると、デバイスをカード排出モードにし、カードを排出します。

非接触型ＩＣカードデバイスの場合、このメソッドは必ずしも実装される必要はなく、常に正常終了します。

接触型ＩＣカードデバイスで、挿入口にセンサーを有するタイプは、カードがデバイスから抜き取られることで正常終了します。

非接触ＩＣカードデバイスで、挿入口にセンサーを有しないタイプは、排出処理が実行されることで、正常終了します。

このメソッドは、**EndRemoval**メソッドとセットで呼び出します。

カードの排出に成功すると、OPOS_SUCCESSを返します。

カードが排出されずタイムアウトすると、OPOS_E_TIMEOUTを返しますが、カード排出モードは維持されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	カードが抜き取られました。
OPOS_E_BUSY	非同期処理中で実行できません。
OPOS_E_ILLEGAL	無効な*Timeout*値が指定されました。
OPOS_E_TIMEOUT	カードが適切に排出されずに、設定されたタイムアウト値が経過しました。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

CancelValue メソッド

形式　**LONG CancelValue (LONG** *SequenceNumber***, LONG** *Timeout***);**

パラメータ	説明
SequenceNumber	シーケンス番号
Timeout	デバイスから応答がくるまでの最大待ち時間 (ミリ秒単位) です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　カードへ要求した電子バリュー操作の取消を行います。

*SequenceNumber*によって指定された要求としてカードへ電子バリューの操作の取消を行います。

取消の対象となる操作は**ApprovalCode**プロパティに設定された決済番号で識別します。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又はサポートされていません。
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

CaptureCard メソッド

形式　　**LONG CaptureCard ();**

説明　　カード挿入口などに置き忘れられたカードを、保管庫に移動します。

接触ＩＣカードで、カード挿入口センサがあり、かつ置き忘れられたカードを安全に保管する保管庫を有するデバイスで有効です。

カード排出処理を行ったあと一定時間経過してもカード挿入口センサがカードを検出するときに、カードを保管する目的でアプリケーションから呼び出されます。

戻り値　　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	カードが正常に保管されました。
OPOS_E_FAILURE	カードの保管に失敗しました。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

CashDeposit メソッド　　第1.15版で追加

形式　LONG CashDeposit (LONG *SequenceNumber*, CURRENCY *Amount*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	チャージのためのシーケンス番号
Amount	チャージ金額
Timeout	デバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　チャージを行います。

*SequenceNumber*によって指定された処理として、 *Amount* 額をチャージします。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

CheckCard メソッド　　第1.15版で追加

形式　LONG CheckCard (LONG *SequenceNumber*, LONG *Timeout*);

パラメータ	説明
SequenceNumber	承認のためのシーケンス番号
Timeout	デバイスから応答がくるまでの最大待ち時間（ミリ秒単位）です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　カードチェックを行います。

*SequenceNumber*によって指定された処理としてカードチェックを行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

電子マネー決済デバイスでは、電子マネー決済デバイスに対して、残高照会を要求します。

残高は、**Balance**プロパティに設定されます。

残高照会の結果、残高以外の情報をデバイスが返すことができる場合には、決済処理の結果と同様に、それぞれのプロパティに情報が設定されるでしょう。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に承認処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又はカードチェックはサポートされていません。(**CapCheckCard**プロパティ参照)
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。

その他　　**ResultCode**の項目を参照してください。

参照　　**CapCheckCard**プロパティ、ErrorEventイベント

CheckServiceRegistrationToMedium メソッド　*第1.14.1版で追加*

形式　**LONG CheckServiceRegistrationToMedium**
(LONG *SequenceNumber***, LONG** *Timeout***);**

パラメータ	説明
SequenceNumber	シーケンス番号
Timeout	デバイスから応答がくるまでの最大待ち時間 (ミリ秒単位) です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　媒体へ、電子バリューサービスが登録可能か確認します。

登録できない場合には、OPOS_E_EXTENDEDエラーが通知され、ResultCodeExtendedプロパティに登録できない理由を示すエラーコードが設定されます。

*SequenceNumber*によって指定された要求として媒体へ電子バリューの登録が可能かどうか確認します。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	媒体へ、サービスの登録が可能です。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又はサポートされていません。
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ResultCodeExtended**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

ClearParameterInformationメソッド　　第1.14版で追加

形式　**LONG ClearParameterInformation ();**

説明　コントロールに保持されている、**SetParameterInformation**メソッドで設定されているタグ付きの値を全て初期化します。

戻り値　次の値のいずれかが戻されます。

値	意味
OPOS_SUCCESS	直前の取引の結果が取得できました。
その他	**ResultCode**の項目を参照してください。

CloseDailyEVService メソッド　　*第1.14.1版で追加*

形式　**LONG CloseDailyEVService (LONG*** *pData*, **BSTR*** *pString*);

パラメータ	説明
pData	数値データのポインタ。値は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。値はサービスオブジェクトによって変化します。

説明　電子バリューリーダライタのサービスの終了処理を行います。

デバイスもしくはサービスがサービスの終了処理を持つ場合にサポートされます。

CurrentServiceプロパティに設定されたサービスの終了処理を実行します。処理の内容は実装に依存します。

非同期処理でこのメソッドが起動されると、サービスオブジェクトはEVRW_TE_NOTIFY_PROGRESSに1から100を加えた進捗のパーセント値で**TransitionEvent**を通知し、処理の状態をアプリケーションに報告します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
その他	**ResultCode**の項目を参照してください。

DeactivateEVService メソッド　　*第1.14.1版で追加*

形式　**LONG DeactivateEVService (LONG*** *pData*, **BSTR*** *pString*);

パラメータ	説明
pData	数値データのポインタ。値は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。値はサービスオブジェクトによって変化します。

説明　電子バリューリーダライタの非活性化処理を行います。

デバイスもしくはサービスが非活性化処理機能を持つ場合にサポートされます。

非活性化処理は、サービスをアンインストールするときやデバイスを撤去するとき、機能を無効化するときに行われる処理です。処理の内容やパラメータの内容は実装に依存します。

非同期処理でこのメソッドが起動されると、サービスオブジェクトはEVRW_TE_NOTIFY_PROGRESSに1から100を加えた進捗のパーセント値で**TransitionEvent**を通知し、処理の状態をアプリケーションに報告します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
その他	**ResultCode**の項目を参照してください。

EndDetection メソッド

形式　**LONG EndDetection ();**

説明　カードの検出を終了します。

呼び出されると、デバイスはカードの検出モードを終了します。

カードが検出されると、OPOS_SUCCESSが返されます。

カードが検出されていないと、OPOS_E_EXTENDEDエラーが戻され、拡張エラーコードにOPOS_EVRW_NOCARDがセットされます。

このメソッドは、**BeginDetection**メソッドとセットで呼び出します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	カードが正常に検出されました。
OPOS_E_ILLEGAL	デバイスがカード検出モードではありません。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

EndRemoval メソッド

形式　**LONG EndRemoval ();**

説明　カードの排出を終了します。

呼び出されると、デバイスはカード排出モードを終了します。

カードがデバイスに感知されないと、OPOS_SUCCESSがアプリケーションに返されます。

カードがデバイスに残っている場合、OPOS_E_EXTENDEDエラーが戻され、拡張エラーコードにOPOS_EVRW_RELEASEがセットされます。

非接触型ＩＣカードデバイスの場合、このメソッドは必ずしも実装される必要はなく、常に正常終了します。

このメソッドは、カード排出処理のため、**BeginRemoval**メソッドとセットで呼び出します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	カードが正常に排出されました。
OPOS_E_ILLEGAL	デバイスはカード排出モードではありません。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

EnumerateCardServices メソッド

形式　**LONG EnumerateCardServices ();**

説明　カード内の利用可能なサービスを列挙し、**CardServiceList**プロパティに設定します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	処理が成功しました。
OPOS_E_EXTENDED	内容については**ResultCodeExtended**を参照してください。
その他	**ResultCode**の項目を参照してください。

LockTerminal メソッド　*第1.14.1版で変更*

形式　**LONG LockTerminal ();**

説明　デバイスもしくはサービスをロックし、**UnlockTerminal**メソッドによりロック状態が解除されるまで、使用できないようにします。

デバイスもしくはサービスがセキュリティロック機能を持つ場合にサポートされます。

ロックするために鍵となる情報が必要な場合には、**AdditionalSecurityInformation**プロパティに設定します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	セキュリティロック機能を持ちません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ResultCodeExtended**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

OpenDailyEVService メソッド　　*第1.14.1版で追加*

形式　**LONG OpenDailyEVService (LONG*** *pData*, **BSTR*** *pString*);

パラメータ	説明
pData	数値データのポインタ。値は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。値はサービスオブジェクトによって変化します。

説明　電子バリューリーダライタのサービスの開始処理を行います。

デバイスもしくはサービスがサービスの開始処理を持つ場合にサポートされます。

CurrentServiceプロパティに設定されたサービスの開始処理を実行します。処理の内容は実装に依存します。

非同期処理でこのメソッドが起動されると、サービスオブジェクトはEVRW_TE_NOTIFY_PROGRESSに1から100を加えた進捗のパーセント値で**TransitionEvent**を通知し、処理の状態をアプリケーションに報告します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
その他	**ResultCode**の項目を参照してください。

QueryLastSuccessfulTransactionResult メソッド　　第1.14版で追加

形式　**LONG QueryLastSuccessfulTransactionResult ( );**

説明　直前に実行され、成功した取引メソッド（**ReadValue**メソッド、**WriteValue**メソッド、**AddValue**メソッド、**SubtractValue**メソッド、**CancelValue**メソッド、**AccessLog**メソッド）の結果を確認するためのメソッドです。

直前に実行され、成功した取引メソッドの種類は、**TransactionType**プロパティに設定されます。

メソッドが成功すると、メソッドの戻り値はOPOS_SUCCESSとなり、プロパティ値は**ResultCode**プロパティ値を含めて直前に実行され、成功した取引メソッドの結果を示します。

直前に実行され、成功した取引メソッドが**ReadValue**メソッドの場合には、プロパティの値は**DataEvent**イベントもしくは**ErrorEvent**イベントが通知された状態を示します。

取引メソッドの要求後、取引が終了し、結果が通知されている最中に電源断や接続断が発生し、取引結果が不明となる場合があります。電子バリューリーダライタデバイスによっては、このような場合の為に直前の取引結果をデバイスに問い合わせる機能を持つものがあります。本メソッドはこの機能を実現するためのものです。

戻り値　次の値のいずれかが戻されます。

値	意味
OPOS_SUCCESS	直前の取引の結果が取得できました。
その他	**ResultCode**の項目を参照してください。

ReadValue メソッド　　第1.16.1版で更新

形式　**LONG ReadValue (LONG** *SequenceNumber***, LONG** *Timeout***);**

パラメータ	説明
SequenceNumber	シーケンス番号
Timeout	デバイスから応答がくるまでの最大待ち時間 (ミリ秒単位) です。OPOS_FOREVER(-１)、０、正の値が指定できます。タイムアウトの設定はデバイスの実装によって異なり、読み込み完了までの最大時間、またはデバイス応答までの最大時間のいずれかです。

説明　カードから電子バリューの読み込みを行います。

*SequenceNumber*によって指定された要求としてカードから電子バリューの読み込みを行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又はサポートされていません。
OPOS_E_TIMEOUT	デバイスへの書き込みが、設定されたタイムアウト期間内に正常に完了しませんでした。 ※補足：readValue メソッドの実行が正常に完了した後、デバイスからの応答を待機している間に、指定されたタイムアウトを超えても応答がない場合は、E_TIMEOUT ではなく ErrorEvent が提供されます。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

RegisterServiceToMedium メソッド　*第1.14.1版で追加*

形式　**LONG RegisterServiceToMedium (LONG** *SequenceNumber***, LONG** *Timeout***);**

パラメータ	説明
SequenceNumber	シーケンス番号
Timeout	デバイスから応答がくるまでの最大待ち時間 (ミリ秒単位) です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　媒体へ、電子バリューサービスの登録を行います。

*SequenceNumber*によって指定された要求として媒体へ電子バリューの登録を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又はサポートされていません。
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ResultCodeExtended**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

RetrieveResultInformationメソッド　*第1.16.1版で更新*

形式　**LONG RetrieveResultInformation (BSTR** *Name*, **BSTR*** *Value*);

パラメータ	説明
Name	値を取得するタグの名前です。
Value	*Name*パラメータにより得られるタグの値が戻ります。*Name*パラメータが認識されない、もしくは、現在のカードの型にサポートされない場　合、値には何も返しません。

説明　最後に実行したメソッドの処理結果について、電子バリューリーダライタデバイスから返される追加情報を取得します。

追加情報はタグ付の値として、コントロールに保持されます。

新たなメソッドの処理が実行された時点で、追加情報は更新されます。

取得可能なタグと値は以下の通り。

タグ名	説明	型	備考
AccessLogLastDateTime	前回取引ログ取得日付時刻	日付時刻型	前回の取引ログ取得を実施した日付時刻。
AccountNumber	会員番号	文字列型	電子バリューサービスの会員番号。同じ情報をプロパティに持つが、タグを使うことを推奨する。
Amount	要求決済金額	通貨型	電子バリューリーダライタに要求する決済金額。同じ情報をプロパティに持つが、タグを使うことを推奨する。
AmountForPoint	ポイント計算対象金額	通貨型	ポイントを計算する対象となる金額。決済と同時に加算ポイント値を電子バリューリーダライタデバイスが計算するような場合で、ポイントの対象とならない商品が取引に含まれるような場合に、ポイント計算の対象となる金額を指定する。
AuthenticationStatus	認証状態	列挙型	認証の状態を示す。
AutoCharge	オートチャージ	論理型	メソッド発行時点でのオートチャージの実施要求、もしくは処理完了時のオートチャージ結果。

Balance	残高	通貨型	電子バリューサービスの残高。同じ情報をプロパティに持つが、タグを使うことを推奨する。
BalanceOfPoint	残ポイント数	通貨型	ポイントがある場合の残ポイント数。同じ情報をプロパティに持つが、タグを使うことを推奨する。
BusinessUnitID	店舗番号	文字列型	店舗番号。
CardCompanyName	カード会社名	文字列型	電子バリューの媒体（カードや携帯電話）を発行している会社名。
CardTransactionLogID	カード取引ログID	文字列型	電子バリューサービスの媒体（カードや携帯電話）に格納される取引明細の識別番号。
CardTransactionNumber	カード取引通番	数値型	電子バリューサービスの媒体（カードや携帯電話など）が採番し管理する取引番号。
ChargeMethod	チャージ種別	列挙型	電子バリューサービスにチャージするときの手段（現金、ポイント交換など）を指定する。
ChargeableAmount	チャージ可能金額	通貨型	チャージ決済が可能な金額。
ChargeableCount	チャージ可能回数	数値型	チャージ決済が可能な回数。

DateTime	日付時刻	日付時刻型	メソッド発行時点・イベント通知時点もしくは処理完了時点での日付時刻。
EffectiveDaysOfKey	鍵の有効期間	数値型	鍵が有効である日数
EVRWApprovalCode	リーダライタ承認番号	文字列型	電子バリューリーダライタで採番され管理される、処理の承認番号。
EVRWDataUpdateDateTime	リーダライタ内部パラメータ更新日付	日付時刻型	電子バリューリーダライタの内部パラメータが更新された日付時刻
EVRWDateTime	リーダライタ日付時刻	日付時刻型	電子バリューリーダライタで管理する日付時刻。
EVRWID	リーダライタ番号	文字列型	電子バリューリーダライタの識別番号。
EVRWTransactionLogID	リーダライタ取引ログID	文字列型	電子バリューリーダライタに格納される取引明細の識別番号。
EVRWTransactionNumber	リーダライタ取引通番	数値型	電子バリューリーダライタが採番し管理する取引番号。
EndAccountID	会員番号終点	文字列型	電子バリューリーダライタに締めや集計を要求するときの、会員番号で指定される終点。

EndDateTime	日付時刻終点	日付時刻型	電子バリューリーダライタに締めや集計を要求するときの、日付時刻で指定される終点。
EndEVRWTransactionNumber	リーダライタ取引通番終点	数値型	電子バリューリーダライタに締めや集計を要求するときの、リーダライタ取引通番で指定される終点。
EndPOSTransactionNumber	POS取引通番終点	数値型	電子バリューリーダライタに締めや集計を要求するときの、POS取引番号で指定される終点。
ExpirationDate	満期日、有効期限	日付時刻型	同じ情報をプロパティに持つが、タグを使うことを推奨する。
ExpiredAccountID	失効会員番号	文字列型	電子バリューサービスの媒体（カードや携帯電話）に、過去に失効された会員番号の情報を保持している場合に設定される。
ForceOnlineCheck	強制オンラインチェック	論理型	決済時に強制的にオンラインでセンタにチェックを行うことを強制するフラグ。
InsufficientAmount	残高不足金額	通貨型	電子バリューリーダライタで残高不足となった場合の、不足金額。

ItemCode	商品コード	文字列型	決済された取引で扱われた商品の商品コード。
KeyExpirationDateTime	鍵有効期限	日付時刻型	鍵の有効期限の日付時刻
KeyUpdateDateTime	鍵更新日付	日付時刻型	鍵が更新された日付時刻
LastTimeBalance	取引前残高	通貨型	決済が行われる前の残高。
LastTimeCardTransactionLogID	取引前カード取引ログID	文字列型	一つ前の取引の、電子バリューサービスの媒体（カードや携帯電話）に格納される取引明細の識別番号。
LastTimeEVRWTransactionLogID	取引前リーダライタ取引ログID	文字列型	一つ前の取引の、電子バリューリーダライタに格納される取引明細の識別番号。
LastUsedDateTime	最終利用日時	日付時刻型	同じ情報をプロパティに持つが、タグを使うことを推奨する。
LogCheck	履歴チェックフラグ	論理型	決済取消を行うときに、取引履歴を確認するかどうかを指定するフラグ。
MediaData	カード情報	文字列型	電子バリューの媒体（カードや携帯電話）の情報で、POSアプリケーションが内容を関知しない情報。内容はサービス事業者やベンダが自由に設定可能。

MediumID	決済媒体番号	文字列型	電子バリューサービスの媒体（カードや携帯電話など）の識別番号。同じ情報をプロパティに持つが、タグを使うことを推奨する。
MediumIssuerInformation	媒体発行者情報	文字列型	媒体の発行者に関する情報
MemberInformation	会員情報	文字列型	会員証サービスにおける、会員情報
MerchantID	加盟店ＩＤ	文字列型	加盟店を識別するＩＤ
ModuleID	決済モジュール番号	文字列型	複数のサービスを提供する電子バリューリーダライタで、電子バリューリーダライタの中に複数存在する個々の決済モジュールや決済アプリの識別番号。
NegativeInformationType	ネガ情報種別	列挙型	ネガ情報の種別（全件、差分など）
NegativeInformationUpdateDateTime	ネガ情報更新日付	日付時刻型	ネガ情報が更新された日付時刻
NumberOfAddition	加算取引件数	数値型	チャージ決済取引の件数。
NumberOfEVRWTransactionLog	リーダライタ取引ログ件数	数値型	電子バリューリーダライタに格納される取引明細の件数。
NumberOfFreeEVRWTransactionLog	リーダライタ取引ログ残容量	数値型	電子バリューリーダライタに格納される取引明細の空き件数。

NumberOfRecord	履歴件数	数値型	履歴の件数。
NumberOfSentEVRWTransactionLog	送信済みリーダライタ取引ログ件数	数値型	電子バリューリーダライタに格納される取引明細のうち、決済センターに送信済みの件数。
NumberOfSubtraction	減算取引件数	数値型	決済取引の件数。
NumberOfTransaction	総取引件数	数値型	取引の件数。
NumberOfUncompletedAddition	加算結果不明取引件数	数値型	チャージ決済取引で、電子バリューリーダライタと電子バリューの媒体（カードや携帯電話）の間の通信異常などで結果が不明となった取引の件数。
NumberOfUncompletedSubtraction	減算結果不明取引件数	数値型	決済取引で、電子バリューリーダライタと電子バリューの媒体（カードや携帯電話）の間の通信異常などで結果が不明となった取引の件数。
NumberOfUncompletedVoid	取消結果不明取引件数	数値型	取引の取消で、電子バリューリーダライタと電子バリューの媒体（カードや携帯電話）の間の通信異常などで結果が不明となった取引の件数。
NumberOfVoid	取消取引件数	数値型	取引の取消の件数。

OtherAmount	併用決済金額	通貨型	当該取引を他の決済方法と併用して決済している場合の、併用決済金額。
POSDateTime	POS日付時刻	日付時刻型	POSで管理する会計上の日付時刻。
POSTransactionNumber	POS取引通番	数値型	POS上の取引通番。
PaymentCondition	支払種別	列挙型	ポストペイ型の電子バリューサービスの場合の、決済額の支払種別を示す。
PaymentDetail	支払種別詳細	文字列型	ポストペイ型の電子バリューサービスの場合の、決済額の支払種別の詳細を示す。
PaymentMethod	決済種別	列挙型	電子バリューリーダライタに要求する決済について、取引金額の全額決済か、他決済手段と併用をしているかを指定する。
PaymentMethodForPoint	ポイント計算対象決済方法	列挙型	ポイント計算対象金額の決済方法を示す。
Point	ポイント値	数値型	POSから電子バリューリーダライタに要求するポイント値。同じ情報をプロパティに持つが、タグを使うことを推奨する。
RegistrableServiceCapacity	登録可能サービス数	数値型	登録可能なサービス数

RequestedAutoChargeAmount	実オートチャージ要求金額	通貨型	オートチャージでチャージを要求する金額。
ResponseCode1	結果コード	数値型	処理の結果コード。内容はサービス事業者やベンダが自由に設定可能。
ResponseCode2	結果詳細コード	数値型	処理の詳細な結果コード。内容はサービス事業者やベンダが自由に設定可能。
ResultOnSettlement	取引状態	列挙型	電子バリューリーダライタと電子バリューの媒体（カードや携帯電話）の間での決済取引の結果状態。
RetryTimeout	カード処理再試行タイムアウト	数値型	電子バリューリーダライタと電子バリューの媒体（カードや携帯電話）の間で処理の再試行が必要となったときの、タッチされるまでの待ち時間（ミリ秒）。
SettledAmount	実決済金額	通貨型	電子バリューリーダライタで実際に決済された金額。同じ情報をプロパティに持つが、タグを使うことを推奨する。
SettledAutoChargeAmount	オートチャージ金額	通貨型	電子バリューリーダライタで実際に決済されたオートチャージ金額。

SettledMemberInformation	更新された会員情報	文字列型	会員証サービスにおける、更新されたあとの会員情報
SettledOtherAmount	実併用決済金額	通貨型	電子バリューサービスとの併用決済が行われた場合の、電子バリューサービス外の実決済金額。
SettledPoint	実決済ポイント値	数値型	電子バリューリーダライタで実際に決済されたポイント値。
SettlementNumber	精算通番	数値型	精算処理の通番。
SettledVoucherID	実決済クーポン／チケットＩＤ	文字列型	クーポン／チケットサービスにおいて、電子バリューリーダライタで実際に決済されたクーポン／チケットのＩＤを設定する。クーポン／チケットは、権利の種類を示す識別子と個数の対で表現する。識別子 001 のチケット６枚は、「001:6」という文字列であらわされる。「:」は、識別子と枚数のセパレータを示す。同じ情報をプロパティに持つが、タグを使うことを推奨する。
SignatureFlag	署名フラグ	論理型	決済のあと、署名が必要かどうかを示すフラグ。

SoundAssistFlag	音声案内フラグ	論理型	音声案内を行うかどうかを示すフラグ。
StartAccountID	会員番号起点	文字列型	電子バリューリーダライタに締めや集計を要求するときの、会員番号で指定される起点。
StartDateTime	日付時刻起点	日付時刻型	電子バリューリーダライタに締めや集計を要求するときの、日付時刻で指定される起点。
StartEVRWTransactionNumber	リーダライタ取引通番起点	数値型	電子バリューリーダライタに締めや集計を要求するときの、リーダライタ取引通番で指定される起点。
StartPOSTransactionNumber	POS取引通番起点	数値型	電子バリューリーダライタに締めや集計を要求するときの、POS取引番号で指定される起点。
SummaryTermType	集計期間種別	列挙型	集計処理の期間を指定する。
TargetService	対象サービス文字列	文字列型	ActivateEVServiceメソッドを実行する際に、対象となるサービスを対象となるサービスを識別する文字列
TaxOthers	税・その他金額	通貨型	電子バリューリーダライタに要求する決済金額のうち、税やその他の金額。

TotalAmountOfAddition	加算取引金額合計	通貨型	チャージ決済取引の金額の合計。
TotalAmountOfSubtraction	減算取引金額合計	通貨型	決済取引の金額の合計。
TotalAmountOfTransaction	総取引金額	通貨型	取引の金額の合計。
TotalAmountOfUncompletedAddition	加算結果不明金額合計	通貨型	チャージ決済取引で、電子バリューリーダライタと電子バリューの媒体（カードや携帯電話）の間の通信異常などで結果が不明となった取引の金額の合計。
TotalAmountOfUncompletedSubtraction	減算結果不明金額合計	通貨型	決済取引で、電子バリューリーダライタと電子バリューの媒体（カードや携帯電話）の間の通信異常などで結果が不明となった取引の金額の合計。
TotalAmountOfUncompletedVoid	取消結果不明取引金額	通貨型	取引の取消で、電子バリューリーダライタと電子バリューの媒体（カードや携帯電話）の間の通信異常などで結果が不明となった取引の金額の合計。
TotalAmountOfVoid	取消取引金額合計	通貨型	取引の取消の金額合計。

TouchTimeout	タッチ待ちタイムアウト	数値型	電子バリューリーダライタに電子バリューの媒体（カードや携帯電話）がタッチされるまでの待ち時間（ミリ秒）。
TransactionType	取引種別	列挙型	電子バリューサービスでの取引の種別。
UILCDControl	LCD制御有無	論理型	電子バリューリーダライタがLCDを持つときに、LCDの制御の有無を指定する。
UILEDControl	LED制御有無	論理型	電子バリューリーダライタがLEDを持つときに、LEDの制御の有無を指定する。
UISOUNDControl	鳴動制御有無	論理型	電子バリューリーダライタが鳴動するときに、鳴動の制御の有無を指定する。
VOIDorRETURN	取消返品種別	列挙型	キャンセルする際、取消とするか、返品とするかを指定する。
VoidTransactionType	取消対象取引区分	列挙型	取引の取消を行う場合の、取引の種類を指定する。

VoucherID	クーポン／チケットＩＤ	文字列型	クーポン／チケットサービスにおいて、クーポン／チケットのＩＤを設定する。クーポン／チケットは、権利の種類を示す識別子と個数の対で表現する。識別子 001 のチケット６枚は、「001:6」という文字列であらわされる。「:」は、識別子と枚数のセパレータを示す。同じ情報をプロパティに持つが、タグを使うことを推奨する。
VoucherIDList	クーポン／チケットＩＤのリスト	文字列型	クーポン／チケットサービスにおいて、列挙されたクーポン／チケットのＩＤを設定する。識別子 001 のチケット６枚、識別子 002 のチケット１枚、識別子 034のチケット２枚を保持している場合、「001:6, 002:1, 034:2」という形式のＣＳＶ文字列で表現される。「,」は、複数の権利を保持している場合のセパレータを示す。同じ情報をプロパティに持つが、タグを使うことを推奨する。
WorkstationID	POS番号	文字列型	POSの識別番号。

WorkstationMakerCode	POSのメーカーコード	文字列型	POSのメーカーコード
WorkstationSerialNumber	POSの製造番号	文字列型	POSの製造番号（シリアル番号）

タグの値は全て文字列型で、以下の書式で設定されます。

型	書式
文字列型	文字列
数値型	３２ビット整数を文字列で表したもの
通貨型	６４ビット整数を文字列で表したもの 小数点以下を４桁固定で定義する たとえば"1234567"は、"123.4567"を意味する
日付時刻型	yyyy'-'mm'-'dd'T'hh':'mm':'ss'.'sss zzzzzzの形式の日付時刻 '-'は日付の各要素の区切り文字。 yyyyは年を、mmは月を、ddは日付を示す。 'T'は日付と時刻の区切り文字で、':'は時刻の各要素の区切り文字。 hhは時を、mmは分を、ssは秒を示す。 '.'は秒とミリ秒の区切り文字。 sssはミリ秒を示す。 zzzzzzはタイムゾーンを表し、GMTか、GMTからの差分を(('+'|'-')hh':'mm)の形式で示す。'+'と'-'はGMTとの差分の正負を示し、hhは00から14の間の値でGMTとの差分時刻を示し、mmは00から59の間の値でGMTとの差分の分を示す。 絶対時間を表すことで、夏時間などの影響を受けないようにする。 2008-04-12T23:20:50.275は2008年4月12日23時20分50.275秒を示す。 2008-04-12T22:20:50.275+01:00は同日付時刻のジェノバの時刻を表し、2008-04-12T17:20:50.275-05:00は同日付時刻のニューヨークの時刻を表す。
論理型	文字列'true'と'false'の列挙型。
列挙型	それぞれのタグ毎に定義される識別子を示す文字列。たとえば、EVRW_TAG_AS_AUTHENTICATEDに定義されている値「1」を文字列に変換したものが設定される。

列挙型のタグは、以下の表の値を取ります。

タグ		定義		備考
AuthenticationStatus	認証状態	EVRW_TAG_AS_AUTHENTIFIED	認証済み	認証の状態
		EVRW_TAG_AS_UNAUTHENTIFIED	認証前	
CancelTransactionType	取引中断種別	EVRW_TAG_CTT_CANCEL	中断	
		EVRW_TAG_CTT_CHARGE	チャージの中断	
		EVRW_TAG_CTT_RETURN	返品	
		EVRW_TAG_CTT_SALES	売上の中断	
ChargeMethod	チャージ種別	EVRW_TAG_CM_CASH	現金チャージ	現金によるチャージか、ポイント交換によるチャージかを指定する
		EVRW_TAG_CM_CREDIT	クレジットチャージ	
		EVRW_TAG_CM_POINT	ポイントチャージ	
NegativeInformationType	ネガ情報種別	EVRW_TAG_NIT_ALL	全件ネガ情報	扱うネガ情報の種類
		EVRW_TAG_NIT_UPDATED	差分ネガ情報	
PaymentCondition	支払種別	EVRW_TAG_PC_LUMP	一括	支払種別
		EVRW_TAG_PC_BONUS_1	ボーナス１	
		EVRW_TAG_PC_BONUS_2	ボーナス２	

		EVRW_TAG_PC_BONUS_3	ボーナス３	
		EVRW_TAG_PC_BONUS_4	ボーナス４	
		EVRW_TAG_PC_BONUS_5	ボーナス５	
		EVRW_TAG_PC_INSTALLMENT_1	分割１	
		EVRW_TAG_PC_INSTALLMENT_2	分割２	
		EVRW_TAG_PC_INSTALLMENT_3	分割３	
		EVRW_TAG_PC_BONUS_COMBINATION_1	ボーナス併用１	
		EVRW_TAG_PC_BONUS_COMBINATION_2	ボーナス併用２	
		EVRW_TAG_PC_BONUS_COMBINATION_3	ボーナス併用３	
		EVRW_TAG_PC_BONUS_COMBINATION_4	ボーナス併用４	
		EVRW_TAG_PC_REVOLVING	リボルビング	

PaymentMethod	決済種別	EVRW_TAG_PM_FULL_SETTLEMENT	全額決済	全額決済か、他決済手段と併用しているかを指定する
		EVRW_TAG_PM_COMBINED	併用決済	
PaymentMethodForPoint	ポイント計算対象決済方法	EVRW_TAG_PMFP_EM	電子マネー	ポイント計算対象金額の決済手段
		EVRW_TAG_PMFP_CASH	現金	
		EVRW_TAG_PMFP_CREDIT	クレジット	
		EVRW_TAG_PMFP_OTHER	そのほか	
ResultOnSettlement	取引状態	EVRW_TAG_ROS_OK	正常終了	決済取引の結果
		EVRW_TAG_ROS_UNKNOWN	不明	
		EVRW_TAG_ROS_NG	異常終了	
SummaryTermType	集計期間種別	EVRW_TAG_STT_1	前回日計～現在	集計を行う期間を指定する種別
		EVRW_TAG_STT_2	前々回日計から前回日計	
		EVRW_TAG_STT_3	前々々回日計から前々回日計	
TransactionType	取引種別	EVRW_TAG_TT_READ	読み込み（照会）	決済、チャージ、など、取引の種別
		EVRW_TAG_TT_WRITE	書き込み	
		EVRW_TAG_TT_ADD	加算（チャージ）	

		EVRW_TAG_TT_SYBTRACT	減算（売上）	
		EVRW_TAG_TT_CANCEL_RETURN	取消／返品	
		EVRW_TAG_TT_GET_LOG	取引ログの取得	
		EVRW_TAG_TT_CANCEL_SALES	売上取消	
		EVRW_TAG_TT_CANCEL_CHARGE	チャージ取消	
		EVRW_TAG_TT_RETURN	返品	
		EVRW_TAG_TT_COMPLETION	承認後売上	
		EVRW_TAG_TT_PRE_SALES	事前承認	
VOIDorRETURN	取消返品種別	EVRW_TAG_VR_VOID	取消	
		EVRW_TAG_VR_RETURN	返品	
VoidTransactionType	取消対象取引区分	EVRW_TAG_VTT_SALES	売上取消	取消の対象（売上かチャージか）を指定する
		EVRW_TAG_VTT_CHARGE	チャージ取消	
		EVRW_TAG_VTT_CANCEL	取消	

		EVRW_TAG_VTT_RETURN	返品	

戻り値　次の値のいずれかが戻されます。

値	意味
OPOS_SUCCESS	直前の取引の結果が取得できました。
その他	**ResultCode**の項目を参照してください。

SetParameterInformationメソッド　　*第1.16.1版で更新*

形式　**LONG SetParameterInformation (BSTR** *Name*, **BSTR** *Value*);

パラメータ	説明
Name	値を設定するタグの名前です。
Value	*Name*パラメータで指定されたタグに設定する値を指定します。*Name*パラメータが認識されない、もしくは、現在のカードの型にサポートされない場　合、値の内容は変更されません。

説明　メソッド発行前に、電子バリューリーダライタデバイスが要求する追加情報をセットします。

追加情報はタグ付の値として、コントロールに保持されます。設定可能なタグと値については、**RetrieveResultInformation**メソッドの説明を参照してください。

アプリケーションは、**ClearParameterInformation**メソッドを呼び出すことにより、明示的にコントロールに保持されている、電子バリューリーダライタデバイスが要求する追加情報を全て初期化することができます。

戻り値　次の値のいずれかが戻されます。

値	意味
OPOS_SUCCESS	直前の取引の結果が取得できました。
OPOS_E_ILLEGAL	存在しないタグの名前が指定されました。
その他	**ResultCode**の項目を参照してください。

SubtractValue メソッド

形式　**LONG SubtractValue (LONG** *SequenceNumber***, LONG** *Timeout***);**

パラメータ	説明
SequenceNumber	シーケンス番号
Timeout	デバイスから応答がくるまでの最大待ち時間 (ミリ秒単位) です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　カードへ電子バリューの減算を行います。

*SequenceNumber*によって指定された要求としてカードへ電子バリューの減算を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又はサポートされていません。
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

TransactionAccess メソッド　　第1.16.1版で更新

形式　**LONG TransactionAccess(LONG** *Control***);**

パラメータ	説明
Control	一括処理。値は以下を参照。

値	意味
EVRW_TA_TRANSACTION	一括処理の開始。
EVRW_TA_NORMAL	バッファの操作を実施して一括処理を終了する。

説明　一括処理モードに入る時または抜ける時にこのメソッドを呼び出します。

*Control*がEVRW_TA_TRANSACTIONならば、一括処理モードに入ります。その後の**WriteValue**メソッド、**AddValue**メソッド、**SubtractValue**メソッド、**CancelValue**メソッド、およびプロパティの設定は、*Control*パラメータにPTR_TP_NORMALがセットされて**TransactionAccess**が呼び出されるまで、バッファリングされます。リーダライタでバッファリングされるか、サービスオブジェクトでバッファリングされるかは、デバイスの実装に依存します。

*Control*がEVRW_TA_NORMALならば、一括処理モードを抜けます。**WriteValue**メソッド、**AddValue**メソッド、**SubtractValue**メソッド、**CancelValue**メソッド、およびプロパティの設定により要求がバッファリングされていれば、それらの要求はカードに対して実行されます。一括処理全体は、一つの処理要求として処理されます。このメソッドは、**AsyncMode**がFALSEならば、同期的に実行され、**AsyncMode**がTRUEならば、非同期的に実行されます。

ClearOutputメソッドを呼び出すことによって、一括処理モードは取り消されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。

UnlockTerminal メソッド　*第1.14.1版で変更*

形式　**LONG UnlockTerminal ();**

説明　ロックを解除し、デバイスもしくはサービスを使用できるようにします。

デバイスもしくはサービスがセキュリティロック機能を持つ場合にサポートされます。

ロックを解除するために鍵となる情報が必要な場合には、**AdditionalSecurityInformation**プロパティに設定します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	セキュリティロック機能を持ちません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ResultCodeExtended**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

UnregisterServiceToMedium メソッド　　*第1.14.1版で追加*

形式　**LONG UnregisterServiceToMedium (LONG** *SequenceNumber***, LONG** *Timeout***);**

パラメータ	説明
SequenceNumber	シーケンス番号
Timeout	デバイスから応答がくるまでの最大待ち時間 (ミリ秒単位) です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　媒体から、電子バリューサービスの削除を行います。

*SequenceNumber*によって指定された要求として媒体から電子バリューの削除を行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又はサポートされていません。
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ResultCodeExtended**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

UpdateData メソッド　*第1.14.1版で追加*

形式　**LONG UpdateData (LONG** *DataType*, **LONG*** *pData*, **BSTR*** *pString*);

パラメータ	説明
DataType	更新を行うデータのタイプを指定します。値は以下の通りです。

値	意味
EVRW_UD_KEY	鍵情報
EVRW_UD_NEGATIVE_LIST	ネガティブリスト情報
EVRW_UD_OTHERS	そのほかの情報。

pData	数値データのポインタ。値は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。値はサービスオブジェクトによって変化します。

説明　電子バリューリーダライタのデータの更新を行います。

更新対象となるデータについてはデバイスやサービスオブジェクトにより異なり、処理の内容やパラメータの内容は実装に依存します。

進捗の通知（第1.14.1版で追加）
非同期処理でこのメソッドが起動されると、サービスオブジェクトはEVRW_TE_NOTIFY_PROGRESSに1から100を加えた進捗のパーセント値で**TransitionEvent**を通知し、処理の状態をアプリケーションに報告します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	デバイスは鍵情報の更新機能を持ちません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ResultCodeExtended**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

UpdateKey メソッド　*第1.14.1版で変更*

形式　**LONG UpdateKey (LONG*** *pData*, **BSTR*** *pString*);

パラメータ	説明
pData	数値データのポインタ。値は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。値はサービスオブジェクトによって変化します。

説明　電子バリューリーダライタの鍵情報の更新を行います。

デバイスが鍵情報の更新の機能を持つ場合にサポートされます。

処理の内容やパラメータの内容は実装に依存します。

進捗の通知（第1.14.1版で追加）
非同期処理でこのメソッドが起動されると、サービスオブジェクトはEVRW_TE_NOTIFY_PROGRESSに1から100を加えた進捗のパーセント値で**TransitionEvent**を通知し、処理の状態をアプリケーションに報告します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	デバイスは鍵情報の更新機能を持ちません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ResultCodeExtended**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

WriteValue メソッド

形式　**LONG WriteValue (LONG** *SequenceNumber***, LONG** *Timeout***);**

パラメータ	説明
SequenceNumber	シーケンス番号
Timeout	デバイスから応答がくるまでの最大待ち時間 (ミリ秒単位) です。OPOS_FOREVER(-１)、０、正の値が指定できます。

説明　カードへ電子バリューの書き込みを行います。

*SequenceNumber*によって指定された要求としてカードへ電子バリューの書き込みを行います。

*Timeout*がOPOS_FOREVER(-１)の場合は、タイムアウトは発生せず、デバイスから応答があるまで無限に待ちます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにも格納されます。

値	意味
OPOS_SUCCESS	正常に処理が行われました。
OPOS_E_ILLEGAL	無効な*Timeout*パラメータが指定されました。又はサポートされていません。
OPOS_E_TIMEOUT	指定された*Timeout*ミリ秒の間に、デバイスからの応答がありません。
OPOS_E_EXTENDED	**ResultCodeExtended**に詳細なコードが格納されています。 詳細は**ErrorEvent**の項を参照してください。
OPOS_E_BUSY	デバイスが現在コマンドを受け付けられません。
その他	**ResultCode**の項目を参照してください。

イベント

DataEvent イベント

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

説明　　デバイスからデータが入力されたことをアプリケーションに通知します。このイベントが通知されるまえに、データは適切なプロパティに設定されます。

ErrorEvent イベント

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

パラメータ	説明
ResultCode	エラーイベントを起こした要因のコード。 値は**ResultCode**の項目を参照。
ResultCodeExtended	エラーイベントを起こした要因の拡張コード。値は下記の値を参照。
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へのポインタです。 下記の値を参照。

もし*ResultCode*がOPOS_E_EXTENDEDであれば、*ResultCodeExtended*は次に示す値のいずれかに設定されます。

値	意味
OPOS_EVRW_CENTERERROR	承認機関（センター）からエラーが返されました。
OPOS_EVRW_COMMANDERROR	デバイスに送信されたコマンドが間違っています。コントロールが正常に動作している限りは、このエラーが返されることはありません。
OPOS_EVRW_RESET	処理中にデバイスのリセットキー（中断キー）等で処理が中断されました。
OPOS_EVRW_COMMUNICATIONERROR	デバイスと承認機関（センター）間で通信障害が発生しました。
OPOS_EVRW_LOGOVERFLOW	取引ログが大きすぎて格納できませんでした。取引ログ取得は中断され、**TransactionLog**プロパティの値は不定です。
OPOS_EVRW_DAILYLOGOVERFLOW	処理を試みましたが、デバイスの取引ログが一杯なため、決済できませんでした。

OPOS_EVRW_DEFICIENT
残高が不足しているため、減算できませんでした。

OPOS_EVRW_OVERDEPOSIT
チャージ可能量を超えるため、追加できませんでした。

*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間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。

アプリケーションはエラーイベントハンドラ内で*pErrorRespons*に次のいずれかの値を設定できます。

値	意味
OPOS_ER_RETRY	入力または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以外の場合、使用できないことがあります。*ErrorLocus*がOPOS_EL_OUTPUTの場合のデフォルトです。
OPOS_ER_CLEAR	非同期出力、バッファリングされている出力データ (全ての非同期出力を含む)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 のときに限り設定されます。エラーを認識し、デバイスに処理の継続を指示します。デバイスはエラー状態のままですが、**DataEventEnabled** プロパティの設定にしたがって、別の**DataEvent** が引き続き通知されます。すべての入力データが通知されて、**DataEventEnabled** が再びTRUEに設定されたとき、*ErrorLocus*がOPOS_EL_INPUT に設定された新たな**ErrorEvent** 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　非同期でメソッド実行時にエラーを察知し、コントロールの状態がエラー状態に移った時に通知します。

入力エラーは、**ReadValue**メソッドを実行し、データを読む間に発生します。**DataEventEnabled** がTRUEにならないかぎりこのイベントは通知されないので、アプリケーションは通常のシーケンスでイベントを処理できます。エラー状態は、イベントが通知される前に、適切なプロパティに設定されます。

出力エラーは、非同期で**WriteValue**メソッド、**AddValue**メソッド、**SubtractValue**メソッド、**CancelValue**メソッド、**AccessLog**メソッド、**TransactionAccess**メソッドを実行中に発生します。エラー状態は、イベントが通知される前に、適切なプロパティに設定されます。

参照　「ステータス、結果コード、状態モデル」

OutputCompleteEvent イベント

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

説明　前に開始された非同期出力リクエストが正常に終了したときに通知します。

StatusUpdateEvent イベント

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

*Status*パラメータは、デバイスの状態が設定されます。

値	意味
EVRW_SUE_LS_OK	取引ログは充分な空き容量があります。
EVRW_SUE_LS_NEARFULL	取引ログはフルに近い状態です。
EVRW_SUE_LS_FULL	取引ログは一杯です。
EVRW_SUE_DS_NOCARD	カード検出センサがカードを未検出です。
EVRW_SUE_DS_DETECTED	カード検出センサがカードを検出しました。
EVRW_SUE_DS_ENTERED	挿入口センサがカードを検出しました。
EVRW_SUE_DS_CAPTURED	保管庫センサがガードを検出しました。
*電源通知の**StatusUpdateEvent**の値*	「第2章 共通プロパティ、メソッド、イベント」の**StatusUpdateEvent**の記述を参照。

説明　デバイスの状態が変化した時に通知されます。

CapLogStatusプロパティがTRUEの場合、取引ログの状態が通知されます。

参照　**CapLogStatus**プロパティ、**LogStatus**プロパティ

TransitionEvent イベント　　第1.15版で追加

形式　**void TransitionEvent (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。イベント番号とサービスオブジェクトで変化する特定値です。
pString	文字列のポインタ。イベント番号とサービスオブジェクトで変化する特定値です。このデータのフォーマットは**BinaryConversion**プロパティの値に依存します。詳細は**BinaryConversion**プロパティを参照してください。

説明　非同期の入出力処理中に、サービスオブジェクトからアプリケーションに通知されるイベントです。アプリケーションから要求された非同期の入出力処理の実行中に、アプリケーションに通知や確認が必要な場合に用いられます。

電子バリューリーダライタのサービスオブジェクトでは、以下のイベント番号について、イベントの内容を規定しています。

イベント名	説明	備考
EVRW_TE_NOTIFY_TOUCH_RETRY	更新リトライ通知	再タッチ要求の通知（一定時間が経過するまで、再タッチをキャンセルすることができない）
EVRW_TE_NOTIFY_TOUCH_RETRY_CANCELABLE	更新リトライ通知（キャンセル可能）	再タッチ要求の通知（いつでも、再タッチをキャンセルすることができる）
EVRW_TE_CONFIRM_TOUCH_RETRY	更新リトライ確認（継続か中止か）	再タッチ要求の確認（イベント終了時、再タッチ待ちを継続するか(1)、キャンセルするか(0)をpDataに指定する）
EVRW_TE_CONFIRM_CANCEL	処理中止確認	処理中止の確認（イベント終了時、処理中止するか(1)、継続するか(0)をpDataに指定する）
EVRW_TE_NOTIFY_INVALID_OPERATION	異常操作通知	異常操作の発生の通知（pDataに事象コードが設定される）
EVRW_TE_CONFIRM_INVALID_OPERATION	異常操作確認	異常操作の発生の通知（pDataに事象コードが指定される。処理を継続するか(1)、処理を異常終了させるか(0)をpDataに指定する）
EVRW_TE_CONFIRM_REMAINDER_SUBTRACTION	残高不足確認	残高不足の発生と、残高の減額の確認（通知時Balanceプロパティに残高が設定される。イベント終了時、残額を全て減額するか(1)、中断するか(0)をpDataに指定する）

EVRW_TE_CONFIRM_CENTER_CHECK	センターチェック確認	センターチェックの実施確認（イベント終了時、センターチェックする場合か(1)、しないか(0)をpDataに指定する）
EVRW_TE_CONFIRM_TOUCH_TIMEOUT	かざし待ちタイムアウト確認	かざし待ちタイムアウト確認（イベント終了時、かざし待ちを継続するか(1)、しないか(0)をpDataに指定する）
EVRW_TE_CONFIRM_AUTO_CHARGE	オートチャージ確認	オートチャージ確認（イベント終了時、実施するか(1)、しないか(0)をpDataに指定する）
EVRW_TE_NOTIFY_CAPTURE_CARD	カード検出通知	カードを検出したことの通知
EVRW_TE_NOTIFY_PIN	PIN入力通知	PIN入力中になったことの通知
EVRW_TE_NOTIFY_CENTER_CHECK	センターチェック通知	センターチェックを行っていることの通知
EVRW_TE_NOTIFY_COMPLETE	処理完了通知	処理完了したことの通知（**OutputCompleteEvent**イベントを通知する前に事前に通知する必要がある場合に使用する）
EVRW_TE_NOTIFY_TOUCH	タッチ待ち通知	タッチ待ちになったことの通知
EVRW_TE_NOTIFY_BUSY	処理中通知	処理中になったことの通知（一定時間の処理時間が発生する処理の前に通知）

EVRW_TE_CONFIRM_CENTER_CHECK_COMPLETE	センターチェック終了	センターチェックの完了の確認（センターチェック完了後の処理を継続するか(1)、処理中断するか(0)をpDataに指定する）
EVRW_TE_CONFIRM_SELECT	決済選択確認	決済するために選択肢がある場合の確認（pStringにCSV形式で選択肢が設定される。イベント終了時、pDataに選択した要素の番号（先頭が１）を指定する）
EVRW_TE_NOTIFY_LOCK	ロック解除通知	カード（携帯）がロックされているので、利用者がロックを解除することを通知する
EVRW_TE_NOTIFY_CENTER_CHECK_COMPLETE	センターチェック終了通知	センターチェックが終了したことの通知
EVRW_TE_CONFIRM_PIN_ENTRY_BY_OUTER_PINPAD	外部デバイスでのPIN入力要求	外部PIN入力デバイスによるPINの入力要求の確認（イベント終了時に処理を中断するか(0)、処理を継続するか(1)をpDataに指定し、処理を継続する場合はpStringに、PINパッドデバイスから取得したPINデータを指定する。PINパッドから有効なPINが得られなかった場合にはpDataにエラー終了すること(2)を指定する）
EVRW_TE_NOTIFY_PROGRESS	処理の進捗通知	EVRW_TE_NOTIFY_PROGRESSに1から100を加えた進捗のパーセント値を通知

EVRW_TE_CONFIRM_DEVICEDATA	デバイスが要求するデータの受け渡し確認	メソッドの処理中に、電子バリューリーダライタデバイスがデータの受け渡しを要求するときに通知される確認イベント。データはAdditionalSecurityInformationプロパティを介して受け渡しされ、データの内容はサービスオブジェクトによって変化する。
EVRW_TE_CONFIRM_SEARCH_TABLE	テーブルサーチ要求	AdditionalSecurityInformationプロパティを介して、暗号化された情報ブロックの受け渡しが発生する。情報ブロックの内容や暗号化の方法は実装に依存する。
EVRW_TE_CONFIRM_PAYMENT_CONDITION	支払方法選択要求	イベント通知時に、pStringには選択可能な支払方法を示す文字列がCSV形式で指定されている。支払方法を示す文字列は、PaymentConditionタグに指定可能な列挙子の値となる。 pData引数に選択された支払方法を示すPaymentConditionタグの列挙子を指定し、pStringには支払種別の詳細をCSV形式で指定してイベントを終了する。支払種別の詳細を示すCSV形式は、PaymentConditionプロパティの仕様に準ずる。

EVRW_TE_CONFIRM_AUTHORIZE	オーソリ通信要求	AdditionalSecurityInformationプロパティを介して、暗号化された情報ブロックの受け渡しが発生する。情報ブロックの内容や暗号化の方法などは実装に依存する。
EVRW_TE_NOTIFY_CHECK_CARD	カードチェック通知	
EVRW_TE_NOTIFY_SELECT_PAYMENT_CONDITION	支払方法選択通知	

イベント番号EVRW_TE_NOTIFY_INVALID_OPERATIONおよびEVRW_TE_CONFIRM_INVALID_OPERATIONの際にpDataに指定される事象コードは以下のように規定しています。

事象コード	説明	備考
1	再かざしカードの不一致	
2	カードの認証エラー発生	
3	再タッチ要求中に、再度処理未了が発生	
4	PIN入力の失敗	
5	詳細確認を行って処理をすることを要求	
6	カードの不一致	
7	複数枚のカードを検知	
8	残高が0のカードを検知	

第　３３章
RFIDスキャナ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.12	Boolean	R/W	Open
BinaryConversion	1.12	Long	R/W	Open
CapCompareFirmwareVersion	1.12	Boolean	R	Open
CapPowerReporting	1.12	Long	R	Open
CapStatisticsReporting	1.12	Boolean	R	Open
CapUpdateFirmware	1.12	Boolean	R	Open
CapUpdateStatistics	1.12	Boolean	R	Open
CheckHealthText	1.12	String	R	Open
Claimed	1.12	Boolean	R	Open
DataCount	1.12	Long	R	Open
DataEventEnabled	1.12	Boolean	R/W	Open
DeviceEnabled	1.12	Boolean	R/W	Open & Claim
FreezeEvents	1.12	Boolean	R/W	Open
OpenResult	1.12	Long	R	無し
OutputID	1.12	Long	R	Open
PowerNotify	1.12	Long	R/W	Open
PowerState	1.12	Long	R	Open
ResultCode	1.12	Long	R	無し
ResultCodeExtended	1.12	Long	R	Open
State	1.12	Long	R	無し
ControlObjectDescription	1.12	String	R	無し
ControlObjectVersion	1.12	Long	R	無し
ServiceObjectDescription	1.12	String	R	Open
ServiceObjectVersion	1.12	Long	R	Open
DeviceDescription	1.12	String	R	Open
DeviceName	1.12	String	R	Open
専用				
CapContinuousRead	1.12	Boolean	R	Open
CapDisableTag	1.12	Boolean	R	Open

CapLockTag	1.12	Boolean	R	Open
CapMultipleProtocols	1.12	Long	R	Open
CapReadTimer	1.12	Boolean	R	Open
CapWriteTag	1.12	Long	R	Open
ContinuousReadMode	1.12	Boolean	R	Open
CurrentTagID	1.12	String	R	Open
CurrentTagProtocol	1.12	Long	R	Open
CurrentTagUserData	1.12	String	R	Open
ProtocolMask	1.12	Long	R/W	Open & Claim
ReadTimerInterval	1.12	Long	R/W	Open & Claim
TagCount	1.12	Long	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.12	無し
Close	1.12	Open
ClaimDevice	1.12	Open
ReleaseDevice	1.12	Open & Claim
CheckHealth	1.12	Open, Claim, & Enable
ClearInput	1.12	Open & Claim
ClearInputProperties	1.12	Open & Claim
ClearOutput	1.12	Open & Claim
CompareFirmwareVersion	1.12	Open, Claim, & Enable
DirectIO	1.12	Open
ResetStatistics	1.12	Open, Claim, & Enable
RetrieveStatistics	1.12	Open, Claim, & Enable
UpdateFirmware	1.12	Open, Claim, & Enable
UpdateStatistics	1.12	Open, Claim, & Enable
専用		
DisableTag	1.12	Open, Claim, & Enable
FirstTag	1.12	Open
LockTag	1.12	Open, Claim, & Enable
NextTag	1.12	Open
PreviousTag	1.12	Open
ReadTags	1.12	Open, Claim, & Enable
StartReadTags	1.12	Open, Claim, & Enable
StopReadTags	1.12	Open, Claim, & Enable
WriteTagData	1.12	Open, Claim, & Enable

WriteTagID	1.12	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.12	Open, Claim, & Enable
DirectIOEvent	1.12	Open, Claim
ErrorEvent	1.12	Open, Claim, & Enable
OutputCompleteEvent	1.12	*適用外*
StatusUpdateEvent	1.12	Open, Claim, & Enable

概説

RFIDスキャナコントロールのOLEプログラマティクIDは“OPOS.RFIDScanner”です。

このデバイスは、第1.12版で追加されました。

機能

RFIDスキャナコントロールは、次の機能を持っています。

RFIDタグからタグIDとユーザーデータを読み取る。

ユーザーデータの部分読み取り。

RFIDスキャナコントロールは、また次の機能をサポートできます。

タグの連続読み取り。

RFIDタグへのタグIDの書き込み。

タグのロック。

指定されたRFIDタグへのユーザーデータの書き込み。

RFIDタグの無効化。

ユーザーデータの部分書き込み。

モデル

RFIDスキャナコントロールは、イベント駆動入力デバイスと、出力デバイスの両方です。入力と出力は常に非同期に行われます。

タグはタグIDとユーザーデータと呼ばれる連続したフィールドから成ると考えられます。

既存の標準規格では現在、タグIDとユーザーデータのフィールドを定義しておらず、それらは、RFIDスキャナコントロールとそのプロトコルプロパティ設定に強く依存した、デバイスに固有の方法で決定されます。

入力モデル

RFIDスキャナは、いくつかの相違はありますが、イベント駆動入力の一般的なモデルに従います。

特徴としては、１回の入力操作で複数のタグのグループを読み取った場合に、ただ１つの**DataEvent**が通知されます。

最も簡単な形式では、*Cmd*パラメータにてデータセレクタ(ID、FullData、PartialData、またはそれらの組み合わせ)や他のパラメータが対応するデフォルト値を持つものとしてReadTagsメソッドを呼び出し、パラメータで指定された基準を満たすタグの集合を返します。

アプリケーションは*FilterID*と*Filtermask*の２つのビットパターンパラメータを指定することでフィルターをかけたタグの読み取りができます。フィルター付きの読み取り操作では、*Filtermask*単独か*FilterID*と*Filtermask*両方のビット単位の論理積にマッチするタグIDを返します。読み取りレンジ内の全てのタグを要求するために、アプリケーションは*Filtermask*にオールゼロを指定できます。全てのタグデータを収集できた時点で、１つのDataEventをキューイングします。タグフィルタリングはハードウェアかRFIDスキャナコントロールのどちらかでサポートしなければなりません。

ユーザーデータの部分読み取りもサポートされる必要があり、ハードウェアで出来ない時にはRFIDスキャナコントロールで行います。ユーザーデータの特定のセグメントにアクセスするために、アプリケーションは*Cmd*パラメータのRFID_RT_PARTIALUSERDATAビットをONにし、対象セグメントの開始位置と長さを指定する構成ができます。CurrentTagUserDataプロパティは指定されたセグメントの現在の内容をあらわし、NextTagのようなナビゲーションメソッドで切り替えます。

AutoDisableプロパティがTRUEならば、DataEventのキューイング時にコントロールは自動的にディセーブルとなります。

DataEventEnabledプロパティがTRUEならば、キューイングしたDataEventはアプリケーションに通知されます。コントロールはこのイベントを発生させる直前に、TagCountプロパティに読み取ったタグの総数をセットし、最初のタグのデータをCurrentTagID,CurrentTagProtocol,CurrentTagUserDataプロパティにコピーし、さらなるデータイベントを発生させないために、DataEventEnabledプロパティをFALSEに設定します。

DataEventを受け取った後、アプリケーションはTagCountプロパティを読むことで読んだタグの総数を決定できます。アプリケーションはFirstTag,NextTag,PreviousTagメソッドを呼び出すことで一連のタグをナビゲートし、CurrentTagID,CurrentTagProtocol,CurrentTagUserDataプロパティによってタグ情報を受け取ることができます。FirstTag,NextTag,PreviousTagメソッドは同期的に行われ、呼び出された時に物理的な入出力は発生しません。

アプリケーションが現在のすべての入力データの処理を終え、次のデータ処理の準備ができたとき、DataEventEnabledプロパティをTRUEに設定することにより、再びデータイベントが通知されるようになります。

入力データの読込み中または処理中に、コントロールにおいてエラーが発生した場合、ErrorEventが発行され、DataEventEnabledプロパティがTRUEならば、アプリケーションに通知されます。

更に別のReadTagsメソッドを呼び出すためには、DataEventかErrorEventを受け取っていなければなりません。コントロールにキューイングされたすべての入力データはClearInputメソッドを呼ぶことにより削除できます。

DataEventもしくはErrorEvent発生により入力された全てのデータプロパティは、ClearInputPropertiesメソッドを呼ぶ事によりデフォルト値に戻す事が出来ます。

CapContinuousReadがTRUEならば、アプリケーションは連続的な読み取りのためにStartReadTagsとStopReadTagsメソッドを呼び出す事が出来ます。StartReadTagsはReadTagsと同じ方法でレンジ内のタグを調べますが、StopReadTagsで中断させられるまで読み取ったタグをDataEventで継続的にキューイングします。

出力モデル

RFIDスキャナは、いくつかの相違と拡張はありますが、一般的な出力モデルに従います。

アプリケーションはCapWriteTagを読むことで書き込みが可能かを決定できます。

書き込みがサポートされているなら、アプリケーションは、WriteTagIDやWriteTagDataメソッドを呼び出すことによって、タグIDとユーザーデータに書き込むことができます。

タグのユーザーデータに関するすべての書き込み操作は部分書き込みと考えられます。（例えばWriteTagDataメソッドは*Start*と*Userdata*パラメータで指定された、タグのユーザーデータフィールドの部分だけを上書きしている）したがって、タグのユーザーデータフィールド全体の内容を上書きするためには、アプリケーションは、*Userdata*パラメータにタグのユーザーデータ部を完全に上書きするのに十分なデータを含めなければなりません。アプリケーションは、既存のデータを完全に上書きするのにヌル(0x0)のバイトで*Userdata*パラメータを埋める必要があり、埋めるのに必要な合計量を決定するために最初にタグのユーザーデータを読む必要があります。

CapLockTagがTRUEならば、アプリケーションはLockTagメソッドを呼び出すことによりタグをロックできます。タグがロックされている時はIDとユーザーデータの両方がリードオンリーとなります。パスワードが必要な場合は、パラメータリストの中でそれを指定することができます。

CapDisableTagがTRUEならば、アプリケーションは永久に無効にしたいタグのタグIDを指定してDisableTagメソッドを呼び出すことによりタグを無効（殺す）にできます。

RFIDスキャナコントロールはその要求を、RFIDハードウェアが受信処理可能になり次第発行するためにバッファリングし、OutputIDプロパティをこの要求のための識別子として設定し、できるかぎり早く制御を戻します。コントロールがその要求を正常に完了した時、OutputCompleteEventをキューイングされます。このイベントのパラメータは、完了した要求のOutputIDを含みます。

非同期の要求が実行されている間に、タイムアウトのようなエラーが発生すると、ErrorEventがキューイングされます。

デバイスの共有

RFIDスキャナは下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、多くのRFIDスキャナに固有のプロパティにアクセスする前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

アプリケーションは、デバイス操作のメソッドを呼び出す前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の必要条件については、「一覧」の表を参照してください。

プロパティ

CapContinuousReadプロパティ

形式 BOOL CapContinuousRead;

説明 TRUEならば、デバイスは連続読み取りをサポートします。 アプリケーションは**StartReadTags**や他の連続読み取りメソッドを呼び出す前にこのプロパティを確認するべきです。

このプロパティは、**Open**メソッドにより初期化されます。

参照 **ContinuousReadMode**プロパティ、**StartReadTags**メソッド、**StopReadTags**メソッド

CapDisableTag プロパティ

形式 BOOL CapDisableTag;

説明 TRUEならば、デバイスはタグの完全な無効化をサポートします。 アプリケーションは**DisableTag**メソッドを呼び出す前にこのプロパティを確認するべきです。

このプロパティは、**Open**メソッドにより初期化されます。

参照 **DisableTag**メソッド

CapLockTag プロパティ

形式 BOOL CapLockTag;

説明 TRUEならば、デバイスはタグのロックをサポートします。 アプリケーションは**LockTag**メソッドを呼び出す前にこのプロパティを確認するべきです。

このプロパティは、**Open**メソッドにより初期化されます。

参照 **LockTag**メソッド

CapMultipleProtocols プロパティ

形式　LONG CapMultipleProtocols;

説明　このプロパティはこのデバイスがサポートする利用可能な事前に定義されたRFIDタグプロトコルを示します。デバイスが 1 つ以上のプロトコルをサポートしているならば、このプロパティはサポートしているすべてのプロトコルの値のビット単位の論理和です。

値	意味
RFID_PR_EPC0	EPCクラス0 リードオンリーパッシブタグ
RFID_PR_0PLUS	非標準 EPCクラス”0+” ライトワンスパッシブタグ
RFID_PR_EPC1	EPCクラス1 ライトワンスパッシブタグ
RFID_PR_EPC1G2	EPCクラス1 gen2(ISO 18000-6C) ライトワンスパッシブタグ
RFID_PR_EPC2	EPCクラス2 リライタブルタグ
RFID_PR_ISO14443A	ISO 14443A HF タグ
RFID_PR_ISO14443B	ISO 14443B HF タグ
RFID_PR_ISO15693	ISO 15693 HF タグ
RFID_PR_ISO180006B	ISO 18000-6B UHF タグ
RFID_PR_OTHER	定義されたプロトコルの1つに収まらないタグ

このプロパティに基づいて、**ProtocolMask**でビットをオフにすることによって除外したいタグをフィルターにかけることができます。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CurrentTag**プロパティ、**ProtocolMask**プロパティ

CapReadTimer プロパティ

形式 BOOL CapReadTimer;

説明 TRUEならば、デバイスは読み取りタイマーをサポートします。 アプリケーションはReadTimerIntervalを設定する前にこのプロパティを確認するべきです。

このプロパティは、Openメソッドにより初期化されます。

参照 ReadTimerIntervalプロパティ

CapWriteTag プロパティ

形式 LONG CapWriteTag;

説明 タグの書き込み可能なフィールドを示します。 可能な値は以下の通り：

値	意味
RFID_CWT_NONE	タグに書き込み可能なフィールド無し(0)
RFID_CWT_ID	タグのIDフィールドが書き込み可能(1)
RFID_CWT_USERDATA	タグのユーザーデータフィールドが書き込み可能(2)
RFID_CWT_ALL	タグの全てのフィールドが書き込み可能(3)

このプロパティの値はデバイスの書き込み能力を示すだけで、どんな種類のタグでも書き込めることを意味しません。アプリケーションは書き込みメソッドを呼び出す前に、このプロパティを確認するべきです。

このプロパティは、Openメソッドにより初期化されます。

参照 WriteTagDataメソッド、WriteTagIDメソッド

ContinuousReadMode プロパティ

形式　BOOL ContinuousReadMode;

説明　TRUEならば、デバイスは連続読み取りモード中です。 このプロパティがTRUEのときは、**ProtocolMask**と**ReadTimerInterval**のプロパティはリードオンリーです。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CapContinuousRead**プロパティ

CurrentTagID プロパティ

形式　BSTR CurrentTagID;

説明　このプロパティは現在のタグのタグIDをあらわします。
このデータのフォーマットは**BinaryConversion**プロパティの値に依存します。詳細は**BinaryConversion**プロパティを参照してください。

コントロールは**DataEvent**を通知する直前に、読み取った最初のタグのデータをこのプロパティに置きます。 コントロールはこのプロパティを保持し、アプリケーションが、**FirstTag**、**NextTag**、および**PreviousTag**メソッドを呼んだときに更新します。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**FirstTag**メソッド、**NextTags**メソッド、**PreviousTag**メソッド、**ReadTags**メソッド、**StartReadTags**メソッド

CurrentTagProtocol プロパティ

形式 LONG CurrentTagProtocol;

説明 コントロールはこのタグを読み取った時のプロトコルをこのプロパティに置きます。この値はProtocolMaskの中から１つが選ばれます。このプロパティはそれぞれの個々のタグのためにコントロールによって更新されます。

コントロールはDataEventを通知する直前に、読み取った最初のタグのデータをこのプロパティに置きます。 コントロールはこのプロパティを保持し、アプリケーションが、FirstTag、NextTag、およびPreviousTagメソッドを呼んだときに更新します。

このプロパティは、Openメソッドによりゼロに初期化されます。

参照 ProtocolMaskプロパティ、FirstTagメソッド、NextTagsメソッド、PreviousTagメソッド、ReadTagsメソッド、StartReadTagsメソッド

CurrentTagUserData プロパティ

形式 BSTR CurrentTagUserData;

説明 コントロールはこのプロパティに物理的なタグから読み取ったデータを置きます。 部分的な読み出しであるなら、それは対象とされたセグメントを置きます。
このデータのフォーマットはBinaryConversionプロパティの値に依存します。詳細はBinaryConversionプロパティを参照してください。

コントロールはDataEventを通知する直前に、読み取った最初のタグのデータをこのプロパティに置きます。 コントロールはこのプロパティを保持し、アプリケーションが、FirstTag、NextTag、およびPreviousTagメソッドを呼んだときに更新します。

このプロパティは、Openメソッドにより初期化されます。

参照 FirstTagメソッド、NextTagsメソッド、PreviousTagメソッド、ReadTagsメソッド、StartReadTagsメソッド

ProtocolMaskプロパティ　R/W

形式　　**LONG ProtocolMask;**

説明　　このプロパティの個々のビットは事前に定義されたRFIDタグプロトコルを示します。ゼロで無いビットエントリは読み取りを要求しているプロトコルを示します。指定されたタイプのプロトコルのタグだけを読み取ります。読み取り機が連続読み取りモード中なら、このプロパティはリードオンリーです。

値	意味
RFID_PR_EPC0	EPCクラス0 リードオンリーパッシブタグ
RFID_PR_0PLUS	非標準 EPCクラス”0+” ライトワンスパッシブタグ
RFID_PR_EPC1	EPCクラス1 ライトワンスパッシブタグ
RFID_PR_EPC1G2	EPCクラス1 gen2(ISO 18000-6C) ライトワンスパッシブタグ
RFID_PR_EPC2	EPCクラス2 リライタブルタグ
RFID_PR_ISO14443A	ISO 14443A HF タグ
RFID_PR_ISO14443B	ISO 14443B HF タグ
RFID_PR_ISO15693	ISO 15693 HF タグ
RFID_PR_ISO180006B	ISO 18000-6B UHF タグ
RFID_PR_OTHER	定義されたプロトコルの1つに収まらないタグ
RFID_PR_ALL	リーダーでサポートしているすべてのタグ

このプロパティは、**Open**メソッドにより**CapMultipleProtocols**プロパティと同じ値に初期化されます。そして通常、アプリケーションによって初期化処理中に更新されます。

参照　　**CapMultipleProtocols**プロパティ、**CurrentTagProtocol**プロパティ

ReadTimerIntervalプロパティ　R/W

形式　LONG ReadTimerInterval;

説明　個々のタグの読み取りの間の最小間隔をミリ秒であらわします。このプロパティは連続読み取りにのみ適用されます。値がゼロなら読み取り間にディレイが無いことをあらわします。**CapReadTimer**がFALSEであるなら、このプロパティの値はゼロです。**CapReadTimer**がFALSEか**ContinuousReadMode**がTRUEの時にこのプロパティを設定しようとするとエラーになります。

このプロパティは、**Open**メソッドによりゼロに初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	意味
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	デバイスは**ReadTimerInterval**の変更をサポートしていません。

参照　**CapReadTimer**プロパティ

TagCount プロパティ

形式　LONG TagCount;

説明　このプロパティは対応する読み取り操作で読まれたタグの総数をあらわしています。

本プロパティは、**DataEvent**が通知される直前にコントロールによって設定されます。

このプロパティは、**Open**メソッドによりゼロに初期化されます。

参照　**ReadTags**メソッド、**StartReadTags**メソッド、**DataEvent**イベント

メソッド

DisableTag メソッド

形式　LONG DisableTag (BSTR *TagID*, LONG *Timeout*, BSTR *Password*);

パラメータ	説明
TagID	無効化するタグのタグID。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
Timeout	メソッドの失敗やタイムアウトの**ErrorEvent**をアプリケーションに通知する前に許容された実行時間をミリ秒であらわします。OPOS_FOREVER(-1)の場合、このメソッドは操作が完了するかエラーが発生するまで待ちます。
Password	この操作が必要とする読み取り機の認証キー。使用しないならば長さ0(または空)。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

説明　*TagID*パラメタに合う特定のタグを完全に無効化します。
このメソッドは常に非同期で実行されます、**OutputID**は処理の開始に成功すると設定され、完了を示す**OutputCompleteEvent**か**ErrorEvent**が通知されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に終了しました。
OPOS_E_ILLEGAL	機能をサポートしていません。
その他	**ResultCode**を参照のこと。

参照　「**出力モデルについて**」

FirstTag メソッド

形式 LONG FirstTag ();

説明 タグリストにおける最初のタグにコントロールのカウンタをリセットして、対応するプロパティにそのタグの情報をコピーします。

アプリケーションがタグのリストの最初から再処理を必要とする場合に使用されます。

このメソッドが呼ばれた時には物理的な入出力は発生しないため同期で実行されます。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に終了しました。
その他	**ResultCode**を参照のこと。

参照 **CurrentTagID**プロパティ、**CurrentTagProtocol**プロパティ、**CurrentTagUserData**プロパティ、**TagCount**プロパティ

LockTag メソッド

形式　LONG LockTag (BSTR *TagID*, LONG *Timeout*, BSTR *Password*);

パラメータ	説明
TagID	ロックするタグのタグID。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
Timeout	メソッドの失敗やタイムアウトの**ErrorEvent**をアプリケーションに通知する前に許容された実行時間をミリ秒であらわします。OPOS_FOREVER(-1)の場合、このメソッドは操作が完了するかエラーが発生するまで待ちます。
Password	この操作が必要とする読み取り機の認証キー。使用しないならば長さ0(または空)。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

説明　この操作はタグをIDとユーザーデータフィールドの両方とも書き込めない読み取り専用モードにします。**CapLockTag**がFALSEの時にこのメソッドを呼び出した場合、エラーが返ります。
このメソッドは常に非同期で実行されます、**OutputID**は処理の開始に成功すると設定され、完了を示す**OutputCompleteEvent**か**ErrorEvent**が通知されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に終了しました。
OPOS_E_ILLEGAL	機能をサポートしていません。
その他	**ResultCode**を参照のこと。

参照　「**出力モデルについて**」

NextTag メソッド

形式 LONG NextTag ();

説明 タグリストにおける次のタグにコントロールのカウンタを動かして、対応するプロパティにそのタグの情報をコピーします。

通常のRFID読み取り処理で使用されます。

このメソッドが呼ばれた時には物理的な入出力は発生しないため同期で実行され、メモリのコピーだけが行われます。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に終了しました。
その他	**ResultCode**を参照のこと。

参照 **CurrentTagID**プロパティ、**CurrentTagProtocol**プロパティ、**CurrentTagUserData**プロパティ、**TagCount**プロパティ

PreviousTag メソッド

形式 LONG PreviousTag ();

説明 タグリストにおける前のタグにコントロールのカウンタを動かして、対応するプロパティにそのタグの情報をコピーします。

アプリケーションが逆順でタグのリストを処理する必要がある場合に使用されます。

このメソッドが呼ばれた時には物理的な入出力は発生しないため同期で実行され、メモリのコピーだけが行われます。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に終了しました。
その他	**ResultCode**を参照のこと。

参照 **CurrentTagID**プロパティ、**CurrentTagProtocol**プロパティ、**CurrentTagUserData**プロパティ、**TagCount**プロパティ

ReadTags メソッド

形式　LONG ReadTags (LONG *Cmd*, BSTR *FilterID*, BSTR *Filtermask*, LONG *Start*, LONG *Length*, LONG *Timeout*, BSTR *Password*);

パラメータ	説明
Cmd	読み取る内容を指定します。下記の値を参照。読み取り機の中にはIDだけを要求された場合にのみ高速に読み取れるものもあります。
FilterID	読み取るタグを決定する際に*Filtermask*との論理積を行うためのビットパターンを保持します。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
Filtermask	読み取るタグを決定する際に*FilterID*との論理積を行うためのビットパターンを保持します。この*Filtermask*と*FilterID*を論理積したものにマッチするタグだけを通知します。フィールドにあるすべてのタグを取得するには、すべて0の*Filtermask*を指定してください。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
Start	ユーザーデータを読み始める場所を0から始まる位置で指定します。このパラメタは*Cmd*がRFID_RT_PARTIALUSERDATAかRFID_RT_ID_PARTIALUSERDATAの時だけ適用され、それ以外では無視されます。
Length	*Start*パラメータで指定した位置から読み取るユーザーデータの長さをバイト数で指定します。このパラメタは*Cmd*がRFID_RT_PARTIALUSERDATAかRFID_RT_ID_PARTIALUSERDATAの時だけ適用され、それ以外では無視されます。
Timeout	メソッドの失敗やタイムアウトの**ErrorEvent**をアプリケーションに通知する前に許容された実行時間をミリ秒であらわします。OPOS_FOREVER(-1)の場合、このメソッドは操作が完了するかエラーが発生するまで待ちます。

Password この操作が必要とする読み取り機の認証キー。使用しないならば長さ0(または空)。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

*Cmd*パラメータの値は、

値	意味
RFID_RT_ID	IDのみ読み取り。
RFID_RT_FULLUSERDATA	すべてのユーザーデータを読み取り。
RFID_RT_PARTIALUSERDATA	指定された部分的なユーザーデータの読み取り。
RFID_RT_ID_FULLUSERDATA	IDとすべてのユーザーデータを読み取り。
RFID_RT_ID_PARTIALUSERDATA	IDと指定された部分的なユーザーデータの読み取り。

説明 パラメータで指定した評価基準を満たす、レンジ内のすべてのタグの読み取りを実行します。

別の**ReadTags**を実施する前に**DataEvent**か**ErrorEvent**を受け取らなければなりません。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に終了しました。
その他	**ResultCode**を参照のこと。

参照 **TagCount**プロパティ、**FirstTag**メソッド、**NextTag**メソッド、**PreviousTag**メソッド、「**入力モデルについて**」

StartReadTags メソッド

形式　LONG StartReadTags (LONG *Cmd*, BSTR *FilterID*, BSTR *Filtermask*, LONG *Start*, LONG *Length*, LONG *Timeout*, BSTR *Password*);

パラメータ	説明
Cmd	読み取る内容を指定します。下記の値を参照。読み取り機の中にはIDだけを要求された場合にのみ高速に読み取れるものもあります。
FilterID	読み取るタグを決定する際に*Filtermask*との論理積を行うためのビットパターンを保持します。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
Filtermask	読み取るタグを決定する際に*FilterID*との論理積を行うためのビットパターンを保持します。この*Filtermask*と*FilterID*を論理積したものにマッチするタグだけを通知します。フィールドにあるすべてのタグを取得するには、すべて0の*Filtermask*を指定してください。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
Start	ユーザーデータを読み始める場所を0から始まる位置で指定します。このパラメタは*Cmd*がRFID_RT_PARTIALUSERDATAかRFID_RT_ID_PARTIALUSERDATAの時だけ適用され、それ以外では無視されます。
Length	*Start*パラメータで指定した位置から読み取るユーザーデータの長さをバイト数で指定します。このパラメタは*Cmd*がRFID_RT_PARTIALUSERDATAかRFID_RT_ID_PARTIALUSERDATAの時だけ適用され、それ以外では無視されます。

Timeout	メソッドの失敗やタイムアウトの**ErrorEvent**をアプリケーションに通知する前に許容された実行時間をミリ秒であらわします。OPOS_FOREVER(-1)の場合、このメソッドは操作が完了するかエラーが発生するまで待ちます。
Password	この操作が必要とする読み取り機の認証キー。使用しないならば長さ0(または空)。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

*Cmd*パラメータの値は、

値	意味
RFID_RT_ID	IDのみ読み取り。
RFID_RT_FULLUSERDATA	すべてのユーザーデータを読み取り。
RFID_RT_PARTIALUSERDATA	指定された部分的なユーザーデータの読み取り。
RFID_RT_ID_FULLUSERDATA	IDとすべてのユーザーデータを読み取り。
RFID_RT_ID_PARTIALUSERDATA	IDと指定された部分的なユーザーデータの読み取り。

説明

パラメータで指定された評価基準を満たすタグの連続読み取りを実行します。

それぞれの読み取り操作で**DataEvent**か**ErrorEvent**のどちらかが通知されます。

コントロールは、**StopReadTags**が呼び出されるまで読み取りを継続します。

StartReadTagsを実行して**ContinuousReadMode**がTRUEの間に、別の**StartReadTags**を呼び出すとエラーを返します。

このメソッドは常に非同期で実行されますが、**OutputID**は設定されず、完了を示す**OutputCompleteEvent**も通知されません。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に終了しました。
OPOS_E_ ILLEGAL	連続読み取り機能をサポートしていません。
その他	**ResultCode**を参照のこと。

参照　「**入力モデルについて**」、**StopReadTags**メソッド

StopReadTags メソッド

形式　**LONG StopReadTags (BSTR** ***Password*****);**

パラメータ	説明
Password	この操作が必要とする読み取り機の認証キー。使用しないならば長さ0(または空)。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

説明　連続読み取りモードを終了します。

連続読み取りモードによってリードオンリーになっていたすべてのプロパティが再び書き込み可能になります。

連続読み取りモードでは無い時に呼び出した場合はエラーを返します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に終了しました。
OPOS_E_ ILLEGAL	連続読み取りモードではありません。
その他	**ResultCode**を参照のこと。

参照　「**入力モデルについて**」、**StartReadTags**メソッド

WriteTagData メソッド

形式 LONG WriteTagData (BSTR *TagID*, BSTR *Userdata*, LONG *Start*, LONG *Timeout*, BSTR *Password*);

パラメータ	説明
TagID	*Userdata*を書き込むタグのタグID。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
Userdata	書き込むデータ。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
Start	*Userdata*を書き始める場所を0から始まる位置で指定します。
Timeout	メソッドの失敗やタイムアウトの**ErrorEvent** をアプリケーションに通知する前に許容された実行時間をミリ秒であらわします。OPOS_FOREVER(-1)の場合、このメソッドは操作が完了するかエラーが発生するまで待ちます。
Password	この操作が必要とする読み取り機の認証キー。使用しないならば長さ0(または空)。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

説明 対象とされたタグのユーザーデータフィールドの全体か一部を上書きします。アプリケーションはこの操作のサポートについて**CapWriteTag**を調べるべきです。
このメソッドは常に非同期で実行されます、**OutputID**は処理の開始に成功すると設定され、完了を示す**OutputCompleteEvent**か**ErrorEvent**が通知されます。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に終了しました。
OPOS_E_ILLEGAL	機能をサポートしていません。
その他	**ResultCode**を参照のこと。

参照　　「出力モデルについて」、CapWriteTagプロパティ

WriteTagID メソッド

形式 LONG WriteTagID (BSTR *SourceID*, BSTR *DestID*, LONG *Timeout*, BSTR *Password*);

パラメータ	説明
SourceID	新しいIDを書き込むタグのタグID。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
DestID	新しいID。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。
Timeout	メソッドの失敗やタイムアウトの**ErrorEvent** をアプリケーションに通知する前に許容された実行時間をミリ秒であらわします。OPOS_FOREVER(-1)の場合、このメソッドは操作が完了するかエラーが発生するまで待ちます。
Password	この操作が必要とする読み取り機の認証キー。使用しないならば長さ0(または空)。このデータのフォーマットは、**BinaryConversion** プロパティの値に依存します。詳細は、**BinaryConversion** を参照してください。

説明 既存のタグIDを新しいIDで上書きします。アプリケーションはこのメソッドがサポートされていることを検証するために**CapWriteTag**を調べるべきです。**CapWriteTag**のRFID_CWT_IDビットがオフの時にこのメソッドを呼び出すと、エラーを返します。

このメソッドは常に非同期で実行されます、**OutputID**は処理の開始に成功すると設定され、完了を示す**OutputCompleteEvent**か**ErrorEvent**が通知されます。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に終了しました。
OPOS_E_ILLEGAL	機能をサポートしていません。
その他	**ResultCode**を参照のこと。

参照　　「出力モデルについて」、CapWriteTagプロパティ

イベント

DataEvent イベント

形式　　void DataEvent (LONG *Status*);

*Status*パラメータはゼロが設定されます。

説明　　RFIDスキャナからアプリケーションに、入力データを送るために通知されます。このイベントが通知される前に、**TagCount**プロパティが更新されます。このイベントで通知されたタグの詳細に関して、アプリケーションはこのイベントの中で個々のタグを列挙するために、**FirstTag, NextTag**メソッドを呼び出し、次に一連の**CurrentTagXXX**プロパティを調べてください。

参照　　**CurrentTagID**プロパティ、**CurrentTagProtocol**プロパティ、**CurrentTagUserData**プロパティ、**TagCount**プロパティ

ErrorEvent イベント

形式　**void ErrorEvent (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**ResultCode**の項目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**ResultCodeExtended**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が、すでにバッファリングされたデータの一部は利用できます。

*pErrorResponse*パラメータが示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_CLEAR	バッファリングされている入力データをクリアし、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、**DataEventEnabled**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、**DataEventEnabled**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。

OPOS_ER_RETRY　通常は、*ErrorLocus*がOPOS_EL_OUTPUTの場合に使用します。
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の場合でも指定できる場合があります。
*ErrorLocus*がOPOS_EL_OUTPUTの場合のデフォルトです。

説明　RFIDスキャナデータを読み込もうとしている間に、エラーを検出すると通知されます。

正しいアプリケーションシーケンシングが行われるよう、**DataEventEnabled**プロパティがTRUEになるまで、入力エラーイベントは通知されません。

参照　「**ステータス、結果コード、状態モデル**」

第　３４章
チェックスキャナ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.7	Boolean	R/W	*適用外*
BinaryConversion	1.7	Long	R/W	Open
CapCompareFirmwareVersion	1.9	Boolean	R	Open
CapPowerReporting	1.7	Long	R	Open
CapStatisticsReporting	1.8	Boolean	R	Open
CapUpdateFirmware	1.9	Boolean	R	Open
CapUpdateStatistics	1.8	Boolean	R	Open
CheckHealthText	1.7	String	R	Open
Claimed	1.7	Boolean	R	Open
DataCount	1.7	Long	R	*適用外*
DataEventEnabled	1.7	Boolean	R/W	*適用外*
DeviceEnabled	1.7	Boolean	R/W	Open,Claim
FreezeEvents	1.7	Boolean	R/W	Open
OpenResult	1.7	Long	R	無し
OutputID	1.7	Long	R	*適用外*
PowerNotify	1.7	Long	R/W	Open
PowerState	1.7	Long	R	Open
ResultCode	1.7	Long	R	無し
ResultCodeExtended	1.7	Long	R	Open
State	1.7	Long	R	無し
ControlObjectDescription	1.7	String	R	無し
ControlObjectVersion	1.7	Long	R	無し
ServiceObjectDescription	1.7	String	R	Open
ServiceObjectVersion	1.7	Long	R	Open
DeviceDescription	1.7	String	R	Open
DeviceName	1.7	String	R	Open

プロパティ (続き)

専用	*版数*	*型*	*アクセスs*	*使用可能条件*
CapAutoContrast	1.9	Boolean	R	Open
CapAutoGenerateFileID	1.7	Boolean	R	Open
CapAutoGenerateImageTagData	1.7	Boolean	R	Open
CapAutoSize	1.7	Boolean	R	Open
CapColor	1.7	Long	R	Open
CapConcurrentMICR	1.7	Boolean	R	Open
CapContrast	1.9	Boolean	R	Open
CapDefineCropArea	1.7	Boolean	R	Open
CapImageFormat	1.7	Long	R	Open
CapImageTagData	1.7	Boolean	R	Open
CapMICRDevice	1.7	Boolean	R	Open
CapStoreImageFiles	1.7	Boolean	R	Open
CapValidationDevice	1.7	Boolean	R	Open
Color	1.7	Long	R/W	Open
ConcurrentMICR	1.7	Boolean	R/W	Open
Contrast	1.9	Long	R/W	Open, & Enable
CropAreaCount	1.7	Long	R	Open
DocumentHeight	1.7	Long	R/W	Open
DocumentWidth	1.7	Long	R/W	Open
FileID	1.7	String	R/W	Open
FileIndex	1.7	Long	R/W	Open
ImageData	1.7	String	R	Open
ImageFormat	1.7	Long	R/W	Open
ImageMemoryStatus	1.7	Long	R	Open, Claim
ImageTagData	1.7	String	R/W	Open
MapMode	1.7	Long	R/W	Open
MaxCropAreas	1.7	Long	R	Open
Quality	1.7	Long	R/W	Open
QualityList	1.7	String	R	Open
RemainingImagesEstimate	1.7	Long	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.7	無し
Close	1.7	Open
ClaimDevice	1.7	Open
ReleaseDevice	1.7	Open, Claim
CheckHealth	1.7	Open, Claim, & Enable
ClearInput	1.7	Open, Claim
ClearInputProperties	1.10	Open, Claim
ClearOutput	1.7	*適用外*
CompareFirmwareVersion	1.9	Open, Claim, & Enable
DirectIO	1.7	Open
ResetStatistics	1.8	Open, Claim, & Enable
RetrieveStatistics	1.8	Open, Claim, & Enable
UpdateFirmware	1.9	Open, Claim, & Enable
UpdateStatistics	1.8	Open, Claim, & Enable
専用		
BeginInsertion	1.7	Open, Claim, & Enable
BeginRemoval	1.7	Open, Claim, & Enable
ClearImage	1.7	Open, Claim, & Enable
DefineCropArea	1.7	Open, Claim, & Enable
EndInsertion	1.7	Open, Claim, & Enable
EndRemoval	1.7	Open, Claim, & Enable
RetrieveImage	1.7	Open, Claim, & Enable
RetrieveMemory	1.7	Open, Claim, & Enable
StoreImage	1.7	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.7	Open, Claim, & Enable
DirectIOEvent	1.7	Open, Claim
ErrorEvent	1.7	Open, Claim, & Enable
OutputCompleteEvent	1.7	*適用外*
StatusUpdateEvent	1.7	Open, Claim, & Enable

概説

チェックスキャナコントロールのOLEプログラマティクIDは“OPOS.CheckScanner”です。

このデバイスは、第1.7 版で追加されました。

詳細はUPOS仕様書Version1.14を参照ください。

第 35 章
ベルト

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.12	Boolean	R/W	*適用外*
BinaryConversion	1.12	Long	R/W	Open
CapCompareFirmwareVersion	1.12	Boolean	R	Open
CapPowerReporting	1.12	Long	R	Open
CapStatisticsReporting	1.12	Boolean	R	Open
CapUpdateFirmware	1.12	Boolean	R	Open
CapUpdateStatistics	1.12	Boolean	R	Open
CheckHealthText	1.12	String	R	Open
Claimed	1.12	Boolean	R	Open
DataCount	1.12	Long	R	*適用外*
DataEventEnabled	1.12	Boolean	R/W	*適用外*
DeviceEnabled	1.12	Boolean	R/W	Open,Claim
FreezeEvents	1.12	Boolean	R/W	Open
OpenResult	1.12	Long	R	無し
OutputID	1.12	Long	R	*適用外*
PowerNotify	1.12	Long	R/W	Open
PowerState	1.12	Long	R	Open
ResultCode	1.12	Long	R	無し
ResultCodeExtended	1.12	Long	R	Open
State	1.12	Long	R	無し
ControlObjectDescription	1.12	String	R	無し
ControlObjectVersion	1.12	Long	R	無し
ServiceObjectDescription	1.12	String	R	Open
ServiceObjectVersion	1.12	Long	R	Open
DeviceDescription	1.12	String	R	Open
DeviceName	1.12	String	R	Open

プロパティ (続き)

専用	*版数*	*型*	*アクセスs*	*使用可能条件*
CapAutoStopBackward	1.12	Boolean	R	Open
CapAutoStopBackwardItemCount	1.12	Boolean	R	Open
CapAutoStopForward	1.12	Boolean	R	Open
CapAutoStopForwardItemCount	1.12	Boolean	R	Open
CapLightBarrierBackward	1.12	Boolean	R	Open
CapLightBarrierForward	1.12	Boolean	R	Open
CapMoveBackward	1.12	Boolean	R	Open
CapSecurityFlapBackward	1.12	Boolean	R	Open
CapSecurityFlapForward	1.12	Boolean	R	Open
CapSpeedStepsBackward	1.12	Long	R	Open
CapSpeedStepsForward	1.12	Long	R	Open
AutoStopBackward	1.12	Boolean	R/W	Open
AutoStopBackwardDelayTime	1.12	Long	R/W	Open
AutoStopBackwardItemCount	1.12	Long	R	Open
AutoStopForward	1.12	Boolean	R/W	Open
AutoStopForwardDelayTime	1.12	Long	R/W	Open
AutoStopForwardItemCount	1.12	Long	R	Open
LightBarrierBackwardInterrupted	1.12	Boolean	R	Open, Claim, & Enable
LightBarrierForwardInterrupted	1.12	Boolean	R	Open, Claim, & Enable
MotionStatus	1.12	Long	R	Open, Claim, & Enable
SecurityFlapBackwardOpened	1.12	Boolean	R	Open, Claim, & Enable
SecurityFlapForwardOpened	1.12	Boolean	R	Open, Claim, & Enable

メソッド

共通	*版数*	*使用可能条件*
Open	1.12	無し
Close	1.12	Open
ClaimDevice	1.12	Open
ReleaseDevice	1.12	Open, Claim
CheckHealth	1.12	Open, Claim, & Enable
ClearInput	1.12	*適用外*
ClearInputProperties	1.12	*適用外*
ClearOutput	1.12	*適用外*
CompareFirmwareVersion	1.12	Open, Claim, & Enable
DirectIO	1.12	Open
ResetStatistics	1.12	Open, Claim, & Enable
RetrieveStatistics	1.12	Open, Claim, & Enable
UpdateFirmware	1.12	Open, Claim, & Enable
UpdateStatistics	1.12	Open, Claim, & Enable
専用		
AdjustItemCount	1.12	Open, Claim, & Enable
MoveBackward	1.12	Open, Claim, & Enable
MoveForward	1.12	Open, Claim, & Enable
ResetBelt	1.12	Open, Claim, & Enable
ResetItemCount	1.12	Open, Claim, & Enable
StopBelt	1.12	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.12	*適用外*
DirectIOEvent	1.12	Open, Claim
ErrorEvent	1.12	*適用外*
OutputCompleteEvent	1.12	*適用外*
StatusUpdateEvent	1.12	Open, Claim, & Enable

概説

ベルトコントロールのOLEプログラマティクIDは “OPOS.Belt”です。

このデバイスは、第1.12 版で追加されました。

詳細はUPOS仕様書Version1.14を参照ください。

第 36 章
ゲート

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.12	Boolean	R/W	*適用外*
BinaryConversion	1.12	Long	R/W	Open
CapCompareFirmwareVersion	1.12	Boolean	R	Open
CapPowerReporting	1.12	Long	R	Open
CapStatisticsReporting	1.12	Boolean	R	Open
CapUpdateFirmware	1.12	Boolean	R	Open
CapUpdateStatistics	1.12	Boolean	R	Open
CheckHealthText	1.12	String	R	Open
Claimed	1.12	Boolean	R	Open
DataCount	1.12	Long	R	*適用外*
DataEventEnabled	1.12	Boolean	R/W	*適用外*
DeviceEnabled	1.12	Boolean	R/W	Open,Claim
FreezeEvents	1.12	Boolean	R/W	Open
OpenResult	1.12	Long	R	無し
OutputID	1.12	Long	R	*適用外*
PowerNotify	1.12	Long	R/W	Open
PowerState	1.12	Long	R	Open
ResultCode	1.12	Long	R	無し
ResultCodeExtended	1.12	Long	R	Open
State	1.12	Long	R	無し
ControlObjectDescription	1.12	String	R	無し
ControlObjectVersion	1.12	Long	R	無し
ServiceObjectDescription	1.12	String	R	Open
ServiceObjectVersion	1.12	Long	R	Open
DeviceDescription	1.12	String	R	Open
DeviceName	1.12	String	R	Open

プロパティ **(続き)**

専用	*版数*	*型*	*アクセスs*	*使用可能条件*
CapGateStatus	1.12	Boolean	R	Open
GateStatus	1.12	Long	R	Open,& Enable

メソッド

共通	*版数*	*使用可能条件*
Open	1.12	無し
Close	1.12	Open
ClaimDevice	1.12	Open
ReleaseDevice	1.12	Open, Claim
CheckHealth	1.12	Open, & Enable
ClearInput	1.12	*適用外*
ClearInputProperties	1.12	*適用外*
ClearOutput	1.12	*適用外*
CompareFirmwareVersion	1.12	Open, & Enable
DirectIO	1.12	Open
ResetStatistics	1.12	Open, & Enable
RetrieveStatistics	1.12	Open, & Enable
UpdateFirmware	1.12	Open, & Enable
UpdateStatistics	1.12	Open, & Enable
専用		
OpenGate	1.12	Open, & Enable
WaitForGateClose	1.12	Open, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.12	*適用外*
DirectIOEvent	1.12	Open
ErrorEvent	1.12	*適用外*
OutputCompleteEvent	1.12	*適用外*
StatusUpdateEvent	1.12	Open, & Enable

概説

ゲートコントロールのOLEプログラマティックIDは “OPOS.Gate”です。

このデバイスは、第1.12 版で追加されました。

詳細はUPOS仕様書Version1.14を参照ください。

第　３７章
アイテムディスペンサ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.12	Boolean	R/W	*適用外*
BinaryConversion	1.12	Long	R/W	Open
CapCompareFirmwareVersion	1.12	Boolean	R	Open
CapPowerReporting	1.12	Long	R	Open
CapStatisticsReporting	1.12	Boolean	R	Open
CapUpdateFirmware	1.12	Boolean	R	Open
CapUpdateStatistics	1.12	Boolean	R	Open
CheckHealthText	1.12	String	R	Open
Claimed	1.12	Boolean	R	Open
DataCount	1.12	Long	R	*適用外*
DataEventEnabled	1.12	Boolean	R/W	*適用外*
DeviceEnabled	1.12	Boolean	R/W	Open,Claim
FreezeEvents	1.12	Boolean	R/W	Open
OpenResult	1.12	Long	R	無し
OutputID	1.12	Long	R	*適用外*
PowerNotify	1.12	Long	R/W	Open
PowerState	1.12	Long	R	Open
ResultCode	1.12	Long	R	無し
ResultCodeExtended	1.12	Long	R	Open
State	1.12	Long	R	無し
ControlObjectDescription	1.12	String	R	無し
ControlObjectVersion	1.12	Long	R	無し
ServiceObjectDescription	1.12	String	R	Open
ServiceObjectVersion	1.12	Long	R	Open
DeviceDescription	1.12	String	R	Open
DeviceName	1.12	String	R	Open

プロパティ (続き)

専用	*版数*	*型*	*アクセスs*	*使用可能条件*
CapEmptySensor	1.12	Boolean	R	Open
CapIndividualSlotStatus	1.12	Boolean	R	Open
CapJamSensor	1.12	Boolean	R	Open
CapNearEmptySensor	1.12	Boolean	R	Open
DispenserStatus	1.12	Long	R	Open, Claim, & Enable
MaxSlots	1.12	Long	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.12	無し
Close	1.12	Open
ClaimDevice	1.12	Open
ReleaseDevice	1.12	Open, Claim
CheckHealth	1.12	Open, Claim, & Enable
ClearInput	1.12	*適用外*
ClearInputProperties	1.12	*適用外*
ClearOutput	1.12	*適用外*
CompareFirmwareVersion	1.12	Open, Claim, & Enable
DirectIO	1.12	Open
ResetStatistics	1.12	Open, Claim, & Enable
RetrieveStatistics	1.12	Open, Claim, & Enable
UpdateFirmware	1.12	Open, Claim, & Enable
UpdateStatistics	1.12	Open, Claim, & Enable
専用		
AdjustItemCount	1.12	Open, Claim, & Enable
DispenseItem	1.12	Open, Claim, & Enable
ReadItemCount	1.12	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.12	*適用外*
DirectIOEvent	1.12	Open, Claim
ErrorEvent	1.12	*適用外*
OutputCompleteEvent	1.12	*適用外*
StatusUpdateEvent	1.12	Open, Claim, & Enable

概説

アイテムディスペンサコントロールのOLEプログラマティックIDは“OPOS.ItemDispnser”です。

このデバイスは、第1.12 版で追加されました。

詳細はUPOS仕様書Version1.14を参照ください。

第 38 章
ライト

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.12	Boolean	R/W	*適用外*
BinaryConversion	1.12	Long	R/W	Open
CapCompareFirmwareVersion	1.12	Boolean	R	Open
CapPowerReporting	1.12	Long	R	Open
CapStatisticsReporting	1.12	Boolean	R	Open
CapUpdateFirmware	1.12	Boolean	R	Open
CapUpdateStatistics	1.12	Boolean	R	Open
CheckHealthText	1.12	String	R	Open
Claimed	1.12	Boolean	R	Open
DataCount	1.12	Long	R	*適用外*
DataEventEnabled	1.12	Boolean	R/W	*適用外*
DeviceEnabled	1.12	Boolean	R/W	Open,Claim
FreezeEvents	1.12	Boolean	R/W	Open
OpenResult	1.12	Long	R	無し
OutputID	1.12	Long	R	*適用外*
PowerNotify	1.12	Long	R/W	Open
PowerState	1.12	Long	R	Open
ResultCode	1.12	Long	R	無し
ResultCodeExtended	1.12	Long	R	Open
State	1.12	Long	R	無し
ControlObjectDescription	1.12	String	R	無し
ControlObjectVersion	1.12	Long	R	無し
ServiceObjectDescription	1.12	String	R	Open
ServiceObjectVersion	1.12	Long	R	Open
DeviceDescription	1.12	String	R	Open
DeviceName	1.12	String	R	Open

プロパティ (続き)

専用	*版数*	*型*	*アクセスs*	*使用可能条件*
CapAlarm	1.12	Long	R	Open
CapBlink	1.12	Boolean	R	Open
CapColor	1.12	Long	R	Open
CapPattern	1.16	Long	R	Open
MaxLights	1.12	Long	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.12	無し
Close	1.12	Open
ClaimDevice	1.12	Open
ReleaseDevice	1.12	Open, Claim
CheckHealth	1.12	Open, Claim, & Enable
ClearInput	1.12	*適用外*
ClearInputProperties	1.12	*適用外*
ClearOutput	1.12	*適用外*
CompareFirmwareVersion	1.12	Open, Claim, & Enable
DirectIO	1.12	Open
ResetStatistics	1.12	Open, Claim, & Enable
RetrieveStatistics	1.12	Open, Claim, & Enable
UpdateFirmware	1.12	Open, Claim, & Enable
UpdateStatistics	1.12	Open, Claim, & Enable
専用		
SwitchOff	1.12	Open, Claim, & Enable
switchOffPattern	1.16	Open, Claim, & Enable
SwitchOn	1.12	Open, Claim, & Enable
switchOnMultiple	1.16	Open, Claim, & Enable
switchOnPattern	1.16	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.12	*適用外*
DirectIOEvent	1.12	Open, Claim
ErrorEvent	1.12	*適用外*
OutputCompleteEvent	1.12	*適用外*
StatusUpdateEvent	1.12	Open, Claim, & Enable
TransitionEvent	1.15	*適用外*

概説

ライトコントロールのOLEプログラマティクIDは“OPOS.Lights”です。

このデバイスは第1.12版で追加されました。

機能　　*第1.16版で更新*

ライトコントロールは次の機能を持っています。
ライトの「点灯」及び「消灯」コマンド

ライトコントロールは、以下の追加機能を有する場合があります。
調整可能な点滅周期でデバイスレベルの点滅をする機能。
複数のライトをもつ機能。
異なる色の光をもつ機能。
異なるアラームをもと機能。

デバイスの共有

ライトは下記のように排他的に使用すべきデバイスです。デバイスを共有する際のルールは以下のとおりです。

アプリケーションはデバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、幾つかのプロパティとメソッドにアクセスする前、又はイベントを受信する前に、デバイスの排他アクセス権を獲得して、デバイスをイネーブルにしなければなりません。

使用上の詳細な必要条件は、「一覧」を参照してください。

プロパティ

CapAlarm プロパティ

形式 LONG CapAlarm;

説明 デバイスが保持しているアラームのパターン値を示します。
CapAlarm は下記定数で定義されたビットフラグ形式を表します。

値	説明
LGT_ALARM_NOALARM	アラーム機能をもっていません。
LGT_ALARM_SLOW	低速ビープ音の機能をもっています。
LGT_ALARM_MEDIUM	中速ビープ音の機能をもっています。
LGT_ALARM_FAST	高速ビープ音の機能をもっています。
LGT_ALARM_CUSTOM1	第1カスタムアラーム機能をもっています。
LGT_ALARM_CUSTOM2	第２カスタムアラーム機能をもっています。

このプロパティはOpen メソッドにより初期化されます。

デバイスがアラーム機能を保持していない場合、
LGT_ALARM_NOALARMに初期化されます。

CapBlink プロパティ

形式 BOOL CapBlink

説明 TRUE:点滅機能を持ちます。この機能は、デバイス自身が保持するか、サービスによってエミュレートされたものかのいずれかです。
FALSE:点滅機能をもちません。

このプロパティはOpenメソッドにより初期化されます。

CapColor プロパティ

形式 LONG CapColor

説明 デバイスが保持している異なった色パターン値を示します。

CapColorは下記定数で定義されたビットフラグ形式を表します。

値	説明
LGT_COLOR_PRIMARY	プライマリーカラーを保持します。(通常はグリーン).
LGT_COLOR_CUSTOM1	第１カスタムカラーを保持します。(通常はレッド).
LGT_COLOR_CUSTOM2	第2カスタムカラーを保持します。(通常はイエロー).
LGT_COLOR_CUSTOM3	第3カスタムカラーを保持します。
LGT_COLOR_CUSTOM4	第４カスタムカラーを保持します。
LGT_COLOR_CUSTOM5	第５カスタムカラーを保持します。

このプロパティは**Open**メソッドにより初期化されます。
デバイスが1つの色パターンしか持たない場合は、
LGT_COLOR_PRIMARYに初期化されます。

CapPattern プロパティ *第1.16版で追加*

形式 LONG CapPattern

説明 デバイスがサポートしているライトの点灯パターン値を示します。

CapPatternは下記定数で定義されたビットフラグ形式を表します。

値	説明
LGT_PATTERN_NOPATTERN	点灯パターンなし
LGT_PATTERN_CUSTOM	デバイスで定義された点灯パターン

このプロパティは、**Open**メソッドにより初期化されます。デバイスが点灯パターンに対応していない場合は、
LGT_PATTERN_NOPATTERNが設定されます。

参照 **SwitchOnPattern**メソッド

MaxLights プロパティ

形式　　LONG MaxLights

説明　　デバイスが保持できるライトの数の最大値を示します。

このプロパティは**Open**メソッドにより初期化されます。

メソッド

SwitchOff メソッド

形式　LONG SwitchOff (Long *LightNumber*)

パラメーター	説明
LightNumber	ライトの数を指定します。有効なライトの数は、１から**MagLights**迄です。

説明　*LightNumber*で指定された数のライトを消灯します。

戻り値　次の値いずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が渡されました。または*LightNumber*のパラメータ値が**MaxLights**を越えています。
その他	**ResultCode**の項目を参照してください。

参照　**MaxLights** プロパティ

SwitchOffPattern メソッド　　*第1.16 版で追加*

形式　LONG SwitchOffPattern ();

説明　実行中のパターン点灯を停止します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	パターン点灯は実行されていません。
その他	**ResultCode**の項目を参照してください。

参照　**SwitchOnPattern**メソッド

SwitchOn メソッド　　*第1.12 版で更新*

形式　LONG SwitchOn (Long *LightNumber*, Long *BlinkOnCycle*, Long *BlinkOffCycle,* Long *Color*, Long *Alarm*);

パラメータ	説明
LightNumber	ライトの数を指定します。有効なライトの数は、１から**MaxLights**の値までです。
BlinkOnCycle	ゼロ(0)の値は、点滅サイクルがないことを示します。正の値は点滅サイクル時間をミリ秒単位で示します。負の値は許可されません。
BlinkOffCycle	ゼロ(0)の値は、点滅サイクルがないことを示します。正の値は、無点滅サイクル時間をミリ秒単位で示すします。負の値は許可されません。
Color	デバイスで指定されたライトの色値を指定します。**CapColor**で定義される色値の一つである必要があります。
Alarm	使用されるアラームタイプを指定します。、**CapAlarm**で定義される値の一つである必要があります。

説明　*LightNumber*で指定された複数ライトを点灯、或いは、点滅させます。

*BlinkOnCycle*および*BlinkOffCycle*がゼロ(0) または**CapBlink**がfalseの場合、パラメータ*BlinkOnCycle*および*BlinkOffCycle*は無視され、ライトは点灯になります。
CapBlinkがtrueで、*BlinkOnCycle*と*BlinkOffCycle*が正の場合、ライトが点滅します。

CapColorがLGT_COLOR_PRIMARYの場合、ライトは異なる色をサポートせず、色は無視されます。そうでない場合、**SwitchOn**は*Color*で指定された色を使用します。

CapAlarmがLGT_ALARM_NOALARMの場合、ライトは異なるアラームをサポートせず、アラームは無視されます。そうでない場合、**SwitchOn**はアラームで指定されたアラームを使用します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。

OPOS_E_ILLEGAL	*LightNumber*のパラメータ値が**MaxLights**を越えています。または無効な*Color* 又は*Alarm*値が指定されています。
その他	**ResultCode**の項目を参照してください。

参照 **CapAlarm** プロティ, **CapBlink** プロパティ, **CapColor** プロパティ, **MaxLights** プロパティ.

SwitchOnMultiple メソッド *第1.16 版で追加*

形式 **LONG SwitchOnMultiple (BSTR** LightNumbers, **Long** *BlinkOnCycle*, **Long** *BlinkOffCycle*, **Long** *Color*, **Long** *Alarm*);

パラメータ	説明
LightNumbers	ライト番号をカンマ区切りのリストで指定します。有効なライト番号は1～**MaxLights**です。
BlinkOnCycle	ゼロ（0）の値は点滅サイクルがないことを示します。正の値はサイクル時間の点滅をミリ秒単位で示します。負の値は許可されません。
Color	デバイスで指定されたライトの色値を指定します。
Alarm	使用されるアラームタイプを指定します。**CapAlarm**で定義されたアラームタイプの1つである必要があります。

説明 *LightNumber*で指定された複数ライトをオンにするか、点滅させます。
*BlinkOnCycle*および*BlinkOffCycle*がゼロ（0）またはCapBlinkがfalseの場合、パラメータ*BlinkOnCycle*および*BlinkOffCycle*は無視され、ライトはオンになります。CapBlinkがtrueで、*BlinkOnCycle*と*BlinkOffCycle*が正の場合、ライトが点滅します。

CapColorがLGT_COLOR_PRIMARYの場合、ライトは異なる色をサポートせず、色は無視されます。そうでない場合、SwitchOnは*Color*で指定された色値を使用します。

CapAlarmがLGT_ALARM_NOALARMの場合、ライトは異なるアラームをサポートせず、アラームは無視されます。そうでない場合、SwitchOnは*Alarm*で指定されたアラームタイプを使用します。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明

OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照 **CapAlarm**プロパティ、**CapBlink**プロパティ、**CapColor**プロパティ、**MaxLights**プロパティ

SwitchOnPattern メソッド　　*第1.16 版で追加*

形式 **LONG SwitchOnPattern (Long** Pattern, **Long** *Alarm*);

パラメータ	説明
Pattern	パターンを指定します。CapPatternで定義されたパターンのいずれかである必要があります。
Alarm	使用されるアラームタイプを指定します。CapAlarmで定義されたアラームタイプの1つである必要があります。

説明 指定されたパターンでライトをオンにします。

CapAlarmがLGT_ALARM_NOALARMの場合、ライトは異なるアラームをサポートせず、アラームは無視されます。そうでない場合、SwitchOnPatternは*Alarm*で指定されたアラームタイプを使用します。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が渡されました。または、パターン点灯をサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照 **CapAlarm**プロパティ、**CapPattern**プロパティ

イベント

DirectIOEvent イベント

形式 void DirectIOEvent (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。イベント番号とサービスオブジェクトで変化する特定値です。
pString	文字列のポインタ。イベント番号とサービスオブジェクトで変化する特定値です。このデータのフォーマットはBinaryConversionプロパティの値に依存します。詳細はBinaryConversionプロパティを参照してください。

説明 アプリケーションに直接サービス情報を提供します。このイベントは、ベンダー固有のビデオキャプチャサービスが、デバイスコントロールでサポートされていないイベントをアプリケーションに提供する手段を提供します。

アプリケーションと直接通信するために、サービスオブジェクトが通知します。

このイベントにより、コントロールオブジェクトではサポートされないイベントを、サービスオブジェクトがアプリケーションに提供することが可能となります。

参照 「イベント」、DirectIOメソッド

StatusUpdateEvent イベント

形式 void StatusUpdateEvent (LONG *Status*);

以下のパラメータ値が設定されます。

パラメータ	説明
Status	ライトの電源状態の変化を示します。

第1.3版以降では、電源状態通知機能が追加され、電源状態通知のStatusUpdateEvent値が追加されていることに注意してください。

第1.9版以降では、ファームウェア更新の機能追加によって、非同期更新ファームウェア更新処理の状態と進行状況を通知するためのステータス値が追加になりました。共通プロパティ、メソッド、イベントの「StatusUpdateEvent」の説明を参照してください。

説明　ライトの電源状態が変化したときに通知されます。

参照　「イベント」

第 39 章
ビデオキャプチャ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.16	Boolean	R/W	*適用外*
BinaryConversion	1.16	Long	R/W	Open
CapCompareFirmwareVersion	1.16	Boolean	R	Open
CapPowerReporting	1.16	Long	R	Open
CapStatisticsReporting	1.16	Boolean	R	Open
CapUpdateFirmware	1.16	Boolean	R	Open
CapUpdateStatistics	1.16	Boolean	R	Open
CheckHealthText	1.16	String	R	Open
Claimed	1.16	Boolean	R	Open
DataCount	1.16	Long	R	*適用外*
DataEventEnabled	1.16	Boolean	R/W	*適用外*
DeviceEnabled	1.16	Boolean	R/W	Open & Claim
FreezeEvents	1.16	Boolean	R/W	Open
OpenResult	1.16	Long	R	無し
OutputID	1.16	Long	R	*適用外*
PowerNotify	1.16	Long	R/W	Open
PowerState	1.16	Long	R	Open
ResultCode	1.16	Long	R	無し
ResultCodeExtended	1.16	Long	R	Open
State	1.16	Long	R	無し
ControlObjectDescription	1.16	String	R	無し
ControlObjectVersion	1.16	Long	R	無し
ServiceObjectDescription	1.16	String	R	Open
ServiceObjectVersion	1.16	Long	R	Open
DeviceDescription	1.16	String	R	Open
DeviceName	1.16	String	R	Open
専用				
CapAssociatedHardTotalsDevice	1.16	String	R	Open

CapAutoExposure	1.16	Boolean	R	Open
CapAutoFocus	1.16	Boolean	R	Open
CapAutoGain	1.16	Boolean	R	Open
CapAutoWhiteBalance	1.16	Boolean	R	Open
CapBrightness	1.16	Boolean	R	Open
CapContrast	1.16	Boolean	R	Open
CapExposure	1.16	Boolean	R	Open
CapGain	1.16	Boolean	R	Open
CapHorizontalFlip	1.16	Boolean	R	Open
CapHue	1.16	Boolean	R	Open
CapPhoto	1.16	Boolean	R	Open
CapPhotoColorSpace	1.16	Boolean	R	Open
CapPhotoFrameRate	1.16	Boolean	R	Open
CapPhotoResolution	1.16	Boolean	R	Open
CapPhotoType	1.16	Boolean	R	Open
CapSaturation	1.16	Boolean	R	Open
CapStorage	1.16	Boolean	R	Open
CapVerticalFlip	1.16	Boolean	R	Open
CapVideo	1.16	Boolean	R	Open
CapVideoColorSpace	1.16	Boolean	R	Open
CapVideoFrameRate	1.16	Boolean	R	Open
CapVideoResolution	1.16	Boolean	R	Open
CapVideoType	1.16	Boolean	R	Open
AutoExposure	1.16	Boolean	R/W	Open, Claim, & Enable
AutoFocus	1.16	Boolean	R/W	Open, Claim, & Enable
AutoGain	1.16	Boolean	R/W	Open, Claim, & Enable
AutoWhiteBalance	1.16	Boolean	R/W	Open, Claim, & Enable
Brightness	1.16	Long	R/W	Open, Claim, & Enable
Contrast	1.16	Long	R/W	Open, Claim, & Enable
Exposure	1.16	Long	R/W	Open, Claim, & Enable
Gain	1.16	Long	R/W	Open, Claim, & Enable
HorizontalFlip	1.16	Boolean	R/W	Open, Claim, & Enable
Hue	1.16	Long	R/W	Open, Claim, & Enable
PhotoColorSpace	1.16	String	R/W	Open, Claim, & Enable
PhotoColorSpaceList	1.16	String	R	Open
PhotoFrameRate	1.16	Long	R/W	Open, Claim, & Enable
PhotoMaxFrameRate	1.16	Long	R	Open
PhotoResolution	1.16	String	R/W	Open, Claim, & Enable
PhotoResolutionList	1.16	String	R	Open
PhotoType	1.16	String	R/W	Open, Claim, & Enable

PhotoTypeList	1.16	String	R	Open
RemainingRecordingTimeInSec	1.16	Long	R	Open, Claim, & Enable
Saturation	1.16	Long	R/W	Open, Claim, & Enable
Storage	1.16	Long	R/W	Open, Claim, & Enable
VerticalFlip	1.16	Boolean	R/W	Open, Claim, & Enable
VideoCaptureMode	1.16	Long	R/W	Open, Claim, & Enable
VideoColorSpace	1.16	String	R/W	Open, Claim, & Enable
VideoColorSpaceList	1.16	String	R	Open
VideoFrameRate	1.16	Long	R/W	Open, Claim, & Enable
VideoMaxFrameRate	1.16	Long	R	Open
VideoResolution	1.16	String	R/W	Open, Claim, & Enable
VideoResolutionList	1.16	String	R	Open
VideoType	1.16	String	R/W	Open, Claim, & Enable
VideoTypeList	1.16	String	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.16	無し
Close	1.16	Open
ClaimDevice	1.16	Open
ReleaseDevice	1.16	Open & Claim
CheckHealth	1.16	Open, Claim, & Enable
ClearInput	1.16	Open & Claim
ClearInputProperties	1.16	*適用外*
ClearOutput	1.16	*適用外*
CompareFirmwareVersion	1.16	Open, Claim, & Enable
DirectIO	1.16	Open
ResetStatistics	1.16	Open, Claim, & Enable
RetrieveStatistics	1.16	Open, Claim, & Enable
UpdateFirmware	1.16	Open, Claim, & Enable
UpdateStatistics	1.16	Open, Claim, & Enable
専用		
StartVideo	1.16	Open, Claim, & Enable
StopVideo	1.16	Open, Claim, & Enable
TakePhoto	1.16	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.16	*適用外*
DirectIOEvent	1.16	Open, Claim

ErrorEvent	1.16	Open, Claim, & Enable
OutputCompleteEvent	1.16	*適用外*
StatusUpdateEvent	1.16	Open, Claim, & Enable
TransitionEvent	1.16	*適用外*

概説

ビデオキャプチャコントロールのOLEプログラマティクIDは"OPOS.VideoCapture"です。

機能

ビデオキャプチャントロールは次の機能を持っています。

写真を撮影し、それをファイルとしてホストに記録し、対象のストレージデバイスに保存することができます。

動画を撮影し、それをファイルとしてホストに記録し、対象のストレージデバイスに保存することができます。

ハイドラ接続されたスキャナーデバイスを使用して、バーコードラベルからエンコードされたデータを読み取ることができます。

ハイドラ接続された個人認識デバイスを使用して、個人の顔や物体を検出することができます。

モデル

ビデオキャプチャデバイスには2つの操作モードがあります。

・写真モード
・動画モード

各モードの動作は以下のとおりです。

・写真モード

CapPhotoプロパティがTRUEの場合、写真モードは写真画像をキャプチャし、それを画像データファイル形式としてホストに保存することができます。その対応可能なデータファイル形式は**PhotoType**プロパティに示され、対応可能なすべての値は**PhotoTypeList**プロパティに一覧で示されます。また、**CapStorage**の値がVCAP_CST_HARDTOTALS_ONLYまたはVCAP_CST_ALLの場合、デバイスは**Storage**プロパティで指定されたターゲットストレージデバイスにファイルを保存できます。

・動画モード

CapVideoプロパティがTRUEの場合、動画モードは動画データをキャプチャし、それを動画データファイル形式としてホストに保存することができます。その対応可能なデータファイル形式は**VideoType**プロパティに示され、すべての対応可能な値は**VideoTypeList**プロパティに一覧で示されます。また、**CapStorage**の値がVCAP_CST_HARDTOTALS_ONLYまたはVCAP_CST_ALLの場合、デバイスは**Storage**プロパティで指定されたターゲットストレージデバイスにファイルを保存できます。

デバイスの動作

ビデオキャプチャデバイスのコントロールオブジェクトは、次のようにデバイスの動作に従います。

以下に説明するように、これらはモードごとに異なります。

写真モード

CapPhotoプロパティがTRUEの場合、このモードを実行できます。このモードを開始する前に、ビデオキャプチャデバイスのコントロールオブジェクトは、**VideoCaptureMode**プロパティをVCAP_VCMODE_PHOTOに設定する必要があります。また、**CapPhotoColorSpace**プロパティ、**CapPhotoFrameRate**プロパティ、**CapPhotoResolution**プロパティ、**CapPhotoType**プロパティはそれぞれTRUEであり、これらの**PhotoColorSpaceList**プロパティ、**PhotoMaxFrameRate**プロパティ、**PhotoResolutionList**プロパティ、および**PhotoTypeList**プロパティには、このターゲットデバイスの写真ファイルデータとして使用される適切な値が必要です。次に、**PhotoColorSpace**プロパティ、**PhotoFrameRate**プロパティ、**PhotoResolution**プロパティ、**PhotoType**プロパティのそれぞれに適切な値を設定する必要があります。

TakePhotoメソッドを実行して写真の撮影を開始します。次に、ビデオキャプチャデバイスのコントロールオブジェクトは、写真画像をキャプチャし、**PhotoTypeList**プロパティにリストされている**PhotoType**プロパティの値で指定された画像データファイル形式としてホストに保存することができます。また、**CapStorage**の値がVCAP_CST_HARDTOTALS_ONLYまたはVCAP_CST_ALLの場合、**Storage**プロパティで指定されたストレージデバイスに保存できます。次に、ファイル名が**TakePhoto**メソッドパラメータによって設定され、写真データファイルをアプリケーションに引き渡すことができます。デバイスが画像データファイルを関連付けられたハードトータルデバイスへ書き込み可能にする必要がある場合、**CapAssociatedHardTotalsDevice**プロパティは、関連付けられたハードトータルデバイスのオープン名を保持します。

このメソッドは、写真を撮るプロセスとして同期して実行されます。記録されたデータの保存プロセスは非同期で実行されます。**StatusUpdateEvent**は、デバイスの状態の開始と終了が変更されたときにアプリケーションに対して呼び出されます。一度に実行できる**TakePhoto**メソッドの呼び出しは1つだけです。このメソッドの実行中、データの記録は非同期で処理され、デバイスの状態が変更されたときに、ビデオキャプチャデバイスの操作をネストしようとすると、保存が完了する前にOPOS_E_BUSYを返します。

指定されたパラメータのタイムアウトを超えた場合、写真ファイルの生成が終了した場合、または**ClearInput**メソッドが実行された場合、写真の撮影プロセスは終了します。

記録状態が開始されたことをアプリケーションに通知するために**TakePhoto**メソッドが実行されると、ステータスVCAP_SUE_START_PHOTOの**StatusUpdateEvent**が呼び出されます。

写真の撮影が終了するか、指定されたタイムアウトを超えると、ステータスVCAP_SUE_END_PHOTOの**StatusUpdateEvent**が呼び出され、写真の撮影が終了したことがアプリケーションに通知されます。

入力の収集または処理中にエラーが発生した場合、**ErrorEvent**イベント（または複数のイベント）がキューに入れられます。

ErrorEvent応答がOPOS_ER_RETRYの場合、記録データを保存するプロセスが再試行されます。ただし、エラーの原因が解決されない限り、**ErrorEvent**はすぐに再度発生します。

ErrorEventがOPOS_ER_CLEARの場合、デバイスにバッファリングされたすべてのデータがクリアされ、エラー状態が終了して、**TakePhoto**メソッドが破棄されます。

キューに入れられたすべての入力は、**ClearInput**メソッドを呼び出すことで削除できます。詳細については、**ClearInput**メソッドの説明を参照してください。

動画モード

このモードを開始する前に、ビデオキャプチャデバイスのコントロールオブジェクトは、**VideoCaptureMode**プロパティをVCAP_VCMODE_VIDEOに設定する必要があります。また、**CapVideoColorSpace**プロパティ、**CapVideoFrameRate**プロパティ、**CapVideoResolution**プロパティ、および**CapVideoType**プロパティはそれぞれTRUEであり、これらの**VideoColorSpaceList**プロパティ、**VideoMaxFrameRate**プロパティ、**VideoResolutionList**プロパティ、および**VideoTypeList**プロパティには、このターゲットデバイスで動画データファイルとして使用される適切な値が必要です。次に、**VideoColorSpace**プロパティ、**VideoFrameRate**プロパティ、**VideoResolution**プロパティ、**VideoType**プロパティのそれぞれに適切な値を設定する必要があります。

StartVideoメソッドを実行して動画の撮影を開始します。このメソッドは同期的に実行されます。動画の撮影中、このメソッドの実行、記録されたデータの保存が処理されます。記録は非同期で処理され、開始デバイスと終了デバイスの状態が変更されると、アプリケーションに対して**StatusUpdateEvent**が呼び出されます。さらに、デバイスの残りの記録時間は、**RemainingRecordingTimeInSec**プロパティで更新されます。次に、ビデオキャプチャデバイスのコントロールオブジェクトが動画をキャプチャし、**VideoTypeList**プロパティにリストされている**VideoType**プロパティのファイル名で指定された値を使用してホストに保存します。また、**CapStorage**の値がVCAP_CST_HARDTOTALS_ONLYまたはVCAP_CST_ALLの場合、**Storage**プロパティで指定されたストレージデバイスに保存できます。また、ファイル名は**StartVideo**メソッドパラメータで設定され、ビデオ画像データファイルをアプリケーションに引き渡すことができます。このメソッドは同期的に実行されます。

ビデオキャプチャは、指定された時間が経過した後、**StopVideo**メソッドが呼び出されたとき、または**ClearInput**メソッドが呼び出されたとき、**StartVideo**メソッドが呼び出された場合に終了します。

秒単位の残りのビデオキャプチャ記録時間は、**RemainingRecordingTimeInSec**プロパティから取得できます。

StartVideoメソッドが実行されると、ステータスVCAP_SUE_START_VIDEOの**StatusUpdateEvent**が呼び出され、動画撮影が開始されたことをアプリケーションに通知します。

動画撮影が終了するか、指定されたタイムアウトを超えると、ステータスVCAP_SUE_END_VIDEOの**StatusUpdateEvent**が呼び出され、動画撮影が終了したことがアプリケーションに通知されます。

StartVideoメソッドで指定された時間がFOREVER(-1)の場合、**StopVideo**メソッドが呼び出されるまで実行が続行されます。**StopVideo**メソッドが呼び出されると、**CapStorage**プロパティの値がVCAP_CST_HARDTOTALS_ONLYまたはVCAP_CST_ALLの場合、以前に取得した動画データがホストに記録され、**Storage**プロパティで指定されたターゲットストレージデバイスに引き渡すことができます。また、**StartVideo**メソッドで設定された指定ファイル名でアプリケーションに引き渡すことができます。

StartVideoメソッドの呼び出しは一度に1つだけ進行できます。ビデオ操作をネストしようとすると、OPOS_E_BUSYを返します。

StartVideoメソッドの実行中にエラーが発生した場合、アプリケーションは**StopVideo**メソッドを呼び出してビデオ撮影プロセスを終了するか、**ErrorEvent**処理を終了する前に**ClearInput**メソッドを呼び出してビデオ撮影プロセスをキャンセルすることができます。この後、**StopVideo**メソッドが呼び出されると、**CapStorage**プロパティ値がVCAP_CST_HARDTOTALS_ONLYまたはVCAP_CST_ALLの場合、**ErrorEvent**が発生する直前までのビデオファイルデータが、**Storage**プロパティで指定されたホストおよびターゲットストレージデバイスに保存され、アプリケーションに引き渡すことができます。

ErrorEvent応答がOPOS_ER_RETRYの場合、記録データを保存するプロセスが再試行されます。ただし、エラーの原因が解決されない限り、**ErrorEvent**はすぐに再度発生します。

ErrorEventがOPOS_ER_CLEARの場合、デバイスにバッファリングされたすべてのデータがクリアされ、エラー状態が終了して、ビデオキャプチャプロセスメソッドが破棄されます。

データの収集または処理中にエラーが発生した場合、**ErrorEvent**イベント（または複数のイベント）がキューに入れられます。

StartVideoメソッドの実行中にエラーが発生しなかった場合、ビデオの撮影プロセスを終了し、いつでもビデオの撮影を停止することができます。 **StopVideo**メソッドが呼び出されると、**CapStorage**プロパティがVCAP_CST_HARDTOTAL_ONLYまたはVCAP_CST_ALLの場合、メソッドが呼び出される直前までのビデオデータは、**Storage**プロパティで指定されたホストおよびターゲットストレージデバイスに記録され、アプリケーションに引き渡すことができます。

キューに入れられたすべてのデータは、**ClearInput**メソッドを呼び出すことで削除できます。詳細については、**ClearInput**メソッドの説明を参照してください。

デバイスの共有

ビデオキャプチャは下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、デバイスの読み込みを始める前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

アプリケーションは、デバイスを操作するメソッドを呼び出す前に、そのデバイスの排他アクセス権を獲得し、イネーブルにしなければなりません。

使用上の詳細な必要条件は、「一覧」を参照してください。

プロパティ

AutoExposure プロパティ R/W

形式　BOOL AutoExposure;

説明　TRUE：カメラのオート露出が有効です。
FALSE：カメラのオート露出が無効です。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。または、この機能をサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照　**CapAutoExposure**プロパティ

AutoFocus プロパティ R/W

形式　BOOL AutoFocus;

説明　TRUE：カメラのオートフォーカスが有効です。
FALSE：カメラのオートフォーカスが無効です。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。または、この機能をサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照　**CapAutoFocus**プロパティ

AutoGain プロパティ R/W

形式 BOOL AutoGain;

説明 TRUE：カメラのオートゲインが有効です。
FALSE：カメラのオートゲインが無効です。

AutoGainプロパティがTRUEの場合、**Gain**プロパティの値を読み取ることができます。ただし、**Gain**プロパティの値を書き込んだり変更したりすることはできません。**AutoGain**プロパティがFALSEの場合、**Gain**プロパティの値の読み取り、書き込み、および変更が可能です。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。または、この機能をサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照 **CapAutoGain**プロパティ、**Gain**プロパティ

AutoWhiteBalance プロパティ R/W

形式 BOOL AutoWhiteBalance;

説明 TRUE：カメラのオートホワイトバランスが有効です。
FALSE：カメラのオートホワイトバランスが無効です。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。または、この機能をサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照 **CapAutoWhiteBalance**プロパティ

Brightness プロパティ R/W

形式　LONG Brightness;

説明　カメラの明るさを示します。有効な値は、0から100の範囲です。

このプロパティは、Openメソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。または、この機能をサポートしていません。
その他	ResultCodeの項目を参照してください。

参照　CapBrightnessプロパティ

CapAssociatedHardTotalsDevice プロパティ

形式　BSTR CapAssociatedHardTotalsDevice;

説明　CapStorageプロパティがVCAP_CST_ALLまたはVCAP_CST_HARDTOTALS_ONLYの場合、デバイスが記録されたデータを関連付けられたハードトータルデバイスに保存でき、そのオープン名を保持できることを示します。

CapStorageプロパティがVCAP_CST_HOST_ONLYの場合、デバイスはデータを関連付けられたハードトータルデバイスに保存できず、このプロパティは、空の文字列である必要があります。

このプロパティは、Openメソッドにより初期化されます。

参照　CapStorageプロパティ

CapAutoExposure プロパティ

形式　BOOL CapAutoExposure;

説明　TRUE：カメラのオート露出変更が可能です。
FALSE：カメラのオート露出変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　AutoExposureプロパティ

CapAutoFocus プロパティ

形式　BOOL CapAutoFocus;

説明　TRUE：カメラのオートフォーカス変更が可能です。
FALSE：カメラのオートフォーカス変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　AutoFocusプロパティ

CapAutoGain プロパティ

形式　BOOL CapAutoGain;

説明　TRUE：カメラのオートゲイン変更が可能です。
FALSE：カメラのオートゲイン変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　AutoGainプロパティ

CapAutoWhiteBalance プロパティ

形式　BOOL CapAutoWhiteBalance;

説明　TRUE：カメラのオートホワイトバランス変更が可能です。
FALSE：カメラのオートホワイトバランス変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　AutoWhiteBalanceプロパティ

CapBrightness プロパティ

形式　BOOL CapBrightness;

説明　TRUE：カメラの明るさ変更が可能です。
FALSE：カメラの明るさ変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　Brightnessプロパティ

CapContrast プロパティ

形式　BOOL CapContrast;

説明　TRUE：カメラのコントラスト変更が可能です。
FALSE：カメラのコントラスト変更ができません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**Contrast**プロパティ

CapExposure プロパティ

形式　BOOL CapExposure;

説明　TRUE：カメラの露出変更が可能です。
FALSE：カメラの露出変更ができません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**Exposure**プロパティ

CapGain プロパティ

形式　BOOL CapGain;

説明　TRUE：カメラのゲイン変更が可能です。
FALSE：カメラのゲイン変更ができません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**Gain**プロパティ

CapHorizontalFlip プロパティ

形式　BOOL CapHorizontalFlip;

説明　TRUE：カメラの左右反転変更が可能です。
FALSE：カメラの左右反転変更ができません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**HorizontalFlip**プロパティ

CapHue プロパティ

形式　BOOL CapHue;

説明　TRUE：カメラの色相変更が可能です。
FALSE：カメラの色相変更ができません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　Hueプロパティ

CapPhoto プロパティ

形式　BOOL CapPhoto;

説明　TRUE：写真撮影機能をサポートしています。
FALSE：写真撮影機能をサポートしていません。

このプロパティがTRUEならば、写真機能をサポートし、写真を撮ることができます。また、写真モードをアクティブにするには、VideoCaptureModeプロパティでVCAP_VCMODE_PHOTOを設定する必要があります。FALSEならば、写真機能をサポートしていません。

このプロパティは、Openメソッドにより初期化されます。

参照　TakePhotoメソッド、VideoCaptureModeプロパティ

CapPhotoColorSpace プロパティ

形式　BOOL CapPhotoColorSpace;

説明　TRUE：写真色空間の変更ができます。
FALSE：写真色空間の変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　PhotoColorSpaceプロパティ

CapPhotoFrameRate プロパティ

形式　BOOL CapPhotoFrameRate;

説明　TRUE：写真フレームレートの変更ができます。
FALSE：写真フレームレートの変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　PhotoFrameRateプロパティ

CapPhotoResolution プロパティ

形式　BOOL CapPhotoResolution;

説明　TRUE：写真解像度の変更ができます。
FALSE：写真解像度の変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　PhotoResolutionプロパティ

CapPhotoType プロパティ

形式　BOOL CapPhotoType;

説明　TRUE：写真形式の変更ができます。
FALSE：写真形式の変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　PhotoTypeプロパティ

CapSaturation プロパティ

形式　BOOL CapSaturation;

説明　TRUE：カメラの彩度変更が可能です。
FALSE：カメラの彩度変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　Saturationプロパティ

CapStorage プロパティ

形式　LONG CapStorage;

説明　記録された動画または写真データファイルを書き込むことができる場所を示します。
このプロパティは、次のいずれかの値が設定されます。

値	説明
VCAP_CST_HARDTOTALS_ONLY	関連付けられたハードトータルデバイスのみサポートします。
VCAP_CST_HOST_ONLY	ホストのファイルシステムのみサポートします。
VCAP_CST_ALL	関連付けられたハードトータルデバイスとホストのファイルシステムの両方をサポートします。

このプロパティは、**Open**メソッドにより初期化されます。

ハードトータルデバイスがストレージをサポートしている場合、このプロパティは、VCAP_CST_HARDTOTALS_ONLYまたはVCAP_CST_ALLである必要があり、**CapAssociatedHardTotalsDevice**プロパティは、関連付けられたハードトータルデバイスのオープン名を保持します。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**Storage**プロパティ、**CapAssociatedHardTotalsDevice**プロパティ

CapVerticalFlip プロパティ

形式　**BOOL CapVerticalFlip;**

説明　TRUE：カメラの上下反転変更が可能です。
FALSE：カメラの上下反転変更ができません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**VerticalFlip**プロパティ

CapVideo プロパティ

形式　**BOOL CapVideo;**

説明　TRUE：動画撮影機能をサポートしています。
FALSE：動画撮影機能をサポートしていません。

このプロパティがTRUEならば、**StartVideo**メソッドを呼び出すことで動画撮影を行うことができます。また、写真モードをアクティブにするには、**VideoCaptureMode**プロパティでVCAP_VCMODE_VIDEOを設定する必要があります。FALSEならば、動画撮影を行うことができません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**StartVideo**メソッド、**VideoCaptureMode**プロパティ

CapVideoColorSpace プロパティ

形式　**BOOL CapVideoColorSpace;**

説明　TRUE：動画色空間の変更ができます。
FALSE：動画色空間の変更ができません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　VideoColorSpaceプロパティ

CapVideoFrameRate プロパティ

形式　BOOL CapVideoFrameRate;

説明　TRUE：動画フレームレートの変更ができます。
FALSE：動画フレームレートの変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　VideoMaxFrameRateプロパティ、VideoFrameRateプロパティ

CapVideoResolution プロパティ

形式　BOOL CapVideoResolution;

説明　TRUE：動画解像度の変更ができます。
FALSE：動画解像度の変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　VideoResolutionListプロパティ、VideoResolutionプロパティ

CapVideoType プロパティ

形式　BOOL CapVideoType;

説明　TRUE：動画形式の変更ができます。
FALSE：動画形式の変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　VideoTypeListプロパティ、VideoTypeプロパティ

Contrast プロパティ R/W

形式　LONG Contrast;

説明　カメラのコントラストを示します。有効な値は、0から100の範囲です。

このプロパティは、Openメソッドにより初期化されます。

戻り値

値	説明

OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。または、この機能をサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照 **CapContrast**プロパティ

Exposure プロパティ R/W

形式 **LONG Exposure;**

説明 カメラの露出を示します。有効な値は、0から100の範囲です。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。または、この機能をサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照 **CapExposure**プロパティ

Gain プロパティ R/W

形式 **LONG Gain;**

説明 カメラのゲインを示します。有効な値は、0から100の範囲です。

AutoGainプロパティがTRUEの場合、**Gain**プロパティの値を読み取ることができます。ただし、**Gain**プロパティの値を書き込んだり変更したりすることはできません。**AutoGain**プロパティがFALSEの場合、**Gain**プロパティの値の読み取り、書き込み、および変更が可能です。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明

OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。または、この機能をサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照　**CapGain**プロパティ、**AutoGain**プロパティ

HorizontalFlip プロパティ R/W

形式　**BOOL HorizontalFlip;**

説明　TRUE：カメラの左右反転が有効です。
FALSE：カメラの左右反転が無効です。

VerticalFlipプロパティと呼ばれる同様のプロパティがあります。ただし、**VerticalFlip**プロパティと**HorizontalFlip**プロパティの値は個別に設定できます。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。または、この機能をサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照　**CapHorizontalFlip**プロパティ、**VerticalFlip**プロパティ、**CapVerticalFlip**プロパティ

Hue プロパティ R/W

形式　**LONG Hue;**

説明　カメラの色相を示します。有効な値は、0から100の範囲です。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明

OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。または、この機能をサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照 **CapHue**プロパティ

PhotoColorSpace プロパティ R/W

形式 **BSTR PhotoColorSpace;**

説明 **CapPhotoColorSpace**プロパティがTRUEであり、**TakePhoto**メソッドが使用されている場合に、ビデオキャプチャデバイスで取得するフレームデータの色空間IDを示します。有効な値は、**CapPhotoColorSpaceList**プロパティに記された内の一つです。このプロパティは、**VideoCaptureMode**プロパティ値がVCAP_VCMODE_PHOTOであり、**CapPhoto**プロパティがTRUEの場合に参照されます。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照 **PhotoColorSpaceList**プロパティ、**VideoCaptureMode**プロパティ、**CapPhoto**プロパティ、**CapPhotoColorSpace**プロパティ、**TakePhoto**メソッド

PhotoColorSpaceList プロパティ

形式 **BSTR PhotoColorSpaceList;**

説明 デバイスがサポートしている色空間情報をカンマ区切りのリストで示します。

各色空間情報は以下の情報で構成されており、下記順番にコロン（“:”）で区切って示します。

パラメータ	説明
色空間ID	RGB, YUV422等の色空間を識別するID。RGBの深度が16ビット、YUV422の深度が32ビットの場合、"RGB:16;YUV422:32;…" の様に示されます。
深度	1ピクセルあたりのbit数

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CapPhotoColorSpace**プロパティ、**PhotoColorSpace**プロパティ、**VideoCaptureMode**プロパティ

PhotoFrameRate プロパティ R/W

形式　**LONG PhotoFrameRate;**

説明　ビデオキャプチャデバイスで記録されるフレームデータ、及び**TakePhoto**メソッドによって撮影した写真のフレームレートを示します。このプロパティは、**VideoCaptureMode**プロパティにVCAP_VCMODE_PHOTOが設定されていて、**CapPhoto**プロパティがTRUEの場合にのみ適用されます。有効な値は、1から**PhotoMaxFrameRate**プロパティの値の範囲です。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**CapPhoto**プロパティ、**CapPhotoFrameRate**プロパティ、**PhotoMaxFrameRate**プロパティ、**VideoCaptureMode**プロパティ、**TakePhoto**メソッド

PhotoMaxFrameRate プロパティ

形式　**LONG PhotoMaxFrameRate;**

説明　**PhotoFrameRate**プロパティに設定可能なフレームレートの最大値を示します。

このプロパティは、**Open**メソッドにより初期化されます。

参照 **PhotoFrameRate**プロパティ、**VideoCaptureMode**プロパティ

PhotoResolution プロパティ R/W

形式 BSTR PhotoResolution;

説明 ビデオキャプチャデバイスで記録されるフレームデータ、及び**TakePhoto**メソッドで撮影した写真の解像度を示します。有効な値は、**PhotoResolutionList**プロパティに記された内の一つです。このプロパティは、**VideoCaptureMode**プロパティにVCAP_VCMODE_PHOTOが設定されていて、**CapPhoto**プロパティがTRUEの場合にのみ適用されます。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照 **CapPhoto**プロパティ、**PhotoResolutionList**プロパティ、**VideoCaptureMode**プロパティ、**TakePhoto**メソッド

PhotoResolutionList プロパティ

形式 BSTR PhotoResolutionList;

説明 **PhotoResolution**プロパティに設定可能な解像度をカンマ区切りのリストで示します。
解像度は"横x縦"の形式で示されます。
例えば、320x240, 640x480, 640x360をサポートしている時は以下の様になります。
"320x240,640x480,640x360"

このプロパティは、**Open**メソッドにより初期化されます。

参照 **CapPhotoResolution**プロパティ、**PhotoResolution**プロパティ、**VideoCaptureMode**プロパティ

PhotoType プロパティ R/W

形式　　BSTR PhotoType;

説明　　TakePhotoメソッドで撮影する写真の画像形式を示します。有効な値は、PhotoTypeListプロパティに記された内の一つです。このプロパティは、VideoCaptureModeプロパティにVCAP_VCMODE_PHOTOが設定されていて、CapPhotoプロパティがTRUEの場合にのみ適用されます。

このプロパティは、Openメソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	ResultCodeの項目を参照してください。

参照　　CapPhotoプロパティ、PhotoTypeListプロパティ、TakePhotoメソッド、VideoCaptureModeプロパティ

PhotoTypeList プロパティ

形式　　BSTR PhotoTypeList;

説明　　PhotoTypeプロパティに設定可能な写真フォーマット値をカンマ区切りのリストで示します。
例えば、BMP, JPEGをサポートしている時は以下の様になります。
"BMP,JPEG"
※表記内容はデバイスにより異なる可能性があります。

このプロパティは、Openメソッドにより初期化されます。

参照　　PhotoTypeプロパティ、VideoCaptureModeプロパティ

RemainingRecordingTimeInSec プロパティ

形式 LONG RemainingRecordingTimeInSec;

説明 このプロパティは、動画の記録が進行中の場合、残りの録画時間を秒単位で保持します。進行中のビデオ録画がない場合、その値は0です。

StartVideoメソッドの呼び出しから戻るとき、このプロパティは、その呼び出しで渡された引数*RecordingTime*の時間に初期化されます。この引数値がOPOS_FOREVER (-1)の場合、このプロパティは、StopVideoメソッドが呼び出されるまでこの値を変更せずに保持します。

このプロパティは、DeviceEnabledプロパティを設定するときに0に初期化されます。

参照 StartVideoメソッド、StopVideoメソッド

Saturation プロパティ R/W

形式 LONG Saturation;

説明 カメラの彩度を示します。有効な値は、0から100の範囲です。

このプロパティは、Openメソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。または、この機能をサポートしていません。
その他	ResultCodeの項目を参照してください。

参照 CapSaturationプロパティ

Storage プロパティ R/W

形式 LONG Storage;

説明 動画または写真が記録されたデータファイルを書き込む場所を示します。
このプロパティは、StartVideoメソッドまたはTakePhotoメソッドを呼び出す前に設定する必要があります。
このプロパティは、次のいずれかの値が設定されます。

値	説明
VCAP_ST_HARDTOTALS	動画または写真のデータファイルは、関連付けられたハードトータルデバイスに書き込まれます。**CapAssociatedHardTotalsDevice**プロパティは、関連付けられたハードトータルデバイスのオープン名を保持します。
VCAP_ST_HOST	動画または写真のデータファイルは、ホストのファイルシステムに書き込まれます。
VCAP_ST_HOST_HARDTOTALS	動画または写真のデータファイルは、関連付けられたハードトータルデバイスとホストのファイルシステムに書き込まれます。**CapAssociatedHardTotalsDevice**プロパティは、関連付けられたハードトータルデバイスのオープン名を保持します。

このプロパティは、**CapStorage**プロパティが保持する値に従って**Open**メソッドにより初期化されます。

CapStorageプロパティの値がVCAP_CST_ALLの場合、VCAP_ST_HOST_HARDTOTALSに初期化されます。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。または、記録が進行中です。
その他	**ResultCode**の項目を参照してください。

参照　**CapStorage**プロパティ

VerticalFlip プロパティ R/W

形式　BOOL VerticalFlip;

説明　TRUE：カメラの上下反転が有効です。
FALSE：カメラの上下反転が無効です。

HorizontalFlipプロパティと呼ばれる同様のプロパティがあり、**VerticalFlip**プロパティと**HorizontalFlip**プロパティの値は個別に設定できます。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。または、この機能をサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照　**CapVerticalFlip**プロパティ、**HorizontalFlip**プロパティ、**CapHorizontalFlip**プロパティ

VideoCaptureMode プロパティ R/W

形式　LONG VideoCaptureMode;

説明　ビデオキャプチャの動作モードを示します。

有効な値は以下の通りです。

値	説明

VCAP_VCMODE_PHOTO
写真撮影と記録を行うモードです。
CapPhotoプロパティがTRUEの場合に設定できます。
PhotoTypeプロパティ、**PhotoColorSpace**プロパティ、**PhotoResolution**プロパティ、**PhotoFrameRate**プロパティの値は、**PhotoTypeList**プロパティの写真の画像形式リスト、**PhotoColorSpaceList**プロパティの色空間値リスト、**PhotoResolutionList**プロパティの解像度値リスト、**PhotoMaxFrameRate**プロパティの値の範囲のフレームレート値が適用されます。
写真撮影は**TakePhoto**メソッドによって実行されます。

VCAP_VCMODE_VIDEO
動画撮影と記録を行うモードです。
CapVideoプロパティがTRUEの場合に設定できます。
VideoTypeプロパティ、**VideoColorSpace**プロパティ、**VideoResolution**プロパティ、**VideoFrameRate**プロパティの値は、**VideoTypeList**プロパティの写真の画像形式リスト、**VideoColorSpaceList**プロパティの色空間値リスト、**VideoResolutionList**プロパティの解像度値リスト、**VideoMaxFrameRate**プロパティの値の範囲のフレームレート値が適用されます。
動画の撮影と記録は**StartVideo**メソッドによって実行され、**StopVideo**メソッドを使用して終了します。

このプロパティは、**Open**メソッドにより初期化されます。

このプロパティのデフォルト値はVCAP_VCMODE_PHOTOです。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。または、この機能をサポートしていません。

その他	**ResultCode**の項目を参照してください。

参照 **PhotoColorSpace**プロパティ、**VideoColorSpace**プロパティ、**PhotoResolution**プロパティ、**VideoResolution**プロパティ、**VideoFrameRate**プロパティ、**PhotoFrameRate**プロパティ、**CapPhotoColorSpace**プロパティ、**CapVideoColorSpace**プロパティ、**CapPhotoResolution**プロパティ、**CapVidoeResolution**プロパティ、**VideoMaxFrameRate**プロパティ、**PhotoMaxFrameRate**プロパティ、**CapPhoto**プロパティ、**CapVideo**プロパティ、**VideoType**プロパティ、**VideoTypeList**プロパティ、**PhotoType**プロパティ、**PhotoTypeList**プロパティ、**TakePhoto**メソッド、**StartVideo**メソッド、**StopPhoto**メソッド

VideoColorSpace プロパティ R/W

形式 BSTR VideoColorSpace;

説明 **CapVideoColorSpace**プロパティがTRUEであり、**StartVideo**メソッドによって使用される場合、ビデオキャプチャデバイスによって取得されるフレームデータの色空間IDを示します。有効な値は、**VideoColorSpaceList**プロパティに記された内の一つです。

このプロパティは、**VideoCaptureMode**プロパティにVCAP_VCMODE_VIDEOが設定され、**CapVideo**プロパティがTRUEの場合に適用されます。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照 **CapVideoColorSpace**プロパティ、**VideoColorSpaceList**プロパティ、**VideoCaptureMode**プロパティ、**StartVideo**メソッド

VideoColorSpaceList プロパティ

形式 BSTR VideoColorSpaceList;

説明　デバイスがサポートしている色空間情報をカンマ区切りのリストで示します。

各色空間情報は以下の情報で構成されており、下記順番にコロン（“:”）で区切って示します。

パラメータ	説明
色空間ID	RGB, YUV422等の色空間を識別するID。RGBの深度が16ビット、YUV422の深度が32ビットの場合、"RGB:16;YUV422:32;…" の様に示されます。
深度	1ピクセルあたりのbit数

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CapVideoColorSpace**プロパティ、**VideoCaptureMode**プロパティ、**VideoColorSpace**プロパティ

VideoFrameRate プロパティ R/W

形式　**LONG VideoFrameRate;**

説明　ビデオキャプチャデバイスで取得するフレームデータ、及び**StartVideo**メソッドで撮影した動画のフレームレートを示します。有効な値は、1から**VideoMaxFrameRate**プロパティの値の範囲です。

このプロパティは、**VideoCaptureMode**プロパティにVCAP_VCMODE_VIDEOが設定され、**CapVideo**プロパティがTRUEの場合に適用されます。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**CapVideo**プロパティ、**CapVideoFrameRate**プロパティ、**VideoCaptureMode**プロパティ、**VideoMaxFrameRate**プロパティ、**StartVideo**メソッド

VideoMaxFrameRate プロパティ

形式　LONG VideoMaxFrameRate;

説明　VideoFrameRateプロパティに設定可能なフレームレートの最大値を示します。

このプロパティは、Openメソッドにより初期化されます。

参照　VideoFrameRateプロパティ、VideoCaptureModeプロパティ

VideoResolution プロパティ R/W

形式　BSTR VideoResolution;

説明　ビデオキャプチャデバイスで取得するフレームデータ、及びStartVideoメソッドで撮影した動画の解像度を示します。有効な値は、VideoResolutionListプロパティに記された内の一つです。

このプロパティは、VideoCaptureModeプロパティにVCAP_VCMODE_VIDEOが設定され、CapVideoプロパティがTRUEの場合に適用されます。

このプロパティは、Openメソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	ResultCodeの項目を参照してください。

参照　VideoResolutionListプロパティ、CapVideoプロパティ、VideoCaptureModeプロパティ、StartVideoメソッド

VideoResolutionList プロパティ

形式　BSTR VideoResolutionList;

説明　VideoResolutionプロパティに設定可能な解像度をカンマ区切りのリストで示します。
解像度は"横x縦"の形式で示されます。
例えば、320x240, 640x480, 640x360をサポートしている時は以下の様になります。
"320x240,640x480,640x360"

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CapVideoResolution**プロパティ、**VideoResolution**プロパティ

VideoType プロパティ R/W

形式　**BSTR VideoType;**

説明　**StartVideo**メソッドで撮影する動画の形式を示します。有効な値は、**VideoTypeList**プロパティに記された内の一つです。

このプロパティは、**VideoCaptureMode**プロパティにVCAP_VCMODE_VIDEOが設定され、**CapVideo**プロパティがTRUEの場合に適用されます。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**VideoCaptureMode**プロパティ、**CapVideo**メソッド、**VideoTypeList**プロパティ、**StartVideo**メソッド

VideoTypeList プロパティ

形式　**BSTR VideoTypeList;**

説明　**VideoType**プロパティに設定可能な動画フォーマットの値をカンマ区切りのリストで示します。
例えば、AVI_IYUV, AVI_MJPGをサポートしている時は以下の様になります。(注1)
"AVI_IYUV,AVI_MJPG"
※表記内容はデバイスにより異なる可能性があります。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**VideoType**プロパティ、**VideoCaptureMode**プロパティ

(注1) 動画形式の関連情報を参考として以下に記載します。

AVI：Digital container format：
https://en.wikipedia.org/wiki/Digital_container_format

MJPG：Motion JPEG：
https://en.wikipedia.org/wiki/Motion_JPEG

IYUV：4:2:0 Video Pixel Formats：
https://docs.microsoft.com/en-us/windows-hardware/drivers/display/4-2-0-video-pixel-formats

4:2:2 Video Pixel Formats：
https://docs.microsoft.com/en-us/windows-hardware/drivers/display/4-2-2-video-pixel-formats

Video Formats and their Abbreviation：
http://technewzbd.blogspot.com/2013/05/video-formats-and-their-abbreviation.html

注:ビデオキャプチャデバイスのプロパティ値の関係

以下にリストされているプロパティは、写真／動画モードグループ内で関連しており、値が変更されると、それに応じて他の値が変更される場合があります。

写真モードグループのプロパティ

PhotoType, **PhotoColorSpace**, **PhotoColorSpaceList**, **PhotoFrameRate**, **PhotoMaxFrameRate**, **PhotoResolution**, **PhotoResolutionList**

動画モードグループのプロパティ

VideoType, **VideoColorSpace**, **VideoColorSpaceList**, **VideoFrameRate**, **VideoMaxFrameRate**, **VideoResolution**, **VideoResolutionList**

メソッド

StartVideo メソッド

形式　LONG StartVideo (BSTR *FileName*, BOOL *OverWrite*, LONG *RecordingTime*);

パラメータ	説明
FileName	記録する動画ファイル名を指定します。
OverWrite	同名ファイル存在時の挙動を指定します。TRUEならば上書きし、FALSEならエラーを返します。
RecordingTime	録画を行う時間を秒単位で指定します。OPOS_FOREVER(-1)を指定すると、**StopVideo**メソッドを呼ぶまで録画を続けます。

説明　このメソッドを呼び出す前に、**VideoCaptureMode**プロパティにVCAP_VCMODE_VIDEOを設定して、動画撮影モードに変更する必要があります。

このメソッドは、**CapVideo**プロパティがTRUEの場合に実行できます。

VideoColorSpaceプロパティ、**VideoResolution**プロパティ、**VideoFrameRate**プロパティ、**VideoType**プロパティの設定内容で録画と記録を開始します。

このメソッドは同期的に実行します。

動画の撮影中、記録されたデータの保存は非同期で処理され、開始状態と停止状態が変更されると、アプリケーションに対して**StatusUpdateEvent**が呼び出されます。

*RecordingTime*に指定した時間が経過するか、**StopVideo**メソッドを呼ぶことで録画が完了し、*FileName*で指定した動画ファイルが記録され、アプリケーションに引き渡すことができるようになります。
また、録画記録中は**Status**プロパティにOPOS_S_BUSYが設定されます。

動画ファイルが記録される場所は、**Storage**プロパティによって制御されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	*FileName*が長すぎるか使用できない文字が含まれています、または*RecordhingTime*に0が指定されています。
OPOS_E_EXISTS	*FileName*がすでに存在します。(*OverWrite*がFALSEの時)
OPOS_E_BUSY	録画中なので実行できません。
その他	**ResultCode**の項目を参照してください。

参照 **VideoColorSpace**プロパティ、**VideoResolution**プロパティ、**VideoFrameRate**プロパティ、**VideoType**プロパティ、**StopVideo**メソッド、**StatusUpdateEvent**イベント、**VideoCaptureMode**プロパティ

StopVideo メソッド

形式 LONG StopVideo ();

説明 **StartVideo**メソッドで開始した録画処理を終了し、動画ファイルの記録を完了させます。

このメソッドは同期的に実行します。

デバイスは**StatusUpdateEvent**を呼び出して、動画の撮影と記録が停止したことをアプリケーションに通知します。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	録画されていません。
その他	**ResultCode**の項目を参照してください。

参照 **StartVideo**メソッド、**StatusUpdateEvent**イベント

TakePhoto メソッド

形式　LONG TakePhoto (BSTR *FileName*, BOOL *OverWrite*, LONG *Timeout*);

パラメータ	説明
FileName	記録する画像ファイル名を指定します。
OverWrite	同名ファイル存在時の挙動を指定します。TRUEならば上書きし、FALSEならエラーを返します。
Timeout	メソッドの失敗やタイムアウトの**ErrorEvent**をアプリケーションに通知する前に許容された実行時間をミリ秒であらわします。OPOS_FOREVER(-1)の場合、サービスは写真が撮影されるかアプリケーションエラーが発生するまで待ちます。

説明　**PhotoColorSpace**プロパティ、**PhotoResolution**プロパティ、**PhotoFrameRate**プロパティ、**PhotoType**プロパティの設定内容で写真を撮影し、画像を記録します。

このメソッドを呼び出す前に、**VideoCaptureMode**プロパティにVCAP_VCMODE_PHOTOを設定して、写真撮影モードに変更する必要があります。

このメソッドは、**CapPhoto**プロパティがTRUEの場合に実行できます。

このメソッドは同期的に実行します。

記録されたデータの保存プロセスは非同期で実行されます。開始状態と停止状態が変更されると、アプリケーションに対して**StatusUpdateEvent**が呼び出されます。

画像ファイルが記録される場所は、**Storage**プロパティによって制御されます。

タイムアウトはミリ秒数を指定します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。

OPOS_E_ILLEGAL	下記のいずれかが発生しました。 *FileName*が長すぎるか使用できない文字が含まれています。 **VideoCaptureMode**プロパティがVCAP_VCMODE_PHOTOではなく、**CapPhoto**プロパティがTRUEではありません。
OPOS_E_EXISTS	*FileName*がすでに存在します。 (*OverWrite*=FALSE時)
その他	**ResultCode**の項目を参照してください。

参照 **VideoCaptureMode**プロパティ、**PhotoColorSpace**プロパティ、**PhotoResolution**プロパティ、**CapPhoto**プロパティ、**PhotoType**プロパティ、**PhotoFrameRate**プロパティ、**StatusUpdateEvent**イベント

イベント

DirectIOEvent イベント

形式　void DirectIOEvent (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。イベント番号とサービスオブジェクトで変化する特定値です。
pString	文字列のポインタ。イベント番号とサービスオブジェクトで変化する特定値です。このデータのフォーマットはBinaryConversionプロパティの値に依存します。詳細はBinaryConversionプロパティを参照してください。

説明　アプリケーションに直接サービス情報を提供します。このイベントは、ベンダー固有のビデオキャプチャサービスが、デバイスコントロールでサポートされていないイベントをアプリケーションに提供する手段を提供します。

アプリケーションと直接通信するために、サービスオブジェクトが通知します。

このイベントにより、コントロールオブジェクトではサポートされないイベントを、サービスオブジェクトがアプリケーションに提供することが可能となります。

参照　DirectIOメソッド

ErrorEvent イベント

形式 **void ErrorEvent (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**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**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。

もし、*ResultCode*がOPOS_E_EXTENDEDであれば *ResultCodeExtended*は次に示す値のいずれかに設定されます。

値	説明
OPOS_EVCAP_NOROOM	ファイルを作成するのに十分なエリアがありません。

ErrorLocus パラメータは次の値です。

値	意味
OPOS_EL_INPUT	非同期記録データ保存の処理中にエラーが発生しました。

*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	記録したデータの保存を再試行します。エラー状態は解除されます。通常、*ErrorLocus*がOPOS_EL_INTPUTの場合、非同期記録データの保存に有効で、この場合、非同期記録データの保存が再試行され、エラー状態が解除されます。 これがデフォルトの応答です。

OPOS_ER_CLEAR	全てのバッファリングされている撮影または、保存データをクリアします。エラー状態は解除されます。

説明　ビデオキャプチャデバイスエラーが検出され、エラー状態を処理するためにアプリケーションによる適切な応答が必要であることをアプリケーションに通知します。

エラーが検出され、コントロールの**State**がエラー状態に遷移したときに通知します。

参照　**「ステータス、結果コード、状態モデル」**

StatusUpdateEvent イベント

形式 void StatusUpdateEvent (LONG *Status*);

パラメータ	説明
Status	ユニットの電源状態の変化または状態の変化を示します。

第1.3版以降では、電源状態通知機能が追加され、電源状態通知の**StatusUpdateEvent**値が追加されていることに注意してください。

ファームウェア更新の機能により、非同期更新ファームウェア更新処理の状態と進行状況を通知するためのステータス値が追加されました。共通プロパティ、メソッド、イベントの「**StatusUpdateEvent**」の説明を参照してください。

値	説明
VCAP_SUE_START_VIDEO	動画撮影が開始されると通知されます。
VCAP_SUE_STOP_VIDEO	動画撮影が停止すると通知されます。
VCAP_SUE_START_PHOTO	写真撮影が開始されると通知されます。
VCAP_SUE_END_PHOTO	写真撮影が終了すると通知されます。

説明 ビデオキャプチャデバイスの電源状態の変化または状態の変化があることをアプリケーションに通知します。

ビデオキャプチャデバイスが電源状態の変化または状態の変化を検出すると、キューに入れられます。

参照 「**ステータス、結果コード、状態モデル**」；「デバイス電源通知モデルについて」；CapPowerReportingプロパティ、PowerNotifyプロパティ

第 40 章
個体認識

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.16	Boolean	R/W	Open
BinaryConversion	1.16	Long	R/W	Open
CapCompareFirmwareVersion	1.16	Boolean	R	Open
CapPowerReporting	1.16	Long	R	Open
CapStatisticsReporting	1.16	Boolean	R	Open
CapUpdateFirmware	1.16	Boolean	R	Open
CapUpdateStatistics	1.16	Boolean	R	Open
CheckHealthText	1.16	String	R	Open
Claimed	1.16	Boolean	R	Open
DataCount	1.16	Long	R	Open
DataEventEnabled	1.16	Boolean	R/W	Open
DeviceEnabled	1.16	Boolean	R/W	Open & Claim
FreezeEvents	1.16	Boolean	R/W	Open
OpenResult	1.16	Long	R	無し
OutputID	1.16	Long	R	*適用外*
PowerNotify	1.16	Long	R/W	Open
PowerState	1.16	Long	R	Open
ResultCode	1.16	Long	R	無し
ResultCodeExtended	1.16	Long	R	Open
State	1.16	Long	R	無し
ControlObjectDescription	1.16	String	R	無し
ControlObjectVersion	1.16	Long	R	無し
ServiceObjectDescription	1.16	String	R	Open
ServiceObjectVersion	1.16	Long	R	Open
DeviceDescription	1.16	String	R	Open
DeviceName	1.16	String	R	Open

専用				
CapIndividualList	1.16	String	R	Open
IndividualIDs	1.16	String	R	Open
IndividualRecognitionFilter	1.16	String	R/W	Open & Claim & Enable
IndividualRecognitionInformation	1.16	String	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.16	無し
Close	1.16	Open
ClaimDevice	1.16	Open
ReleaseDevice	1.16	Open & Claim
CheckHealth	1.16	Open, Claim, & Enable
ClearInput	1.16	Open & Claim
ClearInputProperties	1.16	Open & Claim
ClearOutput	1.16	*適用外*
CompareFirmwareVersion	1.16	Open, Claim, & Enable
DirectIO	1.16	Open
ResetStatistics	1.16	Open, Claim, & Enable
RetrieveStatistics	1.16	Open, Claim, & Enable
UpdateFirmware	1.16	Open, Claim, & Enable
UpdateStatistics	1.16	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.16	Open, Claim, & Enable
DirectIOEvent	1.16	Open, Claim
ErrorEvent	1.16	Open, Claim, & Enable
OutputCompleteEvent	1.16	*適用外*
StatusUpdateEvent	1.16	Open, Claim, & Enable
TransitionEvent	1.16	*適用外*

概説

個体認識コントロールのOLEプログラマティクIDは"OPOS.IndividualRecognition"です。

機能

個体認識コントロールは次の機能を持っています。

カメラの映像を解析し、人やボール等の個体を認識します。

モデル

個体認識コントロールは、イベント駆動入力の一般的な入力モデルに従います。

AutoDisableプロパティがTRUEならば、DataEventのキューイング時にコントロールは自動的にディセーブルとなります。

DataEventEnabledプロパティがTRUEならば、キューイングしたDataEventはアプリケーションに通知されます。コントロールはこのイベントを発生させる直前に、データをプロパティにコピーし、さらなるデータイベントを発生させないために、DataEventEnabledプロパティをFALSEに設定します。これにより、アプリケーションが現在の入力の処理や関連するプロパティの処理をしている間、コントロールはその後の入力データをキューイングするようになります。アプリケーションが現在の入力データの処理を終え、次のデータ処理の準備ができたとき、DataEventEnabledプロパティをTRUEに設定することにより、再びデータイベントが通知されるようになります。

入力データの読込み中または処理中に、コントロールにおいてエラーが発生した場合、ErrorEventが発行され、DataEventEnabledプロパティがTRUEのならば、アプリケーションに通知されます。

DataCountプロパティを読むことでコントロールによってキューイングされたデータイベント数が得られます。

コントロールにキューイングされたすべての入力データはClearInputメソッドを呼ぶことにより削除できます。

DataEventもしくはErrorEvent発生により入力された全てのデータプロパティは、ClearInputPropertiesメソッドを呼ぶ事によりデフォルト値に戻す事が出来ます

認識可能な個体はCapIndividualListプロパティで示します。

デバイスでサポートしている機能の確認や有効・無効の設定等は**IndividualRecognitionInformation**プロパティで行います。

認識されたされたデータは、**IndividualRecognitionInformation**プロパティ、IndividualIDsに格納されます。

個体を認識する方法は、**IndividualRecognitionFilter**関数によって異なります。詳細は、**IndividualRecognitionFilter**セクションを参照してください。

IndividualRecognitionFilter

個体認識フィルタープロパティは、個体認識デバイスの個体認識機能の情報として、以下のデータを定義します。

・各種サポート機能の有無。
（サポートされる機能はデバイスによって定義されます）

・各種機能の状態を有効、無効にします。

・各種機能で扱うタイプ（例：性別認識の「男性」、「女性」）

・各種機能のフィルター設定。

次のデータは、**IndividualRecognitionInformation**プロパティで定義されています

・個体認識入力データ

デバイスは、個体認識機能情報と個体認識入力データを定義します。

アプリケーションはこれらのコンテンツを参照して、サポート範囲などを決定します。また、各種機能の有効/無効状態やフィルタ設定を変更し、各機能を制御します。

アプリケーションによって設定されたさまざまな機能の有効/無効状態、およびフィルター設定は、**DeviceEnabled**プロパティを**true**に設定し、個別の認識機能を有効にすることによって適用されます。

アプリケーションが各種機能を設定する場合、対象となる機能のみを指定・設定することができます。

デバイスは、アプリケーションによって設定されたコンテンツに基づいて**DataEvent**を起動し、入力データを**IndividualRecognitionInformation**プロパティに格納します。

デバイスの共有

個体認識は下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、デバイスの読み込みを始める前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の詳細な必要条件は、「一覧」を参照してください。

プロパティ

CapIndividualList プロパティ

形式　BSTR CapIndividualList;

説明　認識可能な個体情報をカンマ区切りのリストで示します。

各個体情報は以下の情報で構成されており、下記順番にコロン（“:”）で区切って示します。

パラメータ	説明

個体ID　識別可能な個体を表すID。

個体名　個体の名前。

このプロパティは、**Open**メソッドにより初期化されます。

参照　Individual**IDs**プロパティ

IndividualIDs プロパティ

形式　BSTR IndividualIDs;

説明　個体認識で認識した個体IDをコロン（“:”）で区切って示します。

本プロパティは、DataEventが通知される直前にコントロールによって設定されます。

参照　**CapIndividualList**プロパティ

IndividualRecognitionFilter プロパティ R/W

形式　BSTR IndividualRecognitionFilter;

説明　IndividualRecognitionFilterプロパティでは、以下のデータを示します。

個体認識機能情報

- 各種機能のサポート有無(サポートしている機能がデバイスにより定義される)
- 各種機能の有効・無効状態
- 各種機能が扱うタイプ(例：性別認識の「男性」「女性」等)

・各種機能のフィルタ設定

個体認識機能情報データはすべてデバイスが定義します。

アプリケーションはこれらの内容を参照し、サポート範囲等を判別できます。

また、アプリケーション各種機能の有効・無効状態、フィルタ設定を変更し、各機能を制御できます。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

IndividualRecognitionInformation プロパティ

形式　**BSTR IndividualRecognitionInformation;**

説明　IndividualRecognitionInformationプロパティでは、以下のデータを示します。

個体認識入力データ

個体認識入力データはすべてデバイスが定義します。

本プロパティは、**DataEvent**が通知される直前にコントロールによって設定されます。

IndividualRecognitionFilterフォーマット

個体認識のIndividualRecognitionFilterプロパティでは、JSON形式で各種情報を定義する。

■基本項目

キー	値	値変更可否	説明
IndividualRecognitionFilter	object	×	各種個体認識に関する情報 デバイスはここで定義する個体認識をサポートする。
[IndividualID]	object	×	認識可能な個体の情報 キー名は個体を表すID。
Enabled	boolean	○	対象個体認識の有効・無効状態 アプリケーションはこの値を変更して対象個体認識を制御できる。
Properties	object	×	対象個体認識で扱うプロパティ情報 アプリケーションはここで定義される値を参照、または変更して対象個体認識を制御する。
[Property01]	-	-	
[Property02]	-	-	
Filters	object	×	入力データフィルタ設定情報 アプリケーションはここで定義される値を変更して対象個体認識の入力データをフィルタリングする。
[Filter01]	-	-	
[Filter02]	-	-	

■例：Face(顔認識)機能を持つデバイスの場合

キー	値	値変更可否	説明
IndividualRecognitionFilter	object	×	
Face	object	×	
Enabled	boolean	○	
Properties	object	×	
FaceImageNamePrefix	string	○	顔認識時出力画像ファイルのプレフィックス
Gender	object	×	性別認識に関する情報
Enabled	boolean	○	性別認識有効・無効状態
CapTypeList	array	×	種類リスト(「male」「female」)
Age	object	×	年齢認識に関する情報
Enabled	boolean	○	年齢認識有効・無効状態
Expression	object	×	表情認識に関する情報
Enabled	boolean	○	表情認識有効・無効状態。
CapTypeList	array	×	種類リスト(「smile」等)
Gaze	object	×	視線認識に関する情報。
Enabled	boolean	○	視線認識有効・無効状態。
CapTypeList	array	×	種類リスト(「gaze」「nogaze」)
Distance	object	×	距離認識に関する情報
Enabled	boolean	○	距離認識有効・無効状態
CapTypeList	array	×	種類リスト(「Near」「Far」「Very Far」等)
NearLength	number	○	「near」として認識する距離0～NearLengthまでの範囲で人を認識した時に認識イベントを通知する。

FarLength	number	○	「Far」として認識する距離NearLength～FarLengthまでの範囲で人を認識した時に認識イベントを通知する。 FarLength～の範囲で人を認識した時は「Very Far」として認識する。
Recognition	object	×	顔認証に関する情報
Enabled	boolean	○	顔認証有効・無効状態
ImageList	array	○	比較対象画像ファイル名リスト ここで指定した画像と一致した時DataEventを通知する。(ワイルドカード指定可能)
Filters	object	×	
Gender	object	×	性別認識フィルタに関する情報
TypeList	array	○	フィルタ対象種類リスト 有効な値はCapTypeListで定義されている値。ここで指定した種類を認識対象とする。 フィルタを無効にするにはnullを指定する。
Score	number	○	認識スコア 有効な値は0～100。ここで指定したスコア以上を認識対象とする。フィルタを無効にするには-1を指定する。
Age	object	×	年齢認識フィルタに関する情報
Min	number	○	最低年齢 ここで指定した年齢以上を認識対象とする。フィルタを無効にするには、-1を指定する。
Max	number	○	最高年齢 ここで指定した年齢以下を認識対象とする。フィルタを無

						効にするには、-1を指定する。
			Expression	object	×	表情認識フィルタに関する情報
				TypeList	array	○	フィルタ対象種類リスト 有効な値はCapTypeListで定義されている値。ここで指定した種類を認識対象とする。 フィルタを無効にするにはnullを指定する。
				Score	number	○	認識スコア 有効な値は0～100。ここで指定したスコア以上を認識対象とする。フィルタを無効にするには-1を指定する。
			Gaze		object	×	視線認識フィルタに関する情報
				TypeList	array	○	フィルタ対象種類リスト 有効な値はCapTypeListで定義されている値。ここで指定した種類を認識対象とする。 フィルタを無効にするにはnullを指定する。
			Distance		object	×	距離認識フィルタに関する情報
				TypeList	array	○	フィルタ対象種類リスト 有効な値はCapTypeListで定義されている値。ここで指定した種類を認識対象とする。 フィルタを無効にするにはnullを指定する。

IndividualRecognitionInformationフォーマット

個体認識のIndividualRecognitionInformationプロパティでは、JSON形式で各種情報を定義する。

■基本項目

キー	値	値変更可否	説明
IndividualRecognitionInformation	object	×	各種個体認識の入力データ
[IndividualID]	object	×	個体の入力データが格納される。 キー名は個体を表すID。
DataLists	array<object>	×	対象個体の入力データリスト データの内容は、デバイスや機能毎に異なる。
[Data01]	-	-	
[Data02]	-	-	

■例：Face(顔認識)機能を持つデバイスの場合

キー	値	値変更可否	説明
IndividualRecognitionInformation	object	×	
Face	object	×	
DataLists	array<object>	×	
FaceID	string	×	認識した顔に割り当てられたID
FaceImageName	string	×	認識した顔画像ファイル名
Gender	object	×	認識した性別情報
Type	string	×	認識した種類 CapTypeList項目内の1つが設定される。
Score	number	×	認識した種類の信頼度
Age	object	×	認識した年齢情報
Age	number	×	認識した年齢
Expression	object	×	認識した表情情報
Type	string	×	認識した種類 CapTypeList項目内の1つが設定される。
Score	number	×	認識した種類の信頼度
Gaze	object	×	認識した顔ID毎の視線リスト。
Type	string	×	認識した種類 CapTypeList項目内の1つが設定される。
Distance	object	×	認識した距離情報
Type	string	×	認識した種類 CapTypeList項目内の1つが設定される。

			Recognition	object	×	認識した認証結果情報
			ImageName	string	×	一致した画像ファイル名

第 41 章
サウンドレコーダ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.16	Boolean	R/W	Open
CapCompareFirmwareVersion	1.16	Boolean	R	Open
CapPowerReporting	1.16	Int32	R	Open
CapStatisticsReporting	1.16	Boolean	R	Open
CapUpdateFirmware	1.16	Boolean	R	Open
CapUpdateStatistics	1.16	Boolean	R	Open
CheckHealthText	1.16	String	R	Open
Claimed	1.16	Boolean	R	Open
DataCount	1.16	Int32	R	Open
DataEventEnabled	1.16	Boolean	R/W	Open
DeviceControlDescription:	1.16	String	R	--
DeviceControlVersion:	1.16	Int32	R	--
DeviceEnabled	1.16	Boolean	R/W	Open & Claim
DeviceServiceDescription:	1.16	String	R	Open
DeviceServiceVersion:	1.16	Int32	R	Open
FreezeEvents	1.16	Boolean	R/W	Open
OutputID	1.16	Int32	R	*適用外*
PhysicalDeviceDescription:	1.16	String	R	Open
PhysicalDeviceName:	1.16	String	R	Open
PowerNotify	1.16	Int32	R/W	Open
PowerState	1.16	Int32	R	Open
State	1.16	Int32	R	Open
ControlObjectDescription	1.16	String	R	無し
ControlObjectVersion	1.16	Long	R	無し
ServiceObjectDescription	1.16	String	R	Open
ServiceObjectVersion	1.16	Long	R	Open
DeviceDescription	1.16	String	R	Open
DeviceName	1.16	String	R	Open

専用				
CapAssociatedHardTotalsDevice	1.16	String	R	Open
CapChannel	1.16	Boolean	R	Open
CapRecordingLevel	1.16	Boolean	R	Open
CapSamplingRate	1.16	Boolean	R	Open
CapSoundType	1.16	Boolean	R	Open
CapStorage	1.16	Int32	R	Open
Channel	1.16	String	R/W	Open, Claim, & Enable
ChannelList	1.16	String	R	Open
RecordingLevel	1.16	Int32	R/W	Open, Claim, & Enable
RemainingRecordingTimeInSec	1.16	Int32	R/W	Open, Claim, & Enable
SamplingRate	1.16	String	R/W	Open, Claim, & Enable
SamplingRateList	1.16	String	R	Open
SoundData	1.16	String	R	Open
SoundType	1.16	String	R/W	Open, Claim, & Enable
SoundTypeList	1.16	String	R	Open
Storage	1.16	Int32	R/W	Open, Claim, & Enable

メソッド

共通	*版数*	*使用可能条件*
Open	1.16	無し
Close	1.16	Open
Claim	1.16	Open
ClaimDevice	1.16	Open
Release	1.16	Open & Claim
ReleaseDevice	1.16	Open & Claim
CheckHealth	1.16	Open, & Enable
ClearInput	1.16	Open & Claim
ClearInputProperties	1.16	Open & Claim
ClearOutput	1.0	*適用外*
CompareFirmwareVersion	1.16	Open,& Enable
DirectIO	1.16	Open
ResetStatistics	1.16	Open, & Enable
RetrieveStatistics	1.16	Open, & Enable
UpdateFirmware	1.16	Open, & Enable
UpdateStatistics	1.16	Open, & Enable

専用

StartRecording	1.16	Open, Claim, & Enable
StopRecording	1.16	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.16	Open, Claim, & Enable
DirectIOEvent	1.16	Open, Claim
ErrorEvent	1.16	Open, Claim, & Enable
OutputCompleteEvent	1.16	*適用外*
StatusUpdateEvent	1.16	Open, Claim, & Enable
TransitionEvent	1.16	*適用外*

概説

サウンドレコーダコントロールのOLEプログラマティクIDは"OPOS.SoundRecorder"です。

機能

サウンドレコーダコントロールは次の機能を持っています。

音声を録音し、ファイルに記録します。

記録された音声データを、アプリケーションが読み取り可能なプロパティに渡します。

記録された音声データファイルを、デバイスのメモリやその他のストレージデバイスに保存します。

モデル

サウンドレコーダーコントロールは、広義の一般的な「デバイス入力モデル」を踏襲しています。ただし、サウンドレコーダーの場合は、録音プロセスの開始と停止、サウンドデータファイルの作成、データのプロパティへの配信、ファイルのデバイスおよび/または関連するストレージデバイスへの保存を行うためのメソッドの実行が必要である点が異なります。

サウンドレコーダーモデルでは、次のような動作を定義しています。サウンドレコーダデバイスは、サウンドレコーダデバイスを制御して、入力（録音）条件を設定し、入力データ取得の開始／終了を方法で指定します。また、取得した音声からリアルタイムにサウンドデータファイルを作成し、適切なプロパティにデータを配信します。同時に、録音されたデータファイルをデバイスおよび／または関連する記憶装置に保存します。

"**Sound Recorder** "デバイスコントロールは、startRecordingメソッドで録音を開始します。　startRecordingメソッドを実行するにあたり、CapChannelプロパティ、CapSamplingRateプロパティのそれぞれがtrueの場合は、Channelプロパティ、SamplingRateプロパティ、RecordingLevelプロパティの各値を設定する必要があります。DataEventEnabledプロパティもtrueに設定する必要があります。同様に、CapSoundTypeプロパティがtrueの場合は、SoundTypeプロパティの値から録音フォーマットを設定します。

指定された時間が経過するか、stopRecordingメソッドが呼び出されるか、clearInputメソッドが呼び出されると、録音が終了します。録音終了後、生成されたサウンドデータファイルは、ホストファイルまたはHard Totalsデバイスのいずれか、または両方に記録されます。生成されたサウンドデータは、SoundDataプロパティに配信されます。DataEventEnabled プロパティが true の場合、サウンドデータがプロパティに配信された直後に、DataEvent が待機状態となり、アプリケーションに配信されます。

AutoDisableプロパティがtrueであれば、DataEventがエンキューとなった後、デバイスは自動的にdisableになります。

残りの録音時間（秒）は、ReminingRecordingTimeInSecプロパティで取得できます。

ステータスが**SERC_SUE_START_SOUND_RECORDING**であるStatusUpdateEventは、startRecordingメソッドの実行時に呼び出され、アプリケーションに録音状態が開始されたことを通知します。

startRecording メソッドの指定時間が経過した場合、または stopRecording メソッドが呼び出された場合において録音が終了すると、ステータスが SERC_SUE_STOP_SOUND_RECORDING の StatusUpdateEvent が呼び出され、録音が停止されたことをアプリケーションに通知します。

DataEventEnabledプロパティがTRUEならば、キューイングしたDataEventはアプリケーションに通知されます。このイベントをエンキューにする直前に、デバイスは録音されたデータをSoundDataプロパティに提供します。また、DataEventEnabledプロパティをfalseに設定することで、それ以降のデータイベントを無効にします。これにより、後続の入力データは、アプリケーションが現在の入力と関連するプロパティを処理している間、デバイスによってバッファリングされます。アプリケーションが現在の入力の処理を終え、さらにデータを受け取る準備ができたら、DataEventEnabledをtrueに設定することによって、再びデータイベントが通知されるようになります。

ErrorEventのレスポンスがER_CONTINUEINPUTであれば、入力処理のプロセスは継続されます。ただし、エラーの原因が解決されない限り、すぐにErrorEventが再び通知されます。

ErrorEvent が ER_CLEAR であれば、入力処理プロセスを終了し、レコードが破棄されます。

startRecording メソッドで指定された時間が FOREVER（-1）の場合は、アプリケーション内で stopRecording メソッドが呼び出されるまで実行されます。stopRecording が呼び出されると、前回の録音データは指定されたファイル名でホストファイル、ハードトータ ルデバイス、またはその両方に記録され、サウンドデータは SoundData プロパティに配信されます。DataEventEnabled プロパティが true の場合、DataEvent がエンキューされ、アプリケーションも通知されます。入力データの読込み中または処理中に、コントロールにおいてエラーが発生した場合、ErrorEventが発行され、DataEventEnabled プロパティがTRUEならば、アプリケーションに通知されます。

startRecordingメソッドの呼び出しは、一度に1つしか実行できません。サウンドレコーダーの操作をネストさせようとすると、UposExceptionがスローされます。

startRecording メソッドの実行中に Error が発生した場合、アプリケーションは stopRecording メソッドを呼び出して記録処理を終了させるか、ErrorEvent の処理を終了させる前に clearInput メソッドを呼び出して記録処理をキャンセルする必要があります。

この後、stopRecording メソッドが呼び出されると、ErrorEvent が発生する直前までの録音データが、ホストファイル、Hard Totals デバイス、またはその両方に記録されます。DataEventEnabled プロパティが true の場合、DataEvent がエンキューされ、アプリケーションに通知されます。

startRecordingメソッドの実行中にErrorが発生しなければ、録音処理を終了し、いつでも録音を停止することができます。stopRecording メソッドが呼び出されると、メソッド呼び出しの直前までの録画データが、ホストファイル、ハードトータ ル、またはその両方に記録されます。DataEventEnabled プロパティが true の場合、DataEvent がエンキューされ、アプリケーションに配信されます。

clearInputメソッドを呼び出すことで、デバイスにキューイングされたすべての入力データを削除することができます。データイベントまたはエラーイベントの結果として入力されたすべてのデータプロパティは、clearInputPropertiesメソッドを呼び出すことで、デフォルト値に戻すことができます。

デバイスは、エンコードされたサウンドデータファイルをハードトータルデバイス、ホストファイルシステムのいずれか、または両方に書き込む機能を持っている場合があり、CapStorageプロパティはデバイスのデータ保存場所の機能を示します。

デバイスがハードトータルデバイスとホストファイルシステムのどちらか、または両方に対応している場合、アプリケーションはそれに応じてStorageプロパティを設定し、エンコードされたサウンドデータファイルをどこに書き込むかを設定する必要があります。

デバイスがエンコードされたサウンドデータを関連するハードトータルデバイスに書き込む場合、CapAssociatedHardTotalsDeviceプロパティには、関連するハード・トータルス・デバイスのオープンネームが格納されます。

デバイスの共有

サウンドレコーダは下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、デバイスの読み込みを始める前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の詳細な必要条件は、「一覧」を参照してください。

グラフィックスデバイスコントロールの画像表示モードは以下の通りです。

プロパティ

CapAssociatedHardTotalsDevice プロパティ

形式　STRING CapAssociatedHardTotalsDevice;

説明　**デバイスがハードトータルデバイスに書き込み可能な状態で、**CapStorageが**SREC_CST_ALL**，または**SREC_CST_HARDTOTALS_ONLY**のいずれかである場合、関連するハードトータルデバイスのオープンネームを保持します。
CapStorageプロパティがSREC_CST_HOST_ONLYの場合、このプロパティ値は空文字列でなければなりません。

参照　**CapStorage**プロパティ

CapChannel プロパティ

形式　BOOL CapChannel;

説明　TRUE：録音時のチャンネル変更が可能です。
FALSE：録音時のチャンネル変更ができません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**Channel**プロパティ

CapSamplingRate プロパティ

形式　BOOL CapSamplingRate;

説明　TRUE：録音時のサンプリングレート変更が可能です。
FALSE：録音時のサンプリングレート変更ができません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**SamplingRate**プロパティ

CapSoundType プロパティ

形式　BOOL CapSoundType;

説明　TRUE：録音時のファイル形式変更が可能です。
FALSE：録音時のファイル形式変更ができません。

このプロパティは、**Open**メソッドにより初期化されます。

参照 SoundTypeプロパティ

CapStorage プロパティ

形式 INT32 CapStorage;

説明 列挙型であり、録音したサウンドデータファイルの書込み先をデバイスが示します。下記のいずれかの値を有しています。

値	説明
SREC_CST_HARDTOTALS_ONLY	関連するハードトータルデバイスのみに対応
SREC_CST_HOST_ONLY	ホストのファイルシステムのみ対応
SREC_CST_ALL	関連するハードトータルデバイスとホストのファイルシステムの両方に対応

このプロパティは、Openメソッドにより初期化されます。

ハードトータルデバイスがサポートされている場合、プロパティ値はSREC_CST_HARDTOTALS_ONLY または SREC_CST_ALL とし、プロパティ CapAssociatedHardTotalsDevice には関連するハード・トータルス・デバイスのオープン名を格納します。

参照 Storageプロパティ、CapAssociatedHardTotalsDeviceプロパティ

CapRecordingLevel プロパティ

形式 BOOL CapRecordingLevel;

説明 TRUE：録音時の録音レベル変更が可能です。
FALSE：録音時の録音レベル変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照 CapRecordingLevelプロパティ

CapChannelList プロパティ

形式　BSTR CapChannelList;

説明　Channelプロパティに設定可能なチャンネル値をカンマ区切りのリストで示します。
例えば、デバイスが1, 2, 4チャンネルでの録音をサポートしている時は以下の様になります。
"1,2,4"

このプロパティは、Openメソッドにより初期化されます。

参照　Channelプロパティ

CapSamplingRateList プロパティ

形式　BSTR CapSamplingRateList;

説明　SamplingRateプロパティに設定可能なサンプリングレート値をカンマ区切りのリストで示します。
例えば、デバイスが44,1kHz, 48kHz, 96kHzをサポートしている時は以下の様になります。
"44100,48000,96000"

このプロパティは、Openメソッドにより初期化されます。

参照　SamplingRateプロパティ

CapSoundTypeList プロパティ

形式　BSTR CapSoundTypeList;

説明　SoundTypeプロパティに設定可能な音声ファイル形式をカンマ区切りのリストで示します。
例えば、デバイスがWAV, OGGをサポートしている時は以下の様になります。
"WAV,OGG"

このプロパティは、Openメソッドにより初期化されます。

参照　SoundTypeプロパティ

Channel プロパティ R/W

形式 BSTR Channel;

説明 録音時のチャンネルを示します。
有効な値は、CapChannelListプロパティに記された内の一つです。

このプロパティは、Openメソッドにより初期化されます。

参照 CapChannelプロパティ、CapChannelListプロパティ

ChannelList プロパティ

形式 BSTR ChannelList;

説明 デバイスがサポートしているチャンネルをカンマ区切りで表示します。例えば、デバイスがチャネル1とチャネル2とチャネル4のみの場合、このプロパティは「1,2,4」にセットされます。

このプロパティは、Openメソッドにより初期化されます。

参照 Channelプロパティ

RecordingLevel プロパティ R/W

形式 LONG RecordingLevel;

説明 録音時の録音レベルを示します。有効な値は0から100の範囲です。

このプロパティは、Openメソッドにより初期化されます。

戻り値 本プロパティ設定時、次の値がResultCodeプロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	ResultCodeの項目を参照してください。

参照 CapRecordingLevelプロパティ

RemainingRecordingTimeInSec プロパティ

形式 INT32 RemainingRecordingTimeInSec;

説明　録音中において、残りの録音時間を秒単位で保持します。録音が行われていない場合は0となります。**startRecording**メソッドの呼び出しが戻ってきたとき、引数**recordingTime**として渡された時間を保持します。

このプロパティは、デバイスの**setDeviceEnbaled**メソッドにより初期化されます。

参照　**startRecording**メソッド、**stopRecording**メソッド

SamplingRate プロパティ R/W

形式　BSTR SamplingRate;

説明　録音時のサンプリングレートを示します。
有効な値は、**CapSamplingRateList**プロパティに記された内の一つです。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	指定したサンプリングレートをサポートしません。
OPOS_E_BUSY	録音中のため、プロパティは設定できませんでした。
その他	**ResultCode**の項目を参照してください。

参照　**CapSamplingRate**プロパティ、**CapSamplingRateList**プロパティ

SamplingRateList プロパティ R

形式　BSTR SamplingRateList;

説明　デバイスがサポートしているサンプリングレートをカンマ区切りで表示します。例えば、デバイスが44.1kHzと48kHz、96kHzのみの場合、このプロパティは「44100,48000,96000」にセットされます。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	指定した音声ファイル形式をサポートしません。
OPOS_E_BUSY	録音中のため、プロパティは設定できませんでした。
その他	**ResultCode**の項目を参照してください。

参照　**SamplingRate**プロパティ

SoundData プロパティ R

形式　**BSTR SoundData;**

説明　StartRecordingメソッドまたはStopRecordingメソッドが呼び出されてから録音時間が経過した後のサウンドデータが格納されます。録音されたサウンドデータが無い場合はこのプロパティは長さが0になります。この値はDataEventがエンキューされる前に設定されます。

このデータのフォーマットはBinaryConversionプロパティの値に依存します。詳細はBinaryConversionプロパティを参照してください。

このプロパティは、**Open**メソッドにより長さ0に初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**StartRecording**メソッド、**StopRecording**メソッド、**データイベント**

SoundType プロパティ R/W

形式　**BSTR SoundType;**

説明　記録されるオーディオファイルのフォーマットを保持します。有効な値はCapSoundTypeListプロパティに設定されている値のいずれかになります。

このプロパティは、**Open**メソッドにより長さ0に初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**CapSoundType**プロパティ、**CapSoundTypeList**プロパティ

SoundTypeList プロパティ R

形式　**BSTR SoundTypeList;**

説明　デバイスがサポートしているサウンドファイル形式をカンマ区切りで表示します。例えば、デバイスがWAVとOGGのみの場合、このプロパティは「WAV,OGG」にセットされます。

このプロパティは、**Open**メソッドにより長さ0に初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**SoundType**プロパティ

Storage プロパティ R/W

形式 INT32 Storage;

説明 これは列挙型で、デバイスが録音したサウンドデータファイルをどこに書き込むかを定義します。startRecordingを呼び出す前に設定する必要があります。

次のいずれかの値を保持します。

値	説明
SREC_ST_HARDTOTALS	エンコードされたデータファイルは関連付けられたハードトータルデバイスに書き込まれます。CapAssociatedHardTotalsDeviceプロパティに関連付けされたHardTotalsデバイスのオープン名が格納されます。
SREC_ST_HOST	エンコードされたデータファイルはホストのファイルシステムに書き込まれます。
SREC_ST_HOST_HARDTOTALS	エンコードされたデータファイルは関連付けられたハードトータルデバイスとホストのファイルシステムに書き込まれます。CapAssociatedHardTotalsDeviceプロパティに関連付けされたHardTotalsデバイスのオープン名が格納されます。

このプロパティは、**Open**メソッドにより長さ0に初期化されます。

戻り値 本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照 **CapStorage**プロパティ、**CapAssociatedHardTotalsDevice**プロパティ

メソッド

StartRecording メソッド

形式　LONG StartRecording(BSTR *FileName*, BOOL *OverWrite*, LONG *RecordingTime*);

パラメータ	説明
FileName	記録する音声ファイル名を指定します。
OverWrite	同名ファイル存在時の挙動を指定します。TRUEならば上書きし、FALSEならエラーを返します。
RecordingTime	録音を行う時間を秒単位で指定します。OPOS_FOREVER(-1)を指定すると、**StopRecording**メソッドを呼ぶまで録音を続けます。

説明　音声の録音を開始します。

Channel プロパティ、**SamplingRate** プロパティ、**RecordingLevel** プロパティの設定を行い、**DataEventEnabled** プロパティを true に設定する必要があります。同時に、**SoundType** プロパティで録音フォーマットの設定を開始します。このメソッドを呼び出すと、指定された録音時間が経過すると録音処理が終了し、録音された音声データが **SoundData** プロパティに提供されるので、アプリケーションは **filename** 引数で与えられた保存された音声データファイルを読み込んだり、処理したりすることができます。**DataEventEnabled**プロパティがtrueの場合、**DataEvent**がエンキューされ、アプリケーションに配信されます。SREC_SUE_START_SOUND_RECORDINGのStatusUpdateEventは、startRecordingメソッドが実行されたときに発行され、アプリケーションに録音が開始されたことを通知します。startRecordingメソッドの指定時間が経過した場合、またはstopRecordingメソッドが呼び出されて録音が終了した場合、アプリケーションに録音が終了したことを通知するために、SREC_SUE_STOP_SOUND_RECORDINGのStatusUpdateEvent値が発行されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に開始されました。

OPOS_E_ILLEGAL	*FileName*が長すぎるか使用できない文字が含まれています、または*RecordhingTime*に0が指定されています。
OPOS_E_EXISTS	*FileName*がすでに存在します。(*OverWrite*がFALSEの時)
OPOS_E_BUSY	録音中なので実行できません。
その他	**ResultCode**の項目を参照してください。

参照 **Channel**プロパティ、**SamplingRate**プロパティ、**SoundData**プロパティ、**SoundType**プロパティ、**RecordingLevel**プロパティ、**StopRecording**メソッド、**StatusUpdateEvent**イベント

StopRecording メソッド

形式 **LONG StopRecording();**

説明 **StartRecording**メソッドで開始した録音処理を終了し、音声ファイルの記録を完了させます。

このメソッドは同期的に処理されます。録音お呼びでコード処理が終了すると**DataEventEnable**プロパティがTrueの場合、**DataEvent**イベントがエンキューされる前に**SoundData**プロパティに録音された音声データが記録されます。本メソッドが呼び出されるとSREC_SUE_STOP_SOUND_RECORDINGのStatusUpdateEvenイベントが発行され、アプリケーションに録音が停止したことを通知します。

戻り値 次の値のいずれかが戻され、ResultCodeプロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	録音されていません。
その他	**ResultCode**の項目を参照してください。

参照 **StartRecording**メソッド、**SoundData**プロパティ、**StatusUpdateEvent** イベント

イベント

DataEvent イベント

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

*Status*パラメータはゼロが設定されます。

説明　デバイスからアプリケーションに、SoundRecorderデバイスからのデータが読めるようになったことを通知します。このイベントが配信される前に、サウンドレコーダーの情報は**StartRecording**メソッドで指定された領域にエンキューされます。 保存されたサウンドレコーダーのデバイス情報は、関連する HardTotalsデバイスサービスによって管理される可能性があるため、アプリケーションはHard Totalsサービスもサポートする可能性があります。

参照　Channelプロパティ、SamplingRateプロパティ、SoundTypeプロパティ、RecordingLevelプロパティ、StopRecordingメソッド、StartRecordingメソッド

DirectIOEvent イベント

形式　**void DirectIOEvent (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。イベント番号とサービスオブジェクトで変化する特定値です。
pString	文字列のポインタ。イベント番号とサービスオブジェクトで変化する特定値です。このデータのフォーマットは**BinaryConversion**プロパティの値に依存します。詳細は**BinaryConversion**プロパティを参照してください。

説明　アプリケーションと直接通信するために、サービスオブジェクトが通知します。

このイベントにより、コントロールオブジェクトではサポートされないイベントを、サービスオブジェクトがアプリケーションに提供することが可能となります。

参照　DirectIOメソッド

ErrorEvent イベント

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

パラメータ	説明
ResultCode	エラーイベントを起こした原因のコード。値はResultCode参照。
ResultCodeExtended	エラーイベントを起こした原因の拡張コード。下記の値を参照。
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へのポインタ。下記の値を参照。

*ErrorLocus*パラメータの値は次のいずれかです。

値	説明
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が、すでにバッファリングされたデータの一部は利用できます。

もし、*ResultCode*がOPOS_E_EXTENDEDであれば*ResultCodeExtended*は次に示す値のいずれかに設定されます。

値	説明
OPOS_ESREC_NOROOM	ファイルを作成するのに十分なエリアがありません。

*pErrorResponse*パラメータが示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	説明
OPOS_ER_CLEAR	その非同期出力を再度試みます。エラー状態は抜けます。
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、**DataEventEnabled**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、**DataEventEnabled**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　録音中にエラーを検出すると通知されます。

正しいアプリケーションシーケンシングが行われるよう、**DataEventEnabled**プロパティがTRUEになるまで、入力エラーイベントは通知されません。

参照　「ステータス、結果コード、状態モデル」

StatusUpdateEvent イベント

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

パラメータ	説明
Status	ユニットの電源状態の変化または状態の変化を示します。

第1.3版以降では、電源状態通知機能が追加され、電源状態通知の**StatusUpdateEvent**値が追加されていることに注意してください。

ファームウェア更新の機能により、非同期更新ファームウェア更新処理の状態と進行状況を通知するためのステータス値が追加されました。共通プロパティ、メソッド、イベントの「**StatusUpdateEvent**」の説明を参照してください。

値	説明
SREC_SUE_START_SOUND_RECORDING	音声の録音が開始されると通知されます。
SREC_SUE_STOP_SOUND_RECORDING	音声の録音が停止すると通知されます。

説明　　サウンドレコーダデバイスの電源状態の変化または状態の変化があることをアプリケーションに通知します。

サウンドレコーダデバイスが電源状態の変化または状態の変化を検出すると、キューに入れられます。

参照　　「ステータス、結果コード、状態モデル」；「デバイス電源通知モデルについて」；**CapPowerReporting**プロパティ、**PowerNotify**プロパティ

第 42 章
音声認識

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.16	Boolean	R/W	Open
BinaryConversion	1.16	Long	R/W	Open
CapCompareFirmwareVersion	1.16	Boolean	R	Open
CapPowerReporting	1.16	Long	R	Open
CapStatisticsReporting	1.16	Boolean	R	Open
CapUpdateFirmware	1.16	Boolean	R	Open
CapUpdateStatistics	1.16	Boolean	R	Open
CheckHealthText	1.16	String	R	Open
Claimed	1.16	Boolean	R	Open
DataCount	1.16	Long	R	Open
DataEventEnabled	1.16	Boolean	R/W	Open
DeviceEnabled	1.16	Boolean	R/W	Open & Claim
FreezeEvents	1.16	Boolean	R/W	Open
OpenResult	1.16	Long	R	無し
OutputID	1.16	Long	R	*適用外*
PowerNotify	1.16	Long	R/W	Open
PowerState	1.16	Long	R	Open
ResultCode	1.16	Long	R	無し
ResultCodeExtended	1.16	Long	R	Open
State	1.16	Long	R	無し
ControlObjectDescription	1.16	String	R	無し
ControlObjectVersion	1.16	Long	R	無し
ServiceObjectDescription	1.16	String	R	Open
ServiceObjectVersion	1.16	Long	R	Open
DeviceDescription	1.16	String	R	Open
DeviceName	1.16	String	R	Open

専用				
CapLanguage	1.16	Boolean	R	Open
HearingResult	1.16	Long	R	Open & Claim & Enable
HearingStatus	1.16	Long	R	Open & Claim & Enable
HearingDataPattern	1.16	String	R	Open & Claim & Enable
HearingDataWord	1.16	String	R	Open & Claim & Enable
HearingDataWordList	1.16	String	R	Open & Claim & Enable
LanguageList	1.16	String	R	Open

メソッド

共通	*版数*	*使用可能条件*
Open	1.16	無し
Close	1.16	Open
ClaimDevice	1.16	Open
ReleaseDevice	1.16	Open & Claim
CheckHealth	1.16	Open, Claim, & Enable
ClearInput	1.16	Open & Claim
ClearInputProperties	1.16	Open & Claim
ClearOutput	1.16	*適用外*
CompareFirmwareVersion	1.16	Open, Claim, & Enable
DirectIO	1.16	Open
ResetStatistics	1.16	Open, Claim, & Enable
RetrieveStatistics	1.16	Open, Claim, & Enable
UpdateFirmware	1.16	Open, Claim, & Enable
UpdateStatistics	1.16	Open, Claim, & Enable

専用		
StartHearingFree	1.16	Open, Claim, & Enable
StartHearingSentence	1.16	Open, Claim, & Enable
StartHearingWord	1.16	Open, Claim, & Enable
StartHearingYesNo	1.16	Open, Claim, & Enable
StopHearing	1.16	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.16	Open, Claim, & Enable
DirectIOEvent	1.16	Open, Claim
ErrorEvent	1.16	Open, Claim, & Enable
OutputCompleteEvent	1.16	*適用外*
StatusUpdateEvent	1.16	Open, Claim, & Enable
TransitionEvent	1.16	*適用外*

概説

音声認識コントロールのOLEプログラマティクIDは
"OPOS.VoiceRecognition"です。

機能

音声認識コントロールは次の機能を持っています。

話し言葉を文字列に変換します。

モデル

音声認識コントロールは、イベント駆動入力の一般的な入力モデルに従います。

コントロールは**StartHearingYesNo**メソッドや**StartHearingSentence**メソッド等で音声認識を開始し、音声を認識したとき、DataEventを発生させます。

AutoDisableプロパティがTRUEならば、DataEventのキューイング時にコントロールは自動的にディセーブルとなります。

DataEventEnabledプロパティがTRUEならば、キューイングしたDataEventはアプリケーションに通知されます。コントロールはこのイベントを発生させる直前に、データをプロパティにコピーし、さらなるデータイベントを発生させないために、DataEventEnabledプロパティをFALSEに設定します。これにより、アプリケーションが現在の入力の処理や関連するプロパティの処理をしている間、コントロールはその後の入力データをキューイングするようになります。アプリケーションが現在の入力データの処理を終え、次のデータ処理の準備ができたとき、DataEventEnabledプロパティをTRUEに設定することにより、再びデータイベントが通知されるようになります。

入力データの読込み中または処理中に、コントロールにおいてエラーが発生した場合、ErrorEventが発行され、DataEventEnabledプロパティがTRUEのならば、アプリケーションに通知されます。

DataCountプロパティを読むことでコントロールによってキューイングされたデータイベント数が得られます。

コントロールにキューイングされたすべての入力データはClearInputメソッドを呼ぶことにより削除できます。

DataEventもしくはErrorEvent発生により入力された全てのデータプロパティは、ClearInputPropertiesメソッドを呼ぶ事によりデフォルト値に戻す事が出来ます。

音声認識の種類

音声認識は、認識する言葉の候補を指定し、その言葉を待ち受ける方式が主です。

音声認識の種類には次の4つがあります。

Yes/No/Cancel認識
デバイスで定義されたYes/No/Cancelに分類された単語の音声を待ち受けます。
例えば、「オッケー」という音声はYesに分類されます。
認識した内容は**HearingDataWord**プロパティに設定されます。
詳細は**StartHearingYesNo**メソッドを参照してください。

単語認識
アプリケーションが単語のリストを指定し、その単語の音声を待ち受けます。
認識した内容は**HearingDataWord**プロパティに設定されます。
詳細は**StartHearingWord**メソッドを参照してください。

文章認識
アプリケーションが単語と、それを使用した文章のパターンのリストを指定し、その文章の音声を待ち受けます。
認識した内容は**HearingDataWordList**プロパティ、**HearingDataPattern**プロパティに設定されます。
詳細は**StartHearingSentence**メソッドを参照してください。

自由認識
待受ける言葉を指定せずに、デバイスに任せた音声認識を行います。
認識した内容は**HearingDataWord**プロパティに設定されます。
詳細は**StartHearingFree**メソッドを参照してください。

音声を認識した時、どの種類で認識したかは**HearingResult**プロパティに格納されます。

デバイスの共有

音声認識は下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、デバイスの読み込みを始める前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の詳細な必要条件は、「一覧」を参照してください。

プロパティ

CapLanguage プロパティ

形式　BOOL CapLanguage;

説明　TRUE：認識言語の変更が可能です。
FALSE：認識言語の変更ができません。

このプロパティは、**Open**メソッドにより初期化されます。

HearingResult プロパティ

形式　LONG HearingResult;

説明　音声認識結果を示す値が設定されます。

設定されるパラメータは以下の通りです。

値	説明
VRCG_HRESULT_YESNO_YES	**StartHearingYesNo**メソッドの音声認識結果です。また、YESに分類される回答を得ました。 認識内容は**HearingDataWord**プロパティに設定されます。
VRCG_HRESULT_YESNO_NO	**StartHearingYesNo**メソッドの音声認識結果です。また、NOに分類される回答を得ました。 認識内容は**HearingDataWord**プロパティに設定されます。
VRCG_HRESULT_YESNO_CANCEL	**StartHearingYesNo**メソッドの音声認識結果です。また、CANCELに分類される回答を得ました。 認識内容は**HearingDataWord**プロパティに設定されます。
VRCG_HRESULT_WORD	**StartHearingWord**メソッドの認識結果です。 認識内容は**HearingDataWord**プロパティに設定されます。

VRCG_HRESULT_SENTENCE　StartHearingSentenceメソッドの認識結果です。
認識内容はHearingDataWordListプロパティ、HearingDataPatternプロパティに設定されます。

VRCG_HRESULT_FREE　StartHearingFreeメソッドの認識結果です。
認識内容はHearingDataWordプロパティに設定されます。

このプロパティは、DataEventが通知される直前にコントロールによって設定されます。

参照　HearingDataWordプロパティ、HearingDataWordListプロパティ、HearingDataPatternプロパティ、StartHearingYesNoメソッド、StartHearingWordメソッド、StartHearingSentenceメソッド、StartHearingFreeメソッド

HearingStatus プロパティ

形式　LONG HearingStatus;

説明　音声認識状態を示す値が設定されます。

設定されるパラメータは以下の通りです。

値	説明

VRCG_HSTATUS_NONE　音声認識は実行されていません。

VRCG_HSTATUS_YESNO　StartHearingYesNoメソッドによる音声認識実行中です。

VRCG_HSTATUS_WORD　StartHearingWordメソッドによる音声認識実行中です。

VRCG_HSTATUS_SENTENCE　StartHearingSentenceメソッドによる音声認識実行中です。

VRCG_HSTATUS_FREE　StartHearingFreeメソッドによる音声認識実行中です。

このプロパティは、Openメソッドにより初期化されます。また、音声認識状態が変わる直前にコントロールによって設定されます。

参照　StartHearingYesNoメソッド、StartHearingWordメソッド、StartHearingSentenceメソッド、StartHearingFreeメソッド

HearingDataPattern プロパティ

形式　BSTR HearingDataPattern;

説明　StartHearingSentenceメソッドで認識したパターンIDが設定されます。

このプロパティは、DataEventが通知される直前にコントロールによって設定されます。

参照　StartHearingSentenceメソッド

HearingDataWord プロパティ

形式　BSTR HearingDataWord;

説明　音声認識の認識内容が設定されます。

このプロパティは下記メソッドの入力データとして設定されます。どのメソッドに対するものかを知るには、HearingResultプロパティを確認します。

メソッド	説明
StartHearingYesNoメソッド	認識した単語が設定されます。
StartHearingWordメソッド	StartHearingWordメソッドで指定した単語候補の内、認識した単語が設定されます。
StartHearingFreeメソッド	認識した単語及び文章が設定されます。アルファベットの大文字小文字、日本語の漢字、ひらがな、カタカナ等、設定される内容はデバイスにより異なります。

このプロパティは、DataEventが通知される直前にコントロールによって設定されます。

参照　HearingResultプロパティ、StartHearingYesNoメソッド、StartHearingWordメソッド、StartHearingFreeメソッド

HearingDataWordList プロパティ

形式　BSTR HearingDataWordList;

説明　StartHearingSentenceメソッドで認識した単語情報をカンマ区切りのリストで示します。

各単語情報は以下の情報で構成されており、下記順番にコロン（“:”）で区切って示します。

パラメータ	説明
単語グループID	認識した単語グループID
単語	認識した単語。単語グループ内に定義された内容が設定されます。

例えば、StartHearingSentenceで以下のように候補を設定し、
　単語リスト："商品:コーヒー:紅茶,数:ひとつ:ふたつ"
　文章パターン："Pattern01:[商品]を[数],Pattern02:[商品]を下さい"
「コーヒーをひとつ」という言葉を認識した場合。
パターン「Pattern01」で、単語グループ「商品」の「コーヒー」と、「数」の「ひとつ」が認識される。
その時以下の様になります。
"商品:コーヒー,数:ひとつ"

このプロパティは、DataEventが通知される直前にコントロールによって設定されます。

参照　StartHearingSentenceメソッド

LanguageList プロパティ

形式　BSTR LanguageList;

説明　認識可能な言語をカンマ区切りのリストで示します。
言語を表す値は、RFC4646で定義されている言語、国コードからなる値です。
例えば、デバイスが米国/英語、日本/日本語をサポートしているときは、以下の様になります。
"en-US,ja-JP"

このプロパティは、Openメソッドにより初期化されます。

参照　StartHearingYesNoメソッド、StartHearingWordメソッド、StartHearingSentenceメソッド、StartHearingFreeメソッド

メソッド

StartHearingFree メソッド

形式　LONG StartHearingFree (BSTR *language*);

パラメータ	説明
language	認識する言語を指定します。**LanguageList**プロパティに記された内の一つを指定します。

説明　待受け候補を指定せずに待受けを開始します。

このメソッドは非同期的に実行され、**HearingDataWord**プロパティ、**HearingResult**プロパティ、**HearingStatus**プロパティに結果が設定されます。**StopHearing**メソッドを呼ぶことで音声認識を終了することができます。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	不正な値が指定されました。またはサポートしていない言語が指定されました。
OPOS_E_BUSY	音声認識中なので実行できません。
その他	**ResultCode**の項目を参照してください。

参照　**LanguageList**プロパティ、**StopHearing**メソッド

StartHearingSentence メソッド

形式　LONG StartHearingSentence(BSTR *language*, BSTR *WordList*, BSTR *PatternList*);

パラメータ	説明
language	認識する言語を指定します。**LanguageList**プロパティに記された内の一つを指定します。
WordList	待受け対象とする単語情報をカンマ区切りのリストで指定します。
PatternList	待受け対象とする文章パターン情報をカンマ区切りのリストで指定します。

*WordList*に指定する各単語情報は以下の情報で構成されており、下記順番にコロン（“:”）で区切って示します。

パラメータ	説明
単語グループID	単語リストを識別するID
単語リスト	コロン（“:”）で区切った、待受け対象とする単語候補

例えば、単具グループ「商品」には、単語候補「コーヒー」「紅茶」、単語グループ「数」には、単語候補「ひとつ」「ふたつ」を指定する場合、以下のように指定します。
"商品:コーヒー:紅茶,数:ひとつ:ふたつ"

*PatternList*に指定する各単語情報は以下の情報で構成されており、下記順番にコロン（“:”）で区切って示します。

パラメータ	説明
パターンID	パターンを識別するID
パターン	待受ける文章パターン。*WordList*に指定した単語リストを候補に加えるには、単語グループIDを"["と"]"で囲う。 例："[単語グループID1]を[単語グループID2]"

例えば、*WordList*に"商品:コーヒー:紅茶,数:ひとつ:ふたつ"が指定されており、「コーヒーをひとつ」、「コーヒーをください」と、商品と数を要求する音声を認識するパターンと、商品を要求する音声を認識するパターンを定義する場合は、以下の様に指定する。
"Pattern01:[商品]を[数],Pattern02:[商品]をください"

説明　*WordList*と*PatternList*で定義した文章の待受けを開始します。

このメソッドは非同期的に実行され、**HearingDataWord**プロパティ、**HearingResult**プロパティ、**HearingStatus**プロパティに結果が設定されます。**StopHearing**メソッドを呼ぶことで音声認識を終了することができます。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。

OPOS_E_ILLEGAL	次のいずれかが発生しました。 不正な値が指定されました、 サポートしていない言語が指定されました。 *Language*と*WordList*または*PatternList*の言語が一致しません。 *WordList*に無い*単語グループID*が*PatternList*で宣言されています。
OPOS_E_BUSY	音声認識中なので実行できません。
その他	**ResultCode**の項目を参照してください。

参照　**LanguageList**プロパティ、**StopHearing**メソッド

StartHearingWord メソッド

形式　**LONG StartHearingWord(BSTR** *language*, **BSTR** *WordList***);**

パラメータ	説明
language	認識する言語を指定します。**LanguageList**プロパティに記された内の一つを指定します。
WordList	待受け対象とする単語候補をカンマ区切りのリストで指定します。 例："単語1,単語2,単語3"

説明　*WordList*で指定した単語候補の待受けを開始します。

このメソッドは非同期的に実行され、**HearingDataWord**プロパティ、**HearingResult**プロパティ、**HearingStatus**プロパティに結果が設定されます。**StopHearing**メソッドを呼ぶことで音声認識を終了することができます。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	次のいずれかが発生しました。 不正な値が指定されました、 サポートしていない言語が指定されました。 *Language*と*WordList*の言語が一致しません。
OPOS_E_BUSY	音声認識中なので実行できません。

その他	**ResultCode**の項目を参照してください。

参照　**LanguageList**プロパティ、**StopHearing**メソッド

StartHearingYesNo メソッド

形式　**LONG StartHearingYesNo(BSTR** *language***);**

パラメータ	説明
language	認識する言語を指定します。**LanguageList**プロパティに記された内の一つを指定します。

説明　デバイスで定義された、「はい」「いいえ」「キャンセル」に該当する単語候補の待受けを開始します。

このメソッドは非同期的に実行され、**HearingDataWord**プロパティ、**HearingResult**プロパティ、**HearingStatus**プロパティに結果が設定されます。**StopHearing**メソッドを呼ぶことで音声認識を終了することができます。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	不正な値が指定されました。またはサポートしていない言語が指定されました。
OPOS_E_BUSY	音声認識中なので実行できません。
その他	**ResultCode**の項目を参照してください。

参照　**LanguageList**プロパティ、**StopHearing**メソッド

StopHearing メソッド

形式　**LONG StopHearing();**

説明　実行中の音声認識を終了します。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。

OPOS_E_ILLEGAL	音声認識は実行されていません。
その他	**ResultCode**の項目を参照してください。

参照　StartHearingYesNoメソッド、StartHearingWordメソッド、StartHearingSentenceメソッド、StartHearingFreeメソッド

イベント

DataEvent イベント

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

*Status*パラメータはゼロが設定されます。

説明　デバイスからアプリケーションに、VoiceRecognitionデバイスが音声を認識したことを通知します。

参照　HearingDataWordプロパティ、HearingDataWordListプロパティ、HearingDataPatternプロパティ、StartHearingYesNoメソッド、StartHearingWordメソッド、StartHearingSentenceメソッド、StartHearingFreeメソッド

DirectIOEvent イベント

形式　**void DirectIOEvent (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。イベント番号とサービスオブジェクトで変化する特定値です。
pString	文字列のポインタ。イベント番号とサービスオブジェクトで変化する特定値です。このデータのフォーマットは**BinaryConversion**プロパティの値に依存します。詳細は**BinaryConversion**プロパティを参照してください。

説明　アプリケーションと直接通信するために、サービスオブジェクトが通知します。

このイベントにより、コントロールオブジェクトではサポートされないイベントを、サービスオブジェクトがアプリケーションに提供することが可能となります。

参照　DirectIOメソッド

ErrorEvent イベント

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

パラメータ	説明
ResultCode	エラーイベントを起こした要因のコード。 値は**ResultCode**の項目を参照。
ResultCodeExtended	エラーイベントを起こした要因の拡張コード。値は下記の値を参照。
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へのポインタです。 下記の値を参照。

*ErrorLocus*パラメータの値は次のいずれかです。

値	意味
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。

アプリケーションはエラーイベントハンドラ内で*pErrorResponse*に次のいずれかの値を設定できます。

値	意味
OPOS_ER_CLEAR	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動作です。

OPOS_ER_CONTINUEINPUT

*ErrorLocus*が OPOS_EL_INPUT_DATA のときに限り設定されます。エラーを認識し、デバイスに処理の継続を指示します。デバイスはエラー状態のままですが、**DataEventEnabled** プロパティの設定にしたがって、別の **DataEvent** が引き続き通知されます。すべての入力データが通知されて、**DataEventEnabled** が再びTRUEに設定されたとき、*ErrorLocus*がOPOS_EL_INPUT に設定された新たな**ErrorEvent** 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　エラーが検出され、コントロールの**State**がエラー状態に遷移したときに通知します。

正しいアプリケーションシーケンシングが行われるよう、**DataEventEnabled**プロパティがTRUEになるまで、入力エラーイベントは通知されません。

参照　「ステータス、結果コード、状態モデル」

StatusUpdateEvent イベント

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

パラメータ	説明
Status	ユニットの電源状態の変化または状態の変化を示します。

第1.3版以降では、電源状態通知機能が追加され、電源状態通知の**StatusUpdateEvent**値が追加されていることに注意してください。

ファームウェア更新の機能により、非同期更新ファームウェア更新処理の状態と進行状況を通知するためのステータス値が追加されました。共通プロパティ、メソッド、イベントの「**StatusUpdateEvent**」の説明を参照してください。

説明 デバイスの電源状態の変化または状態の変化があることをアプリケーションに通知します。

デバイスが電源状態の変化または状態の変化を検出すると、キューに入れられます。

参照 「ステータス、結果コード、状態モデル」；「デバイス電源通知モデルについて」；CapPowerReportingプロパティ、PowerNotifyプロパティ

第 43 章
サウンドプレーヤ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.16	Boolean	R/W	*適用外*
CapCompareFirmwareVersion	1.16	Boolean	R	Open
CapPowerReporting	1.16	Int32	R	Open
CapStatisticsReporting	1.16	Boolean	R	Open
CapUpdateFirmware	1.16	Boolean	R	Open
CapUpdateStatistics	1.16	Boolean	R	Open
CheckHealthText	1.16	String	R	Open
Claimed	1.16	Boolean	R	Open
DataCount	1.16	Int32	R	*適用外*
DataEventEnabled	1.16	Boolean	R/W	*適用外*
DeviceEnabled	1.16	Boolean	R/W	Open & Claim
FreezeEvents	1.16	Boolean	R/W	Open
OutputID	1.16	Int32	R	Open
PowerNotify	1.16	Int32	R/W	Open
PowerState	1.16	Int32	R	Open
State	1.16	Int32	R	無し
DeviceControlDescription	1.16	String	R	無し
DeviceControlVersion	1.16	Int32	R	無し
DeviceServiceDescription	1.16	String	R	Open
DeviceServiceVersion	1.16	Int32	R	Open
PhysicalDeviceDescription	1.16	String	R	Open
PhysicalDeviceName	1.16	String	R	Open

専用	*版数*	*型*	*アクセス*	*使用可能条件*
CapAssociatedHardTotalsDevice	1.16	String	R	Open
CapMultiPlay	1.16	Boolean	R	Open
CapSoundTypeList	1.16	String	R	Open
CapStorage	1.16	Int32	R	Open
CapVolume	1.16	Boolean	R	Open
DeviceSoundList	1.16	String	R	Open
OutputIDList	1.16	String	R	Open & Claim & Enable
Storage	1.16	Int32	R/W	Open & Claim & Enable
Volume	1.16	Int32	R/W	Open & Claim & Enable

メソッド

共通	*版数*	*使用可能条件*
Open	1.16	無し
Close	1.16	Open
ClaimDevice	1.16	Open
ReleaseDevice	1.16	Open & Claim
CheckHealth	1.16	Open, Claim, & Enable
ClearInput	1.16	Open, Claim, & Enable
ClearInputProperties	1.16	Open, Claim
ClearOutput	1.16	適用外
CompareFirmwareVersion	1.16	Open, Claim, & Enable
DirectIO	1.16	Open
ResetStatistics	1.16	Open, Claim, & Enable
RetrieveStatistics	1.16	Open, Claim, & Enable
UpdateFirmware	1.16	Open, Claim, & Enable
UpdateStatistics	1.16	Open, Claim, & Enable
専用		
PlaySound	1.16	Open, Claim, & Enable
StopSound	1.16	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.16	適用外
DirectIOEvent	1.16	Open, Claim
ErrorEvent	1.16	Open, Claim, & Enable
OutputCompleteEvent	1.16	Open, Claim, & Enable
StatusUpdateEvent	1.16	Open, Claim, & Enable
TransitionEvent	1.16	*適用外*

概説

サウンドプレーヤコントロールのOLEプログラマティクIDは"OPOS.SoundPlayer"です。

機能

サウンドプレーヤコントロールは次の機能を持っています。

- 音声ファイルを再生します。

モデル

サウンドプレーヤコントロールは、非同期的な出力モデルに従います。

- デバイスはメソッドパラメータを検証し、必要に応じてエラーを生成します。検証が成功すると次の処理を行います。
- オーディオファイルは順番に再生されます。playSoundメソッドが呼ばれると、デバイスはメソッドパラメータで指定されたサウンドの再生を開始し、サウンドファイルデータはキューに配置され、対応するOutputIDはOutputIDプロパティに格納され、OutputIDListプロパティに追加されます。OutputIDプロパティには一意の整数識別子がセットされます。
- サウンドの再生が開始されると、値SPLAY_SUE_START_PLAY_SOUNDのStatusUpdateEventが通知されます。
  サウンドの再生が終了すると、OutputCompleteEventをアプリケーションに通知するためにキューに格納され、対応するOutputIDがOutputIDプロパティに格納されます。同時に値SPLY_SUE_STOP_PLAY_SOUNDのStatusUpdateEventが通知されます。アプリケーションは、返されたOutputCompleteEventプロパティのOutputIDの値と、データの送信に使用される非同期プロセスメソッドの呼び出しによって設定されたOutputIDの値を比較して、デバイスに正常にデータ送信されたかを追跡する必要があります。
- stopSoundメソッドが呼び出されると、デバイスはOutputIDプロパティの値に従って再生を停止し、キューに格納されたサウンドファイルデータがクリアされます。このメソッド実行後、対応するOutputIDプロパティとOutputIDListの値は更新されません。OutputCompleteEventは発生せず、値SPLY_SUE_STOP_PLAY_SOUNDのStatusUpdateEventのみが通知されます。

- 要求の処理中にエラーが発生した場合、ErrorEventがキューに格納されます。これはOutputCompleteEventを含め、既にキューに格納されているイベントの後にアプリケーションに通知されます。イベントがアプリケーションに通知されるまで、以降の非同期出力は発生しません。応答がER_CLEARの場合は、未処理の非同期出力はクリアされます。応答がER_RETRYの場合は、再試行されます。エラー検出時に複数の出力が同時に進行中であった場合は、サービスはこれらの全ての出力を再試行する必要があることに注意してください。
- 非同期出力は、常に先入先出方で実行されます。デバイスが同時再生をサポートしている場合、要求は同時に実行されます。デバイスが同時再生をサポートしているかどうかを確認するには、CapMultiPlayプロパティを確認して下さい。
- アプリケーションがclearOutputメソッドを呼び出すなどの理由で、要求が完了する前に終了した場合、OutputCompleteEventは通知されません。
- アプリケーションは、stopSoundメソッドを呼び出すことにより、出力を個別に削除することも可能です。またこの場合、OutputCompleteEventは通知されません。
- CapSoundTypeListプロパティは、デバイスが再生可能なオーディオファイルタイプの一覧を表示します。
- アプリケーションは、StatusUpdateEventによりステータスの変化について認識します。また対応する全てのステータスプロパティは、イベントが通知される前に更新されます。
- デバイスが、HardTotalsDeviceとhost file systemのいずれかまたは両方をサポートしている場合、アプリケーションは、それに応じて、データファイルにアクセスする場所をStorageプロパティに設定する必要があります。
- デバイスが、HardTotalsDeviceからplaySoundメソッドで再生されたオーディオファイルにアクセスできる必要がある場合、CapAssociatedHardTotalsDeviceプロパティは、関連付けられHardTotalsDeviceのオープン名を保持します。

デバイスの共有

サウンドプレーヤは下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、プロパティにアクセスしたりメソッドを呼び出してデバイスの更新をする前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の詳細な必要条件は、「一覧」を参照してください。

プロパティ

CapAssociatedfHardTotalsDevice プロパティ

形式　BSTR CapAssociatedHardTotalsDevice;

説明　CapStorageがPLY_CST_ALLまたはPLY_CST_HARDTOTALS_ONLYの場合のように、デバイスに書込みむことが出来る場合、関連するHard Totals deviceのオープン名を保持します。CapStorageがSPLY_CST_HOST_ONLYの場合、このプロパティ値は空の文字列である必要があります。
このプロパティは、Openメソッドにより初期化されます。

参照　CapStorageプロパティ

CapMultiPlay プロパティ

形式　BOOL CapMultiPlay;

説明　TRUE　：音声の同時再生をサポートしています。
FALSE　：上記機能をサポートしません。
このプロパティは、Openメソッドにより初期化されます。

参照　PlaySoundメソッド

CapSoundTypeList プロパティ

形式　BSTR CapSoundTypeList;

説明　再生可能な音声ファイル形式をカンマ区切りのリストで示します。

例えば、デバイスがWAV, OGGをサポートしている場合、このプロパティは"WAV,OGG"に設定する必要があります。
このプロパティは、Openメソッドにより初期化されます。

参照　PlaySoundメソッド

CapStorage プロパティ

形式　LONG CapStorage;

説明　列挙型で、デバイスが録音サウンドデータファイルをどこに書き込めるかを通知します。
次のいずれかの値を保持します。

値	意味
SPLY_CST_HARDTOTALS_ONLY	関連するHardTotalsdeviceのみがサポートされます。
SPLY_CST_HOST_ONLY	ホストファイルシステムのみがサポートされます。
SPLY_CST_ALL	関連するHardTotalsdeviceとホストファイルシステムの両方がサポートされます。

このプロパティは、**Open**メソッドにより初期化されます。
HardTotalsdeviceがストレージをサポートしている場合、プロパティ値はSPLY_CST_HARDTOTALS_ONLYまたはSPLY_CST_ALLである必要があり、プロパティCapAssociagtedHardTotalsDeviceは関連するHardTotalsdeviceのオープン名を保持します。

CapVolume プロパティ

形式　**BOOL CapVolume;**

説明　TRUE　：再生時のボリュームの変更が可能です。
FALSE　：再生時のボリュームの変更ができません。
このプロパティは、**Open**メソッドにより初期化されます。

参照　**Volume**プロパティ

DeviceSoundList プロパティ

形式　**BSTR DeviceSoundList;**

説明　デバイスでサポートされているデバイスサウンドIDのカンマ区切りのリストが含まれています。
このプロパティは、**Open**メソッドにより初期化されます。

参照　**PlaySound**メソッド

OutputIDList プロパティ

形式　　BSTR OutputIDList;

説明　　PlaySoundメソッドで再生されるOutputIDのカンマ区切りリストが含まれます。
このプロパティは、Openメソッドにより初期化されます。また、音声再生リクエストの増減により更新されます。

参照　　PlaySoundメソッド

Storage プロパティ

形式　　LONG Storage;

説明　　次のいずれかの値を保持します。

値	意味
SPLY_ST_HARDTOTALS	エンコードされたデータファイルは、関連するHardTotalsデバイスに書き込まれます。プロパティCapAssocatedHardTotalsSDeviceは、関連付けられたHardTotalsdeviceのオープンを保持します。
SPLY_ST_HOST	エンコードされたデータファイルは関連するHardTotalsdeviceとhostfilesystemに書き込まれます。
SPLY_ST_HOST_HARDTOTALS	エンコードされたデータファイは関連するHardTotalsdeviceとhostfilesystemに書き込まれます。プロパティCapAssociatedHardTotalsDeviceは、関連付けられたHardTotalsdeviceのオープン名を保持します。

このプロパティは、CapStoregeが保持する値に従ってopenメソッドによって初期化されます。CapStorageがSPLY_CST_ALLの場合、SPLY_ST_HOST_HARDTOTALSに初期化されます。

参照　　CapStoregeメソッド

Volume プロパティ R/W

形式　　LONG Volume;

説明　再生時の音量を示します。有効な値は0から100の範囲です。
このプロパティは、**Open**メソッドにより初期化されます。

参照　**PlaySound**メソッド

メソッド

PlaySound メソッド

形式　LONG PlaySound(BSTR *FileName*, BOOL *Loop*);

パラメータ	説明
FileName	再生する音声ファイル名を指定します。または**DeviceSoundList**プロパティ記される音声IDリストの1つを指定します。
LOOP	TRUEならばループ再生し、FALSEならループ再生しません。

説明　*FileName*またはデバイス定義サウンドで指定されたオーディオファイルを再生します。

オーディオファイルは、「HardTotals」サービスによって管理されているエリアに配置する必要があります。

このメソッドは非同期で実行されます。再生を途中で終了するには、**StopSound**メソッドを呼びます。

エラー　例外時のErrorCodeプロパティの例は次の通りです。

値	意味
E_ILLEGAL	無効な値が指定されました。またはサポートされていないサウンドファイルが指定されました。
E_NOEXIST	ファイルが存在しません。

参照　**CapSoundType**プロパティ、**DeviceSoundList**プロパティ、**StopSound**メソッド

StopSound　メソッド

形式　LONG StopSound(LONG *OutputID*);

パラメータ	説明
OutputID	終了したい音声のOutputIDプロパティ値を指定します。

説明　*OutputID*で指定した音声再生を終了します。

エラー　例外時のErrorCodeプロパティの例は次の通りです。

値	意味
E_ILLEGAL	指定されたサウンドは再生されていません。

参照　OutputIDプロパティ、StartSoundメソッド

イベント

DirectIOEvent イベント

形式　void DirectIOEvent (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。イベント番号とサービスオブジェクトで変化する特定値です。
pString	文字列のポインタ。イベント番号とサービスオブジェクトで変化する特定値です。このデータのフォーマットはBinaryConversionプロパティの値に依存します。詳細はBinaryConversionプロパティを参照してください。

説明　アプリケーションと直接通信するために、サービスオブジェクトが通知します。

このイベントにより、コントロールオブジェクトではサポートされないイベントを、サービスオブジェクトがアプリケーションに提供することが可能となります。

参照　DirectIOメソッド

ErrorEvent イベント

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

パラメータ	説明
ResultCode	エラーイベントを起こした原因のコード。値は**ResultCode**参照。
ResultCodeExtended	エラーイベントを起こした原因の拡張コード。下記の値を参照。
ErrorLocus	OPOS_EL_OUTPUTに設定します。非同期記録データ格納処理中にエラーが発生したことを示す。
pErrorResponse	エラーイベント応答へのポインタ。下記の値を参照。

もし、*ResultCode*がOPOS_E_EXTENDEDであれば、*ResultCodeExtended*は次に示す値のいずれかに設定されます。

値	意味
OPOS_ESPLY_NOROOM	符号化されたデータの記憶領域に十分な余裕がありません。

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。

*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	再試行します。エラー状態は解除されます。*ErrorLocus*がOPOS_EL_OUTPUTの場合のデフォルト動作です。
OPOS_ER_CLEAR	全てのバッファリングされている非同期出力を含む出力データをクリアします。エラー状態は解除されます。ClearOutputメソッドと同じ働きをします。

説明　エラーが検出され、コントロールの**State**がエラー状態に遷移したときに通知します。

正しいアプリケーションシーケンシングが行われるよう、**DataEventEnabled**プロパティがTRUEになるまで、入力エラーイベントは通知されません。

参照　「**ステータス、結果コード、状態モデル**」

OutputCompleteEvent イベント

形式　void OutputCompleteEvent (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　前に開始された非同期出力リクエストが正常に終了したときに通知します。

参照　「**出力モデルについて、OutputIDプロパティ**」

StatusUpdateEvent イベント

形式　　void StatusUpdateEvent (LONG *Status*);

*Status*は、次に示す値のいずれかに設定されます。

値	意味
SPLY_SUE_START_PLAY_SOUND	サウンド再生が開始しました。
SPLY_SUE_STOP_PLAY_SOUND	サウンド再生が停止しました。

説明　　重要な状態の変化が起こった時に通知されます。

第 44 章
音声合成

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.16	Boolean	R/W	*適用外*
BinaryConversion	1.16	Long	R/W	Open
CapCompareFirmwareVersion	1.16	Boolean	R	Open
CapPowerReporting	1.16	Long	R	Open
CapStatisticsReporting	1.16	Boolean	R	Open
CapUpdateFirmware	1.16	Boolean	R	Open
CapUpdateStatistics	1.16	Boolean	R	Open
CheckHealthText	1.16	String	R	Open
Claimed	1.16	Boolean	R	Open
DataCount	1.16	Long	R	*適用外*
DataEventEnabled	1.16	Boolean	R/W	*適用外*
DeviceEnabled	1.16	Boolean	R/W	Open & Claim
FreezeEvents	1.16	Boolean	R/W	Open
OpenResult	1.16	Long	R	無し
OutputID	1.16	Long	R	Open
PowerNotify	1.16	Long	R/W	Open
PowerState	1.16	Long	R	Open
ResultCode	1.16	Long	R	無し
ResultCodeExtended	1.16	Long	R	Open
State	1.16	Long	R	無し
ControlObjectDescription	1.16	String	R	無し
ControlObjectVersion	1.16	Long	R	無し
ServiceObjectDescription	1.16	String	R	Open
ServiceObjectVersion	1.16	Long	R	Open
DeviceDescription	1.16	String	R	Open
DeviceName	1.16	String	R	Open

専用				
CapLanguage	1.16	Boolean	R	Open
CapPitch	1.16	Boolean	R	Open
CapSpeed	1.16	Boolean	R	Open
CapVoice	1.16	Boolean	R	Open
CapVolume	1.16	Boolean	R	Open
Language	1.16	String	R/W	Open, Claim, & Enable
LanguageList	1.16	String	R	Open
OutputIDList	1.16	String	R	Open, Claim, & Enable
Pitch	1.16	Long	R/W	Open, Claim, & Enable
Speed	1.16	Long	R/W	Open, Claim, & Enable
Voice	1.16	String	R/W	Open, Claim, & Enable
VoiceList	1.16	String	R	Open
Volume	1.16	Long	R/W	Open, Claim, & Enable

メソッド

共通	*版数*	*使用可能条件*
Open	1.16	無し
Close	1.16	Open
ClaimDevice	1.16	Open
ReleaseDevice	1.16	Open & Claim
CheckHealth	1.16	Open, Claim, & Enable
ClearInput	1.16	*適用外*
ClearInputProperties	1.16	*適用外*
ClearOutput	1.16	Open, Claim, & Enable
CompareFirmwareVersion	1.16	Open, Claim, & Enable
DirectIO	1.16	Open
ResetStatistics	1.16	Open, Claim, & Enable
RetrieveStatistics	1.16	Open, Claim, & Enable
UpdateFirmware	1.16	Open, Claim, & Enable
UpdateStatistics	1.16	Open, Claim, & Enable

専用		
Speak	1.16	Open, Claim, & Enable
SpeakImmediate	1.16	Open, Claim, & Enable
StopCurrentSpeaking	1.16	Open, Claim, & Enable
StopSpeaking	1.16	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.16	*適用外*
DirectIOEvent	1.16	Open & Claim
ErrorEvent	1.16	Open, Claim, & Enable
OutputCompleteEvent	1.16	Open, Claim, & Enable
StatusUpdateEvent	1.16	Open, Claim, & Enable
TransitionEvent	1.16	*適用外*

概説

音声合成コントロールのOLEプログラマティクIDは
"OPOS.SpeechSynthesis"です。

機能

音声合成コントロールは次の機能を持っています。

テキストを音声に変換し、発話します。

モデル

音声合成コントロールは、非同期的な出力モデルに従います。

アプリケーションは**Speak**メソッド、**SpeakImmediate**メソッドを呼び出して出力を開始します。コントロールはプログラムメモリー内のリクエストをバッファリングし、このリクエストの一意な整数の識別子をOutputIDプロパティに設定して、できる限り迅速に戻ります。デバイスがリクエストを正常に終了すると、OPOSがOutputCompleteEventを通知します。このイベントのパラメータは完了したリクエストのOutputIDです。アプリケーションは、デバイスへのデータ送信に成功したことを追跡するために、非同期処理メソッドの呼び出しによって設定されたOutputIDと、OutputCompleteEventのOutputIDプロパティを比較します。

Speakメソッド、**SpeakImmediate**メソッドが呼び出されると、**Language**プロパティ、**Volume**プロパティ、**Pitch**プロパティ、**Speed**プロパティに設定された内容で音声合成を開始します。また、キューイングされた音声合成要求を一意に識別するOutputIDが**OutputID**プロパティに格納され、リストされた値として**OutputIDList**プロパティに追加されます。

Speakメソッド、**SpeakImmediate**メソッドにより発話が開始されると、**StatusUpdateEvent**イベントによりSPSY_SUE_START_SPEAKが通知されます。発話が完了するとキューイングされたときに**OutputID**プロパティに設定された値で**OutputCompleteEvent**イベントが通知され、同時に**StatusUpdateEvent**イベントによりSPSY_SUE_STOP_SPEAKが通知されます。

Speakメソッド、**SpeakImmediate**メソッドによる発話中に**SpeakImmediate**メソッドを呼び出すと、発話は即座に中断され、**StatusUpdateEvent**イベントによりSPSY_SUE_STOP_SPEAKが通知されますが、**OutputCompleteEvent**イベントが通知されず、**OutputID**プロパティと**OutputIDList**プロパティの値に対応する現在の**Speak**メソッドまたは**SpeakImmediate**メソッドは変更されません。

非同期リクエストを実行しているときにエラーが発生すると、ErrorEventが通知されます。アプリケーションのイベントハンドラは、未処理の出力を再試行するかまたは、それをクリアできます。ErrorEventが処理中は、コントロールはErrorの状態です。（備考：エラーを引き起こす原因が除去されていない場合は、コントロールは直ちにError状態に再びなり、別のErrorEventを通知します。）

非同期出力はFIFOベースで実行されます。

コントロールがバッファリングした出力はすべて（非同期出力をすべて含む）、ClearOutputメソッドを呼び出すことによって削除できます。クリアした出力に対してOutputCompleteEventは通知されません。このメソッドは（可能ならば）処理中の出力も停止します。また、**StopCurrentSpeaking**メソッドや**StopSpeaking**メソッドを呼び出すことで、出力を個別に削除できます。この場合もOutputCompleteEventは通知されません。

デバイスの共有

音声合成は下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、デバイスの読み込みを始める前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の詳細な必要条件は、「一覧」を参照してください。

プロパティ

CapLanguage プロパティ

形式　BOOL CapLanguage;

説明　TRUE：発話する言語の変更が可能です。
FALSE：発話する言語の変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　Languageプロパティ

CapPitch プロパティ

形式　BOOL CapPitch;

説明　TRUE：発話音声の高低の変更が可能です。
FALSE：発話音声の高低の変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　Pitchメソッド

CapSpeed プロパティ

形式　BOOL CapSpeed;

説明　TRUE：発話音声の速さの変更が可能です。
FALSE：発話音声の速さの変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　Speedメソッド

CapVoice プロパティ

形式　BOOL CapVoice;

説明　TRUE：発話する声色の変更が可能です。
FALSE：発話する声色の変更ができません。

このプロパティは、Openメソッドにより初期化されます。

参照　Voiceメソッド

CapVolume プロパティ

形式　BOOL CapVolume;

説明　TRUE：発話音声の音量の変更が可能です。
FALSE：発話音声の音量の変更ができません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**Volume**メソッド

Language プロパティ R/W

形式　BSTR Language;

説明　発話する言語を示します。有効な値はLanguageListプロパティに記された内の一つです。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。またはサポートしていない言語が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**Speak**メソッド、**SpeakImmediate**メソッド

LanguageList プロパティ

形式　BSTR LanguageList;

説明　発話可能な言語をカンマ区切りのリストで示します。
言語を表す値は、RFC4646で定義されている言語、国コードからなる値です。
例えば、デバイスが米国/英語、日本/日本語をサポートしているときは、以下の様になります。
"en-US,ja-JP"

このプロパティは、**Open**メソッドにより初期化されます。

参照　**Language**メソッド

OutputIDList プロパティ

形式　BSTR OutputIDList;

説明　Speakメソッドまたは**SpeekImmediate**メソッドで再生されている音声の**OutputID**プロパティ値をカンマ区切りのリストで示します。

このプロパティは、**Open**メソッドにより初期化されます。また、発話リクエストの増減により更新されます。

参照　Speakメソッド、SpeakImmediateメソッド

Pitch プロパティ R/W

形式　LONG Pitch;

説明　発話音声の高低を示します。有効な値は50%から200%の範囲です。

このプロパティは、**Open**メソッドにより100%に初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　Speakメソッド、SpeakImmediateメソッド

Speed プロパティ R/W

形式　LONG Speed;

説明　発話音声の速度を示します。有効な値は50%から200%の範囲です。

このプロパティは、**Open**メソッドにより100%に初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。

その他	**ResultCode**の項目を参照してください。

参照 **Speak**メソッド、**SpeakImmediate**メソッド

Voice プロパティ R/W

形式 BSTR Voice;

説明 発話する声色を示します。有効な値は**VoiceList**プロパティに記された内の一つです。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値 本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。またはサポートしていない声色が指定されました。
その他	**ResultCode**の項目を参照してください。

参照 **Speak**メソッド、**SpeakImmediate**メソッド

VoiceList プロパティ

形式 BSTR VoiceList;

説明 発話可能な声色をカンマ区切りのリストで示します。
例えば、デバイスが男性、女性の声色をサポートしているときは、以下の様になります。
"MALE_VOICE,FEMALE_VOICE"
（値の内容についてはデバイスに依存します）

このプロパティは、**Open**メソッドにより初期化されます。

参照 **Voice**メソッド

Volume プロパティ R/W

形式 LONG Volume;

説明 発話音声の音量を示します。有効な値は0から100の範囲です。

このプロパティは、**Open**メソッドにより100に初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**Speak**メソッド、**SpeakImmediate**メソッド

メソッド

Speak メソッド

形式　LONG Speak(BSTR *Text*);

パラメータ	説明
Text	発話するテキストを指定します。

説明　*Text*で指定した文言を発話します。

発話は**Language**プロパティ、**Volume**プロパティ、**Pitch**プロパティ、**Speed**プロパティの設定内容に従って実行しますが、テキスト内に下記のタグを挿入することで、タグ以降の発話に変化を加えることができます。

タグ	説明
volume	発話音声の音量を指定します。有効な値は1から100の範囲です。
pitch	発話音声の高低を指定します。有効な値は50から200の範囲です。
speed	発話音声の速さを指定します。有効な値は50から200の範囲です。
pause	一時停止する時間をミリ秒で指定します。
reset	*volume, pitch, speed*の効果を削除します。

*reset*を除いたタグは"¥¥タグ=値¥¥"の形式で指定します。
例えば*Text*を以下の様に指定した時、
"こんにちは¥¥pause=1000¥¥¥¥pitch=150¥¥今日は¥¥reset¥¥いい天気ですね"
「こんにちは」は元の設定に従い発話します。
その後1000ミリ秒Waitします。
「今日は」はPitchを150%で発話します。
「いい天気ですね」は元の設定に従い発話します。

デバイスがVolumeの変更等をサポートしていない場合、そのタグは無視されます。

このメソッドは非同期的に実行されます。発話を途中で終了するには、**StopCurrentSpeaking**メソッドか**StopSpeaking**メソッドを呼びます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が指定されました。**Language**プロパティで設定されている言語と*Text*で指定されている言語が一致しません。
その他	**ResultCode**の項目を参照してください。

参照　**LanguageList**プロパティ、**Volume**プロパティ、**Pitch**プロパティ、**Speed**プロパティ、**StopCurrentSpeaking**メソッド、**StopSpeaking**メソッド

SpeakImmediate メソッド

形式　LONG SpeakImmediate(BSTR *Text*);

パラメータ	説明
Text	発話するテキストを指定します。

説明　**ClearOutput**メソッドと同じ処理を実行後に、*Text*で指定した文言を発話します。

このメソッドも**Speak**メソッドと同様にタグを挿入することで特定文言に変化を加えることができます。詳細は**Speak**メソッドの説明を参照して下さい。

このメソッドは非同期的に実行されます。発話を途中で終了するには、**StopCurrentSpeaking**メソッドか**StopSpeaking**メソッドを呼びます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が指定されました。**Language**プロパティで設定されている言語と*Text*で指定されている言語が一致しません。
その他	**ResultCode**の項目を参照してください。

参照 **Language**プロパティ、**Volume**プロパティ、**Pitch**プロパティ、**Speed**プロパティ、**Speak**メソッド、**StopCurrentSpeaking**メソッド、**StopSpeaking**メソッド

StopCurrentSpeaking メソッド

形式 LONG StopCurrentSpeaking();

説明 現在実行中の発話を停止します。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	発話は実行されていません。
その他	**ResultCode**の項目を参照してください。

参照 **Speak**メソッド、**SpeakImmediate**メソッド

StopSpeaking メソッド

形式 LONG StopSpeaking(LONG *OutputID*);

パラメータ	説明
OutputID	終了したい発話に関連付けられた**OutputID**プロパティ値を指定します。

説明 *OutputID*に指定した発話を停止及び削除します。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照 **OutputID**プロパティ、**Speak**メソッド、**SpeakImmediate**メソッド

イベント

DirectIOEvent イベント

形式　void DirectIOEvent (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。イベント番号とサービスオブジェクトで変化する特定値です。
pString	文字列のポインタ。イベント番号とサービスオブジェクトで変化する特定値です。このデータのフォーマットはBinaryConversionプロパティの値に依存します。詳細はBinaryConversionプロパティを参照してください。

説明　アプリケーションと直接通信するために、サービスオブジェクトが通知します。

このイベントにより、コントロールオブジェクトではサポートされないイベントを、サービスオブジェクトがアプリケーションに提供することが可能となります。

参照　DirectIOメソッド

ErrorEvent イベント

形式　void ErrorEvent (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。

*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	データ送信を再試行します。エラー状態は解除されます。いくつかの入力デバイスで非同期出力デバイスで*ErrorLocus*がOPOS_EL_OUTPUTの場合、エラー状態が解除され、再送信を行うときに使用します。*ErrorLocus*がOPOS_EL_OUTPUTの場合のデフォルトです。
OPOS_ER_CLEAR	*ErrorLocus*がOPOS_EL_OUTPUTの場合に使用します。全てのバッファリングされている出力データ（非同期出力を含む。）をクリアします。エラー状態は解除されます。

説明　エラーが検出され、コントロールの**State**がエラー状態に遷移したときに通知します。

正しいアプリケーションシーケンシングが行われるよう、**DataEventEnabled**プロパティがTRUEになるまで、入力エラーイベントは通知されません。

参照　「**ステータス、結果コード、状態モデル**」

OutputCompleteEvent イベント

形式　void OutputCompleteEvent (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　前に開始された非同期出力リクエストが正常に終了したときに通知します。

参照　「出力モデルについて、OutputIDプロパティ」

StatusUpdateEvent イベント

形式　void StatusUpdateEvent (LONG *Status*);

*Status*パラメータは、音声合成デバイスの状態が設定されます。

値	意味
SPCH_SUE_START_SPEAK	音声合成を開始しました
SPCH_SUE_STOP_SPEAK	音声合成を停止しました

第 45 章
ジェスチャーコントロール

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.16	Boolean	R/W	*適用外*
BinaryConversion	1.16	Long	R/W	Open
CapCompareFirmwareVersion	1.16	Boolean	R	Open
CapPowerReporting	1.16	Long	R	Open
CapStatisticsReporting	1.16	Boolean	R	Open
CapUpdateFirmware	1.16	Boolean	R	Open
CapUpdateStatistics	1.16	Boolean	R	Open
CheckHealthText	1.16	String	R	Open
Claimed	1.16	Boolean	R	Open
DataCount	1.16	Long	R	*適用外*
DataEventEnabled	1.16	Boolean	R/W	*適用外*
DeviceEnabled	1.16	Boolean	R/W	Open & Claim
FreezeEvents	1.16	Boolean	R/W	Open
OpenResult	1.16	Long	R	無し
OutputID	1.16	Long	R	Open
PowerNotify	1.16	Long	R/W	Open
PowerState	1.16	Long	R	Open
ResultCode	1.16	Long	R	無し
ResultCodeExtended	1.16	Long	R	Open
State	1.16	Long	R	無し
ControlObjectDescription	1.16	String	R	無し
ControlObjectVersion	1.16	Long	R	無し
ServiceObjectDescription	1.16	String	R	Open
ServiceObjectVersion	1.16	Long	R	Open
DeviceDescription	1.16	String	R	Open
DeviceName	1.16	String	R	Open

専用

CapAssociatedHardTotalsDevice	1.16	String	R	Open
CapMotion	1.16	Boolean	R	Open
CapMotionCreation	1.16	Boolean	R	Open
CapPose	1.16	Boolean	R	Open
CapPoseCreation	1.16	Boolean	R	Open
CapStorage	1.16	Long	R	Open
AutoMode	1.16	String	R/W	Open, Claim, & Enable
AutoModeList	1.16	String	R	Open
JointList	1.16	String	R	Open
MotionList	1.16	String	R	Open
PoseCreationMode	1.16	Boolean	R/W	Open, Claim, & Enable
PoseList	1.16	String	R	Open
Storage	1.16	Long	R/W	Open, Claim, & Enable

メソッド

共通	*版数*	*使用可能条件*
Open	1.16	無し
Close	1.16	Open
ClaimDevice	1.16	Open
ReleaseDevice	1.16	Open & Claim
CheckHealth	1.16	Open, Claim, & Enable
ClearInput	1.16	Open & Claim
ClearInputProperties	1.16	Open & Claim
ClearOutput	1.16	Open & Claim
CompareFirmwareVersion	1.16	Open, Claim, & Enable
DirectIO	1.16	Open
ResetStatistics	1.16	Open, Claim, & Enable
RetrieveStatistics	1.16	Open, Claim, & Enable
UpdateFirmware	1.16	Open, Claim, & Enable
UpdateStatistics	1.16	Open, Claim, & Enable

専用

CreateMotion	1.16	Open, Claim, & Enable
CreatePose	1.16	Open, Claim, & Enable
GetPosition	1.16	Open, Claim, & Enable
SetPosition	1.16	Open, Claim, & Enable
SetSpeed	1.16	Open, Claim, & Enable

StartMotion	1.16	Open, Claim, & Enable
StartPose	1.16	Open, Claim, & Enable
StopControl	1.16	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.16	*適用外*
DirectIOEvent	1.16	Open & Claim
ErrorEvent	1.16	Open, Claim, & Enable
OutputCompleteEvent	1.16	Open, Claim, & Enable
StatusUpdateEvent	1.16	Open, Claim, & Enable
TransitionEvent	1.16	*適用外*

概説

ジェスチャーコントロールのOLEプログラマティクIDは"OPOS.GestureControl"です。

機能

ジェスチャーコントロールは次の機能を持っています。

各種ジョイントの動作を制御します。

各種ジョイントと他デバイスを連動させて、動作を自動制御します。

定義されたポーズやモーションを登録、及び再生します。

モデル

ジョイントコントロールは、非同期的な出力モデルに従います。

アプリケーションは**SetPosition**メソッド、**SetSpeed**メソッド、**StartPose**メソッド、**StartMotion**メソッドを呼び出して出力を開始します。コントロールはプログラムメモリー内のリクエストをバッファリングし、このリクエストの一意な整数の識別子をOutputIDプロパティに設定して、できる限り迅速に戻ります。デバイスがリクエストを正常に終了すると、OPOSがOutputCompleteEventを通知します。このイベントのパラメータは完了したリクエストのOutputIDです。アプリケーションは、デバイスへのデータ送信に成功したことを追跡するために、非同期処理メソッドの呼び出しによって設定されたOutputIDと、OutputCompleteEventのOutputIDプロパティを比較します。

非同期リクエストを実行しているときにエラーが発生すると、ErrorEventが通知されます。アプリケーションのイベントハンドラは、未処理の出力を再試行するかまたは、それをクリアできます。ErrorEventが処理中は、コントロールはErrorの状態です。
（備考：エラーを引き起こす原因が除去されていない場合は、コントロールは直ちにError状態に再びなり、別のErrorEventを通知します。）

非同期出力はFIFOベースで実行されます。

コントロールがバッファリングした出力はすべて（非同期出力をすべて含む）、ClearOutputメソッドを呼び出すことによって削除できます。クリアした出力に対してOutputCompleteEventは通知されません。このメソッドは（可能ならば）処理中の出力も停止します。
また、**StopControl**メソッドを呼び出すことで、出力を個別に削除できます。この場合もOutputCompleteEventは通知されません。

自動制御

ジョイントの自動制御とは、デバイスに接続されたカメラ等と連携し、人の顔の動きに合わせて追跡する等、デバイス側でジョイントを自動制御することを表します。

自動制御機能はデバイスに依存します。可能な自動制御については、AutoModeListプロパティで確認し、**AutoMode**プロパティに値を設定することで有効になります。

ポーズ・モーション

ポーズとは、定義された一つまたは複数のジョイントの位置を設定することを指します。
例えば、手を挙げる動作等です。

モーションとは、ポーズを複数連続させたものを指します。
例えば、手を振る動作等です。

ポーズを実行するには、**StartPose**メソッドにポーズファイル名、またはデバイスで定義されたポーズ名を指定します。
ポーズファイルは後述する**CreatePose**メソッドで作成します。デバイスで定義されたポーズは**PoseList**プロパティを確認します。

モーションを実行するには、**StartMotion**メソッドにモーションファイル名、またはデバイスで定義されたモーション名を指定します。
モーションファイルは後述する**CreateMotion**メソッドで作成します。
デバイスで定義されたモーションは**MotionList**プロパティを確認します。

ポーズファイルを作成するには、まずPoseCreationModeプロパティにTRUEを設定して、ポーズ登録モードを有効にします。ポーズ登録モードを有効にすると、各ジョイントはデフォルトの位置に設定されます。この時、自動制御モードが有効なら、自動制御モードは一時的に無効になります。
その後SetPositionメソッドでジョイントのポジションを、ポーズとして定義したい値に設定し、CreatePoseメソッドを呼び出すことでポーズファイルを作成できます。

モーションファイルは作成したポーズファイルやデバイスで定義されたポーズを指定し、**CreateMotion**メソッドを呼び出すことで作成できます。

作成されたポーズファイルとモーションファイルはエリアに記録されるため、「ハードトータル」デバイスまたはホストファイルシステム、あるいはその両方に保存でき、**CapStorage**プロパティにはデバイスのデータファイルの保存場所機能が表示されます。

デバイスが「ハードトータル」デバイスとホストファイルシステムの両方をサポートしている場合、アプリケーションはそれに応じて**Storage**プロパティを設定して、データファイルの書き込み先を指示する必要があります。

デバイスがポーズファイルとモーションファイルを「ハードトータル」デバイスに書き込める必要がある場合、**CapAssociatedHardTotalsDevice**プロパティは、関連付けられた「ハードトータル」デバイスのオープン名を保持します。

デバイスの共有

ジョイントは下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、デバイスの読み込みを始める前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の詳細な必要条件は、「一覧」を参照してください。

プロパティ

AutoMode プロパティ R/W

形式 BSTR AutoMode;

説明 自動制御モードIDを示します。有効な値は空文字""か、AutoModeListプロパティに記された内の一つです。

このプロパティにAutoModeListプロパティに記された内の一つを設定すると、設定したモードで自動制御モードが有効になります。
空文字""を設定すると、自動制御モードが無効になります。

このプロパティは、Openメソッドにより、空文字""に初期化されます。

戻り値 本プロパティ設定時、次の値がResultCodeプロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	ResultCodeの項目を参照してください。

参照 AutoModeListプロパティ

AutoModeList プロパティ

形式 BSTR AutoModeList;

説明 デバイスがサポートしているジョイント自動制御IDをカンマ区切りのリストで示します。

例えば、カメラと連動して、人の顔をJoint01の関節だけを動かして追跡するモードと、すべての関節を動かして追跡するモードがサポートされている場合は、以下のようになります。
"FaceTrack_Joint01,FaceTrack_ALL"(内容や順番については、デバイスに依存します。)

このプロパティは、Openメソッドにより初期化されます。

参照 AutoModeプロパティ

CapAssociatedHardTotalsDeviceプロパティ

形式　BSTR CapAssociatedHardTotalsDevice;

説明　CapStorageがGCTL_CST_ALLまたはGCTL_CST_HARDTOTALS_ONLYの場合のように、デバイスがそのようなデバイスに書き込むことができる場合、関連するハードトータルデバイスのオープン名を保持します。 CapStorageがGCTL_CST_HOST_ONLYの場合、このプロパティ値は空の文字列である必要があります。

このプロパティは、Openメソッドにより初期化されます。

参照　CapStorageメソッド

CapMotion プロパティ

形式　BOOL CapMotion;

説明　TRUE：モーション機能をサポートしています。
FALSE：モーション機能をサポートしていません。

このプロパティがFALSEなら、StartMotionメソッド、CreateMotionメソッドが利用できません。

このプロパティは、Openメソッドにより初期化されます。

参照　StartMotionメソッド、CreateMotionメソッド

CapMotionCreation プロパティ

形式　BOOL CapMotionCreation;

説明　TRUE：モーション登録機能をサポートしています。
FALSE：モーション登録機能をサポートしていません。

このプロパティがFALSEなら、CreateMotionメソッドが利用できません。

このプロパティは、Openメソッドにより初期化されます。

参照　CreateMotionメソッド

CapPose プロパティ

形式　BOOL CapPose;

説明 TRUE：ポーズ機能をサポートしています。
FALSE：ポーズ機能をサポートしていません。

このプロパティがFALSEなら、**PoseCreationMode**プロパティの変更、**StartPose**メソッド、**CreatePose**メソッドが利用できません。

このプロパティは、**Open**メソッドにより初期化されます。

参照 **PoseCreationMode**プロパティ、**StartPose**メソッド、**CreatePose**メソッド

CapPoseCreation プロパティ

形式 BOOL CapPoseCreation;

説明 TRUE：ポーズ登録機能をサポートしています。
FALSE：ポーズ登録機能をサポートしていません。

このプロパティがFALSEなら、**PoseCreationMode**プロパティの変更、**CreatePose**メソッドが利用できません。

このプロパティは、**Open**メソッドにより初期化されます。

参照 **PoseCreationMode**プロパティ、**CreatePose**メソッド

CapStorageプロパティ

形式 LONG CapStorage;

説明 これは列挙であり、デバイスが記録されたモーションおよび/またはポーズデータファイルをどこに書き込むことができるかをアナウンスします。

次のいずれかの値を保持します。

値	説明
GCTL_CST_HARDTOTALS_ONLY	関連するハードトータルデバイスのみがサポートされます。
GCTL_CST_HOST_ONLY	ホストのファイルシステムのみがサポートされています。
GCTL_CST_ALL	関連するハードトータルデバイスとホストのファイルシステムの両方がサポートされています。

このプロパティは、**Open**メソッドにより初期化されます。

Hard Totalsデバイスがストレージでサポートされている場合、プロパティ値はGCTL_CST_HARDTOTALS_ONLYまたはGCTL_CST_ALLである必要があり、プロパティCapAssociatedHardTotalsDeviceは、関連付けられたHardTotalsデバイスのオープン名を保持します。

参照　**Storage**プロパティ、**CapAssociatedHardTotalsDevice**プロパティ

JointList プロパティ

形式　**BSTR JointList;**

説明　デバイスがサポートしているジョイント情報をカンマ区切りのリストで示します。

各ジョイント情報は以下の情報で構成されており、下記順番にコロン（“:”）で区切って示します。

パラメータ	説明
ジョイントID	ジョイントを識別する、サービス内で一意なIDを示します。
ポジション範囲有無	0ならジョイントはポジション範囲を持ちません、1ならポジション範囲を持ちます。例えば、アーム関節は回転幅の範囲を持ちますが、移動用の車輪は移動量の範囲を持ちません。

例えば、pitch, roll, yawが可能な関節と、車輪による回転、前後移動ができるジョイントがサポートされているデバイスの場合は以下のようになります。
"Joint01_Pitch:1,Joint01_Roll:1,Joint01_Yaw:1,Wheel_Turn:0,Wheel_Move:0"

このプロパティは、**Open**メソッドにより初期化されます。

MotionList プロパティ

形式 BSTR MotionList;

説明 デバイスで定義されているモーションIDをカンマ区切りのリストで示します。

例えば、“bowing, welcoming, clapping,…”

このプロパティは、**Open**メソッドにより初期化されます。

PoseCreationMode プロパティ R/W

形式 BOOL PoseCreationMode;

説明 TRUE：ポーズ登録モードを有効です。
FALSE：ポーズ登録モード無効です。

このプロパティにTRUEを設定すると、ポーズ登録モードが有効になります。FALSEを設定すると、ポーズ登録モードが無効になります。

このプロパティは、Openメソッドの呼出し後にデバイスを最初にイネーブルにしたときにFALSEに初期化されます。

戻り値 本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	制御実行中のため設定に失敗しました(TRUE設定時)。またはこの機能をサポートしていません。
その他	**ResultCode**の項目を参照してください。

参照 **CapPose**プロパティ、**CapPoseCreation**プロパティ

PoseList プロパティ

形式 BSTR PoseList;

説明 デバイスで定義されているポーズIDをカンマ区切りのリストで示します。

例えば、“surprise, bow, think,….”

このプロパティは、**Open**メソッドにより初期化されます。

Storageプロパティ

形式　　LONG Storage;

説明　　これは列挙型であり、デバイスが記録されたモーションおよび/またはポーズデータファイルをどこに書き込むかを定義します。 適切なメソッド呼び出しの前に設定する必要があります。

次のいずれかの値を保持します。

値	説明
GCTL_ST_HARDTOTALS	モーションおよび/またはポーズデータファイルは、関連するハードトータルデバイスに書き込まれます。 プロパティCapAssociatedHardTotalsDeviceは、関連付けられたHardTotalsデバイスのオープン名を保持します。
GCTL_ST_HOST	モーションおよび/またはポーズデータファイルは、ホストのファイルシステムに書き込まれます。
GCTL_ST_HOST_HARDTOTALS	モーションおよび/またはポーズデータファイルは、関連するハードトータルデバイスおよびホストのファイルシステムに書き込まれます。 プロパティCapAssociatedHardTotalsDeviceは、関連付けられたHardTotalsデバイスのオープン名を保持します。

このプロパティは、**CapStorage**プロパティが保持する値に従って**Open**メソッドにより初期化されます。

CapStorageプロパティの値がGCTL_CST_ALLの場合、GCTL_ST_HOST_HARDTOTALSに初期化されます。

参照　　**CapStorage**プロパティ、**CapAssociatedHardTotalsDevice**プロパティ

メソッド

CreateMotion メソッド

形式 LONG CreateMotion(BSTR *FileName*, BSTR *PoseList*);

パラメータ	説明
FileName	モーションを記録するモーションファイル名を指定します。
PoseList	登録するポーズ情報をカンマ区切りのリストで指定します。

説明 登録したポーズを指定して、モーションファイルに記録します。

モーションファイルが記録される場所は、**Storage**プロパティで指定された領域です。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	*FileName*が長すぎるか使用できない文字が含まれています。
OPOS_E_EXISTS	*FileName*がすでに存在します。
その他	**ResultCode**の項目を参照してください。

CreatePose メソッド

形式 LONG CreatePose(BSTR *FileName*, LONG *Time*);

パラメータ	説明
FileName	ポーズを記録するポーズファイル名を指定します。
Time	ポーズポジションに到達するまでの時間を指定します。

説明 各ジョイントのポジションをポーズファイルに記録します。

このメソッドを呼び出す前に、**PoseCreationMode**プロパティにTRUEを設定して、ポーズ登録モードを有効にしている必要があります。

モーションファイルが記録される場所は、**Storage**プロパティで指定された領域です。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	*FileName*が長すぎるか使用できない文字が含まれています、またはPoseCreationModeがFALSEです。
OPOS_E_EXISTS	*FileName*がすでに存在します。
その他	**ResultCode**の項目を参照してください。

参照　**PoseCreationMode**プロパティ

GetPosition メソッド

形式　**LONG GetPosition(BSTR** *JointID*, **LONG** **Position*);

パラメータ	説明
JointID	ジョイントIDを指定します。**JointList**プロパティに記された内の一つを指定します。
Position	*JointID*で指定したジョイントのポジションが格納されます。

説明　*JointID*で指定したポジションを取得し、*Position*に格納します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**JointList**プロパティ

SetPosition メソッド

形式　LONG SetPosition(BSTR *PositionList*, LONG *Time*, BOOL *Absolute*);

パラメータ	説明
PositionList	ポジション情報をカンマ区切りのリストで指定します。
Time	制御完了までの時間を秒単位で指定します。この値が小さすぎる場合は、サービスにより適切な値に変更されます。
Absolute	TRUEなら指定するポジションは絶対値を示します。FALSEなら指定するポジションは相対値を示します。

*PositionList*に指定する各ポジション情報は以下の情報で構成されており、下記順番にコロン（“:”）で区切って示します。

パラメータ	説明
ジョイントID	ジョイントIDを指定します。**JointList**プロパティに記された内の一つを指定します。
ポジション	設定するポジションを指定します。 ポジション範囲有無が1のIDの時、有効な値は-1000から1000の範囲です。 1000は対象ジョイントの正方向の限界値を表し、-1000は負方向の限界値を表します。 *Absolute*が相対値(FALSE)で、ここで指定した値が限界値を超える場合は、サービスにより適切な値に変更されます。 ポジション範囲有無が0のIDの時、値の単位は対象ジョイントにより異なります。

例えば、Joint01のYowを正方向の限界まで、Joint02のPitchを中間まで移動させる場合は以下の様に指定します。
"Joint01_Yaw:100,Joint01:Pitch:0"

説明　*PositionList*で指定した内容でジョイントのポジションを設定し、*Time*で指定した時間に制御が完了するように制御を開始します。

このメソッドは非同期的に実行されます。動作を途中で終了するには、**StopControl**メソッドを呼びます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**JointList**プロパティ、**StopControl**メソッド

SetSpeed メソッド

形式　**LONG SetSpeed(BSTR** *SpeedList*, **LONG** *Time*);

パラメータ	説明
SpeedList	スピード情報をカンマ区切りのリストで指定します。
Time	制御する時間を秒単位で指定します。 OPOS_FOREVER(-1)を指定すると、**StopControl**メソッドを呼ぶまで動作を続けます。

*SpeedList*に指定する各スピード情報は以下の情報で構成されており、下記順番にコロン（“:”）で区切って示します。

パラメータ	説明
ジョイントID	ジョイントIDを指定します。**JointList**プロパティに記された内の一つを指定します。
スピード	設定するスピードを指定します。有効な値は-100から100の範囲です。 100は対象ジョイントの正方向の最高速度を表し、-100は負方向の最高速度を表します。

例えば、WheelのXを正方向の最高速度、WheelのYを負方向の半分の速度で移動させる場合は以下の様に指定します。
"Wheel_X:100,Wheel_Y:-50"

説明　*SpeedList*で指定した内容でジョイントのスピードを設定し、*Time*で指定した時間の間制御を行います。

このメソッドは非同期的に実行されます。動作を途中で終了するには、**StopControl**メソッドを呼びます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**JointList**プロパティ、**StopControl**メソッド

StartMotion メソッド

形式　**LONG StartMotion(BSTR** *FileName***);**

パラメータ	説明
FileName	開始するモーションファイル名を指定します。または**MotionList**プロパティに記されるポーズIDリストの1つを指定します。

説明　*FileName*で指定したモーションファイルまたはデバイスで定義されたモーションを開始します。

モーションファイルが記録される場所は、**Storage**プロパティで指定された領域です。

このメソッドは非同期的に実行されます。モーション制御を途中で終了するには、**StopControl**メソッドを呼びます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が指定されました。
OPOS_E_NOEXIST	*FileName*で指定したモーションファイルが存在しません。
その他	**ResultCode**の項目を参照してください。

参照　**MotionList**プロパティ、**StopControl**メソッド

StartPose メソッド

形式　LONG StartPose(BSTR *FileName*);

パラメータ	説明
FileName	開始するポーズファイル名を指定します。またはPoseListプロパティに記されるポーズIDリストの1つを指定します。

説明　*FileName*で指定したポーズファイルまたはデバイスで定義されたポーズを開始します。

モーションファイルが記録される場所は、Storageプロパティで指定された領域です。

このメソッドは非同期的に実行されます。ポーズ制御を途中で終了するには、StopControlメソッドを呼びます。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	ResultCodeの項目を参照してください。

参照　PoseListプロパティ、StopControlメソッド

StopControl メソッド

形式　LONG StopControl(LONG *OutputID*);

パラメータ	説明
OutputID	終了したい制御に関連付けられたOutputIDプロパティ値を指定します。

説明　*OutputID*に指定した制御を停止します。

戻り値　次の値のいずれかが戻され、ResultCodeプロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。

OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照 **SetPosition**メソッド、**SetSpeed**メソッド、**StartPose**メソッド、**StartMotion**メソッド

イベント

DirectIOEvent イベント

形式　void DirectIOEvent (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。イベント番号とサービスオブジェクトで変化する特定値です。
pString	文字列のポインタ。イベント番号とサービスオブジェクトで変化する特定値です。このデータのフォーマットはBinaryConversionプロパティの値に依存します。詳細はBinaryConversionプロパティを参照してください。

説明　アプリケーションと直接通信するために、サービスオブジェクトが通知します。

このイベントにより、コントロールオブジェクトではサポートされないイベントを、サービスオブジェクトがアプリケーションに提供することが可能となります。

参照　DirectIOメソッド

ErrorEvent イベント

形式　void ErrorEvent (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。

*ResultCode*がOPOS_E_EXTENDEDの場合、*ResultCodeExtended*は次に示す値のいずれかに設定されます。

値	意味
OPOS_ EGCTL_NOROOM	ファイルを作成するのに十分なエリアがありません。

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。

*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	データ送信を再試行します。エラー状態は解除されます。いくつかの入力デバイスで非同期出力デバイスで*ErrorLocus*がOPOS_EL_OUTPUTの場合、エラー状態が解除され、再送信を行うときに使用します。*ErrorLocus*がOPOS_EL_OUTPUTの場合のデフォルトです。
OPOS_ER_CLEAR	*ErrorLocus*がOPOS_EL_OUTPUTの場合に使用します。全てのバッファリングされている出力データ（非同期出力を含む。）をクリアします。エラー状態は解除されます。

説明　エラーが検出され、コントロールの**State**がエラー状態に遷移したときに通知します。

正しいアプリケーションシーケンシングが行われるよう、**DataEventEnabled**プロパティがTRUEになるまで、入力エラーイベントは通知されません。

参照　「**ステータス、結果コード、状態モデル**」

OutputCompleteEvent イベント

形式　**void OutputCompleteEvent (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　前に開始された非同期出力リクエストが正常に終了したときに通知します。

参照　「**出力モデルについて、OutputIDプロパティ**」

StatusUpdateEvent イベント

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

*Status*パラメータは、ジェスチャーコントロールデバイスの状態が設定されます。

値	意味
GCTL_SUE_START_MOTION	モーションを開始しました
GCTL_SUE_STOP_MOTION	モーションを停止しました

第 46 章
デバイスモニタ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.16	Boolean	R/W	Open
BinaryConversion	1.16	Long	R/W	Open
CapCompareFirmwareVersion	1.16	Boolean	R	Open
CapPowerReporting	1.16	Long	R	Open
CapStatisticsReporting	1.16	Boolean	R	Open
CapUpdateFirmware	1.16	Boolean	R	Open
CapUpdateStatistics	1.16	Boolean	R	Open
CheckHealthText	1.16	String	R	Open
Claimed	1.16	Boolean	R	Open
DataCount	1.16	Long	R	Open
DataEventEnabled	1.16	Boolean	R/W	Open
DeviceEnabled	1.16	Boolean	R/W	Open & Claim
FreezeEvents	1.16	Boolean	R/W	Open
OpenResult	1.16	Long	R	無し
OutputID	1.16	Long	R	*適用外*
PowerNotify	1.16	Long	R/W	Open
PowerState	1.16	Long	R	Open
ResultCode	1.16	Long	R	無し
ResultCodeExtended	1.16	Long	R	Open
State	1.16	Long	R	無し
ControlObjectDescription	1.16	String	R	無し
ControlObjectVersion	1.16	Long	R	無し
ServiceObjectDescription	1.16	String	R	Open
ServiceObjectVersion	1.16	Long	R	Open
DeviceDescription	1.16	String	R	Open
DeviceName	1.16	String	R	Open

専用				
DeviceData	1.16	String	R	Open & Claim & Enable
DeviceList	1.16	String	R	Open
MonitoringDeviceList	1.16	String	R	Open & Claim & Enable

メソッド

共通	*版数*	*使用可能条件*
Open	1.16	無し
Close	1.16	Open
ClaimDevice	1.16	Open
ReleaseDevice	1.16	Open & Claim
CheckHealth	1.16	Open, Claim, & Enable
ClearInput	1.16	Open & Claim
ClearInputProperties	1.16	Open & Claim
ClearOutput	1.16	*適用外*
CompareFirmwareVersion	1.16	Open, Claim, & Enable
DirectIO	1.16	Open
ResetStatistics	1.16	Open, Claim, & Enable
RetrieveStatistics	1.16	Open, Claim, & Enable
UpdateFirmware	1.16	Open, Claim, & Enable
UpdateStatistics	1.16	Open, Claim, & Enable

専用		
AddMonitoringDevice	1.16	Open & Claim & Enable
ClearMonitoringDevice	1.16	Open & Claim & Enable
DeleteMonitoringDevice	1.16	Open & Claim & Enable
GetDeviceValue	1.16	Open

イベント

名称	*版数*	*発生条件*
DataEvent	1.16	Open, Claim, & Enable
DirectIOEvent	1.16	Open, Claim
ErrorEvent	1.16	Open, Claim, & Enable
OutputCompleteEvent	1.16	*適用外*
StatusUpdateEvent	1.16	Open, Claim, & Enable
TransitionEvent	1.16	*適用外*

概説

デバイスモニタコントロールのOLEプログラマティクIDは"OPOS.DeviceMonitor"です。

デバイスコントロールは、デバイスに接続された複数のデバイスを監視します。

機能

デバイスモニタコントロールは次の機能を持っています。

各種デバイスが計測した値を取得します。

各種デバイスが計測する値の変化をアプリケーションに通知します。

モデル

デバイスモニタは、イベント駆動入力の一般的な入力モデルに従います。

デバイスモニタはデバイスに接続された複数デバイスが計測する値の監視をサポートします。監視可能なデバイスやその種類・値の単位はDeviceListプロパティにリストで記されます。

コントロールは監視対象に設定したデバイスが計測した値の変化を受信し、指定条件に一致したとき、DataEventを発生させます。

デバイスを監視対象に追加するには、**AddMonitoringDevice**メソッドで監視モードを指定して追加します。監視モードの詳細については、**AddMonitoringDevice**メソッドの説明を参照してください。

AutoDisableプロパティがTRUEならば、**DataEvent**のキューイング時にコントロールは自動的にディセーブルとなります。

DataEventEnabledプロパティがTRUEならば、キューイングした**DataEvent**はアプリケーションに通知されます。コントロールはこのイベントを発生させる直前に、データをプロパティにコピーし、さらなるデータイベントを発生させないために、**DataEventEnabled**プロパティをFALSEに設定します。これにより、アプリケーションが現在の入力の処理や関連するプロパティの処理をしている間、コントロールはその後の入力データをキューイングするようになります。アプリケーションが現在の入力データの処理を終え、次のデータ処理の準備ができたとき、**DataEventEnabled**プロパティをTRUEに設定することにより、再びデータイベントが通知されるようになります。

入力データの読込み中または処理中に、コントロールにおいてエラーが発生した場合、**ErrorEvent**が発行され、**DataEventEnabled**プロパティがTRUEのならば、アプリケーションに通知されます。

DataCountプロパティを読むことでコントロールによってキューイングされたデータイベント数が得られます。

コントロールにキューイングされたすべての入力データは**ClearInput**メソッドを呼ぶことにより削除できます。

DataEventもしくは**ErrorEvent**発生により入力された全てのデータプロパティは、ClearInputPropertiesメソッドを呼ぶ事によりデフォルト値に戻す事が出来ます。

通知されたデータは、**DeviceData**プロパティに格納されます。

デバイスモニタではデバイスの計測値をLONG型の整数値で管理しますが、デバイスによっては小数値を扱うものもあります。その場合は、**DeviceList**プロパティで取得できるデバイス毎の係数で計測値を割ることで実際の値を算出することができます。

StatusUpdateEventでステータスが変更されると、アプリケーションに通知されます。また、対応するすべてのステータス・プロパティもイベント通知前に更新されます。

デバイスの共有

デバイスは下記のように排他的に使用すべきデバイスです。

アプリケーションは、デバイスをイネーブルにする前に排他アクセス権を獲得しなければなりません。

アプリケーションは、デバイスの読み込みを始める前に、デバイスの排他アクセス権を獲得してイネーブルにしなければなりません。

使用上の詳細な必要条件は、「一覧」を参照してください。

プロパティ

DeviceData プロパティ

形式　BSTR DeviceData;

説明　**AddMonitoringDevice**メソッドで登録した条件に一致したデバイスの計測情報が設定されます。

各計測情報は以下の情報で構成されており、下記順番にコロン（“:”）で区切って示します。

パラメータ	説明
デバイスID	対象のデバイスID。
計測値	デバイスの計測値。計測値は整数型で表現され、実際の値に変換するには、**DeviceList**プロパティで取得した係数で計測値を割ります。 例："Device01:365"

このプロパティは、DataEventが通知される直前にコントロールによって設定されます。

DeviceList プロパティ

形式　BSTR DeviceList;

説明　デバイスがサポートしているデバイス情報をカンマ区切りのリストで示します。

各デバイス情報は以下の情報で構成されており、下記順番にコロン（“:”）で区切って示します。

パラメータ	説明
デバイスID	デバイスを識別する、サービス内で一意なIDを示します。
種類	デバイスの種類を示します。例えば、タッチセンサーであれば"TouchSensor"等と表現されます。ただし、この値はサービスに依存します。
単位	各種デバイスが保持する値の単位を示します。例えば、タッチセンサーであれば"on/off"、ジャイロスコープであれば"rad/s"等と表現されます。ただし、この値はサービスに依存します。
係数	各種デバイスが保持する実際の計測値を算出するための係数を示します。DeviceDataプロパティや、GetDeviceValueメソッドで取得できるデバイスの計測値は整数で表現されますが、この値を係数で割ることで実際の値となります。 例：サポートしているデバイスがタッチセンサ1つ、ジャイロスコープが1つあった場合は以下のようになります。 "Touch01:TouchSensor:ON/OFF:1,GyroX:Gyroscope:rad/s:100000,GyroY:Gyroscope:rad/s:100000,GyroZ:Gyroscope:rad/s:100000"

このプロパティは、Openメソッドにより初期化されます。

参照　DeviceDataプロパティ、AddMonitoringDeviceメソッド、GetDeviceValueメソッド

MonitoringDeviceList プロパティ

形式　BSTR MonitoringDeviceList;

説明　登録済みデバイスの監視対象情報をカンマ区切りのリストで示します。

各監視対象情報は以下の情報で構成されており、下記順番にコロン（“:”）で区切って示します。

パラメータ	説明
デバイスID	登録済みのデバイスID。
監視モード	登録済みの監視モード。
境界値	登録済みの境界値。境界値が不要な監視モードの時は0が設定されます。
サブ境界値	登録済みのサブ境界値。サブ境界値が不要な監視モードの時は0が設定されます。
インターバル時間	登録済みのインターバル時間(ミリ秒)。

例えば、以下の様に監視対象を設定している場合、
[監視対象1]
デバイスID=Device01, 監視モード=DMON_MMODE_UPDATE, 境界値=0, サブ境界値=0, インターバル時間=0
[監視対象2]
デバイスID=Device02, 監視モード=DMON_MMODE_STRADDLED, 境界値=365, サブ境界値=0, インターバル時間=500
セットされる値は以下の様になります。
"Device01:0:0:0:0,Device02:1:365:0:500"

このプロパティは、Openメソッドにより初期化されます。また、**AddMonitoringDevice**メソッド、**DeleteMonitoringDevice**メソッド、**ClearMonitoringDevice**メソッドの呼び出しにより更新されます。

参照　**AddMonitoringDevice**メソッド、**DeleteMonitoringDevice**メソッド、**ClearMonitoringDevice**メソッド

メソッド

AddMonitoringDevice メソッド

形式　LONG AddMonitoringDevice(BSTR *DeviceID*, LONG *MonitoringMode*, LONG *Boundary*, LONG *SubBoundary*, LONG *IntervalTime*);

パラメータ	説明
DeviceID	監視対象デバイスのデバイスID。有効な値は**DeviceList**プロパティに記されるデバイスIDリストの1つです。
MonitoringMode	監視する際の監視モードを指定します。
Boundary	監視する境界値を指定します。
SubBoundary	監視するサブ境界値を指定します。この値はBoundary未満の値である必要があります。
IntervalTime	イベント発生後、次の監視を始めるまでの間隔をミリ秒で指定します。

*MonitoringMode*に指定する監視モードは以下の通りです。

値	説明
DMON_MMODE_UPDATE	対象デバイスの計測値が更新される毎にイベントを通知します。このモードに設定した場合、引数*Boundary*と*SubBoundary*の値は無視されます。
DMON_MMODE_STRADDLED	対象デバイスの計測値が、引数*Boundary*の値を跨いだ時にイベントを通知します。その他、計測値が*Boundary*の値と一致した時、一致した状態から変更した時にもイベントを通知します。このモードに設定した場合、引数*SubBoundary*の値は無視されます。

DMON_MMODE_HIGH
対象デバイスの計測値が、引数*Boundary*の値以上となった時にイベントを通知します。計測値が更新され、再度*Boundary*の値以上となっていた場合でも、都度イベントを通知します。このモードに設定した場合、引数*SubBoundary*の値は無視されます。

DMON_MMODE_LOW
対象デバイスの計測値が、引数*Boundary*の値以下となった時にイベントを通知します。計測値が更新され、再度*Boundary*の値以下となっていた場合でも、都度イベントを通知します。このモードに設定した場合、引数*SubBoundary*の値は無視されます。

DMON_MMODE_WITHIN
対象デバイスの計測値が、引数*Boundary*と*SubBoundary*で指定した値の範囲内である間イベントを通知します。計測値が更新され、その値が再度範囲内となっていた場合でも、都度イベントを通知します。

DMON_MMODE_OUTSIDE
対象デバイスの計測値が、引数*Boundary*と*SubBoundary*で指定した値の範囲外である間イベントを通知します。計測値が更新され、その値が再度範囲外となっていた場合でも、都度イベントを通知します。

DMON_MMODE_POLLING
*IntervalTime*で指定した間隔で対象デバイスの計測値を通知します。このモードに設定した場合、引数*Boundary*と*SubBaundary*の値は無視されます。

説明　*DeviceID*で指定したデバイスを監視対象に加えます。

*MonitoringMode*に監視モードを指定しますが、デバイスによってはサポートしていない監視モードが存在します。その場合、戻り値にはOPOS_E_ILLEGALが戻ります。

このメソッドにより追加したDeviceは**MonitoringDeviceList**プロパティのリストに追加されます。

すでに監視対象としているデバイスが指定された場合は、新しい条件に変更されます。

追加したデバイスを監視対象から除外するには、**DeleteMonitoringDevice**メソッドか、**ClearMonitoringDevice**メソッドを呼び出します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常に開始されました。
OPOS_E_ILLEGAL	無効な値が渡されました。または、サポートしていない監視モードが指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**DeviceList**プロパティ、**MonitoringDeviceList**プロパティ、**DeleteMonitoringDevice**メソッド、**ClearMonitoringDevice**メソッド

ClearMonitoringDevice メソッド

形式　LONG ClearMonitoringDevice();

説明　すべてのデバイスを監視対象から除外します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
その他	**ResultCode**の項目を参照してください。

参照　**AddMonitoringDevice**メソッド

DeleteMonitoringDevice メソッド

形式　LONG DeleteMonitoringDevice(BSTR *DeviceID*);

パラメータ	説明
DeviceID	監視対象から除外するデバイスのデバイスIDを指定します。

説明　*DeviceID*で指定したデバイスを監視対象から除外します。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が渡されました。
その他	**ResultCode**の項目を参照してください。

参照　**AddMonitoringDevice**メソッド

GetDeviceValue メソッド

形式　**LONG GetDeviceValue(BSTR** *DeviceID*, **LONG** **pValue*);

パラメータ	説明
DeviceID	計測値を取得するデバイスのデバイスIDを指定します。**DeviceList**プロパティに記されるデバイスIDリストの1つを指定します。
pValue	デバイスから取得した計測値。

説明　*DeviceID*で指定したデバイスの計測値を取得します。取得した値は*pValue*に格納されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	意味
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が渡されました。
その他	**ResultCode**の項目を参照してください。

参照　**DeviceList**プロパティ

イベント

DataEvent イベント

形式　　void DataEvent (LONG *Status*);

パラメータ	説明
Status	0が設定されます。

説明　　デバイスモニターデバイスからのデータを読み取ることができるようになると、アプリケーションに通知します。

このイベントが配信される前に、個々の認識情報は、addMonitoringDeviceメソッドによって示される領域にエンキューされます。

参照　　「AddMonitoringDeviceメソッド」

DirectIOEvent イベント

形式　　void DirectIOEvent (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。イベント番号とサービスオブジェクトで変化する特定値です。
pString	文字列のポインタ。イベント番号とサービスオブジェクトで変化する特定値です。

説明　　このイベントにより、コントロールオブジェクトではサポートされないイベントを、サービスオブジェクトがアプリケーションに提供することが可能となります。

このイベントは、ベンダー固有の機能にのみ使用されます。
このイベントを使用すると、他のベンダーのデバイスで使用するアプリケーションプログラムフォームが制限される場合があります。

参照　　「イベント、DirectIOメソッド」

ErrorEvent イベント

形式　**void ErrorEvent (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**ResultCode**の項目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**ResultCodeExtended**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_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が、すでにバッファリングされたデータは利用できます。

*pErrorResponse*パラメータが示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_CLEAR	バッファリングされている入力データをクリアし、エラー状態は解除されます。 これは、*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、**DataEventEnabled**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、**DataEventEnabled**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　このイベントはエラーを検出した時に通知され、デバイスの**State**プロパティはエラー状態になります。

正しいアプリケーションシーケンシングが行われるよう、**DataEventEnabled**プロパティがTRUEになるまで、入力エラーイベントは通知されません。

DataEventとは異なり、デバイスは次の**DataEvent**や**ErrorEvents**を無効にせず、**DataEventEnabled**プロパティ値をtrueのままにしておきます。　後続の入力イベントを一定期間無効にする必要がある場合、アプリケーションはイベントハンドラ内で**DataEventEnabled**をfalseに設定することができます。

参照　「**ステータス、結果コード、状態モデル**」

StatusUpdateEvent イベント

形式　　void StatusUpdateEvent (LONG *Status*);

パラメータ	説明
Status	デバイスモニタのステータス変化を示します。

値	意味
DMON_SUE_START_MONITORING	デバイス監視が開始した時に通知されます。
DMON_SUE_STOP_MONITORING	デバイス監視が終了した時に通知されます。

説明　　デバイスの状態、または電源状態が変化した時に通知されます。

参照　　「イベント」

第 47 章
グラフィックディスプレイ

一覧

プロパティ

共通	*版数*	*型*	*アクセス*	*使用可能条件*
AutoDisable	1.16	Boolean	R/W	*適用外*
BinaryConversion	1.16	Long	R/W	Open
CapCompareFirmwareVersion	1.16	Boolean	R	Open
CapPowerReporting	1.16	Long	R	Open
CapStatisticsReporting	1.16	Boolean	R	Open
CapUpdateFirmware	1.16	Boolean	R	Open
CapUpdateStatistics	1.16	Boolean	R	Open
CheckHealthText	1.16	String	R	Open
Claimed	1.16	Boolean	R	Open
DataCount	1.16	Long	R	*適用外*
DataEventEnabled	1.16	Boolean	R/W	*適用外*
DeviceEnabled	1.16	Boolean	R/W	Open & Claim
FreezeEvents	1.16	Boolean	R/W	Open
OutputID	1.16	Long	R	Open
PowerNotify	1.16	Long	R/W	Open
PowerState	1.16	Long	R	Open
ResultCode	1.16	Long	R	無し
ResultCodeExtended	1.16	Long	R	Open
State	1.16	Long	R	無し
ControlObjectDescription	1.16	String	R	無し
ControlObjectVersion	1.16	Long	R	無し
ServiceObjectDescription	1.16	String	R	Open
ServiceObjectVersion	1.16	Long	R	Open
DeviceDescription	1.16	String	R	Open
DeviceName	1.16	String	R	Open

専用

CapAssociatedHardTotalsDevice	1.16	String	R	Open
CapBrightness	1.16	Boolean	R	Open
CapImageType	1.16	Boolean	R	Open
CapStorage	1.16	Long	R	Open
CapURLBack	1.16	Boolean	R	Open
CapURLForward	1.16	Boolean	R	Open
CapVideoType	1.16	Boolean	R	Open
CapVolume	1.16	Boolean	R	Open
Brightness	1.16	Long	R/W	Open, Claim & Enable
DisplayMode	1.16	Long	R/W	Open, Claim & Enable
ImageType	1.16	String	R/W	Open, Claim & Enable
ImageTypeList	1.16	String	R	Open
LoadStatus	1.16	Long	R	Open
Storage	1.16	Long	R/W	Open, Claim & Enable
URL	1.16	String	R	Open
VideoType	1.16	String	R/W	Open, Claim & Enable
VideoTypeList	1.16	String	R	Open
Volume	1.16	Long	R/W	Open, Claim & Enable

メソッド

共通	*版数*	*使用可能条件*
Open	1.16	無し
Close	1.16	Open
ClaimDevice	1.16	Open
ReleaseDevice	1.16	Open & Claim
CheckHealth	1.16	Open, Claim & Enable
ClearInput	1.16	Open & Claim
ClearInputProperties	1.16	Open & Claim
ClearOutput	1.16	Open & Claim
CompareFirmwareVersion	1.16	Open, Claim, & Enable
DirectIO	1.16	Open
ResetStatistics	1.16	Open, Claim, & Enable
RetrieveStatistics	1.16	Open, Claim, & Enable
UpdateFirmware	1.16	Open, Claim, & Enable
UpdateStatistics	1.16	Open, Claim, & Enable

専用

CancelURLLoading	1.16	Open, Claim, & Enable

goURLBack()	1.16	Open, Claim, & Enable
goURLForward	1.16	Open, Claim, & Enable
LoadImage	1.16	Open, Claim, & Enable
LoadURL	1.16	Open, Claim, & Enable
PlayVideo	1.16	Open, Claim, & Enable
StopVideo	1.16	Open, Claim, & Enable
UpdateURLPage()	1.16	Open, Claim, & Enable

イベント

名称	*版数*	*発生条件*
DataEvent	1.16	*適用外*
DirectIOEvent	1.16	Open, Claim
ErrorEvent	1.16	Open, Claim, & Enable
OutputCompleteEvent	1.16	Open, Claim, & Enable
StatusUpdateEvent	1.16	Open, Claim, & Enable
TransitionEvent	1.16	*適用外*

概説

グラフィックディスプレイコントロールのOLEプログラマティックIDは"OPOS.GraphicDisplay"です。

機能

グラフィックディスプレイコントロールは次の機能を持っています。

- 指定した画像を表示します。
- 指定した動画を再生します。
- 指定した Web ページを表示します。
- Web ページの読み込み状態の変化をアプリケーションに通知します。

モデル

グラフィックスコントロールには下記表示モードがあり、表示モードによってモデルが異なります。

- 画像表示モード
- 動画表示モード
- Web 表示モード

アプリケーションは**DisplayMode**プロパティの値を変更することで表示モードを切り替えることができます。

画像表示モード

グラフィックスコントロールの画像表示モードは、以下の通りです。

アプリケーションは**LoadImage**メソッドを呼び出して画像を表示します。**ImageTypeList**プロパティにはデバイスで表示可能な画像ファイルが記されています。**LoadImage**メソッドで表示する画像ファイルは「ハードトータル」サービスが管理する領域に配置する必要があるため、アプリケーションは「ハードトータル」サービスもサポートする必要があります。

CapImageTypeプロパティがTRUEの場合、このモードを開始する前に、**ImageTypeList**プロパティ内にリストされた値から、適切な画像ファイル種別の値を**ImageType**に設定する必要があります。次にアプリケーションは**LoadImage**を呼び出すことで画像を表示できます。

ステータスが画像読み込みの開始(GDSP_SUE_START_IMAGE_LOAD)、及び読み込み完了(GDSP_SUE_END_IMAGE_LOAD)に変化するタイミングで**StatusUpdateEvent**が発生します。

LoadImageメソッドで表示される画像ファイルが関連する「ハードトータル」サービスデバイスによって管理される場合、アプリケーションは「ハードトータル」サービスをサポートする必要があります。**CapStorage**がGDSP_CST_ALLまたはGDSP_CST_HARDTOTALS_ONLYのどちらかの場合、関連するハードトータルデバイスに保存が可能であり、ストレージデバイスのオープン名は**CapAssociatedHardTotalsDevice**プロパティに保持されます。

デバイスがハードトータルデバイスとホストファイルシステムの両方をサポートしている場合、アプリケーションは**Storage**プロパティに画像データファイルを書き込む場所を指定しなければなりません。

動画表示モード

グラフィックスコントロールの動画表示モードは、非同期の出力デバイスの一般的な動作モデルに従います。

グラフィックコントロールの動画表示モードは、以下の通りです。

CapVideoTypeプロパティがTRUEの場合、このモードを開始する前に、**VideoTypeList**プロパティ内にリストされた値から、適切な動画ファイル種別の値を**VideoType**に設定する必要があります。

次にアプリケーションは**PlayVideo**を呼び出すことで動画を表示できます。また、**StopVideo**メソッドを呼びだすことで再生中の動画を停止します。

ステータスが動画の再生開始(GDSP_SUE_START_PLAY_VIDEO)、及び動画の停止(GDSP_SUE_STOP_PLAY_VIDEO)に変化するタイミングで**StatusUpdateEvent**が発生します。

デバイスは必要に応じてメソッドのパラメーターのエラー状態をただちに検証します。検証が成功した場合、デバイスは以下を行います。

物理デバイスが要求を受信し、処理できるようになるとすぐに物理デバイスに配信するためにプログラムメモリ内のリクエストをバッファリングします。
このリクエストの一意な整数の識別子を **OutputID** プロパティに設定します。
できる限り迅速に処理を戻します。

デバイスがリクエストを正常に完了させたとき、アプリケーションに通知するため**OutputCompleteEvent**が通知されます。

このイベントは完了したリクエストのOutputIDを含みます。

アプリケーションは、デバイスに対するデータ送信に成功したことを追跡するために、非同期処理メソッドの呼び出しによって設定された**OutputID**と、**OutputCompleteEvent**の**OutputID**プロパティを比較する必要があります。

非同期リクエスト処理中にエラーが発生した場合、**ErrorEvent**は既に通知済のイベント(**OutputCompleteEvent**を含む)の後にアプリケーションに通知されます。イベントがアプリケーションに通知されるまで、非同期の出力は発生しません。応答がOPOS_ER_CLEARである場合、未処理の非同期処理はクリアされます。

応答がOPOS_ER_RETRYである場合、出力は再試行されます。

(備考：エラー検出した時に複数の出力が処理中の場合、サービスはこれらすべての出力に対してリトライする必要があります。)

非同期出力はFIFOベースで実行されます。もし、デバイスが同時再生をサポートしている場合、要求は同時に実行されます。

アプリケーションが**ClearOutput**メソッドを呼び出す等によって、リクエストが完了前に終了した場合、**OutputCompleteEvent**は通知されません。

StopVideoメソッドを呼び出して、出力を個別に削除することもできます。この場合も**OutputCompleteEvent**は通知されません。

デバイスで再生可能な動画ファイルは**VideoTypeList**プロパティにリストされます。**PlayVideo**メソッドを使用して再生する動画ファイルは関連付けられたハードトータルサービスデバイスによって管理された領域に配置する必要があります。

デバイスがハードトータルデバイスとホストファイルシステムの両方をサポートしている場合、アプリケーションは動画データファイルを書き込む場所を指定するために、**Storage**プロパティを設定する必要があります。

グラフィックディスプレイの動画表示モードは一般的な非同期出力モデルに従います。動画表示中に電源ステータス、あるいはデバイスのステータスが変化した場合、**StatusUpdateEvent**を通知します。

Web表示モード

グラフィックスコントロールのWeb表示モードは以下の通りです。

アプリケーションはWebページを表示する**LoadURL**メソッドを呼び出します。

Webページの読み込みの開始
(GDSP_SUE_START_LOAD_WEBPAGE)、停止
(GDSP_SUE_FINISH_LOAD_WEBPAGE)、及びキャンセル
(GDSP_SUE_CANCEL_LOAD_WEBPAGE)のタイミングで
StatusUpdateEventが発生します。アプリケーションはWEBページの読み込み状態を検出することができます。

LoadURLメソッドが呼び出されると、**LoadStatus**プロパティに読み込み状態が格納され、**URL**プロパティにURLの情報が格納されます。

読み込み処理中に**CancelURLLoading**メソッドが呼び出された場合、現在アクセス中のURLの情報が**URL**プロパティに格納されます。

WEB表示モードは一般的な非同期出力モデルに従います。

デバイスの共有

グラフィックディスプレイデバイスは次のように排他的に使用されます。

- アプリケーションはデバイスを有効にする前に排他権を取得する必要があります。
- アプリケーションは、プロパティやメソッド呼び出しの前にデバイスの排他権を取得し、有効にしなければなりません。

正確な使用前提条件については、「一覧」を参照してください。

プロパティ

CapAssociatedHardTotalsDevice プロパティ

形式　BSTR CapAssociatedHardTotalsDevice;

説明　CapStorageプロパティがGDSP_CST_ALL、またはGDSP_CST_HARDTOTALS_ONLYでハードトータルデバイスに書き込みが可能な場合、関連付けられたハードトータルデバイスのオープン名を保持します。CapStorageプロパティがGDSP_CST_HOST_ONLYの場合、このプロパティは空の文字列である必要があります。

このプロパティは、Openメソッドにより初期化されます。

参照　CapStorageプロパティ

CapBrightness プロパティ

形式　BOOL CapBrightness;

説明　TRUE：画面の明るさの変更が可能です。
FALSE：画面の明るさの変更はできません。

このプロパティは、Openメソッドにより初期化されます。

参照　Brightnessプロパティ

CapImageType プロパティ

形式　BOOL CapImageType;

説明　TRUE：対象デバイスで使用する画像種別ファイルをImageTypeプロパティの値として指定します。
FALSE：それ以外の場合

このプロパティは、Openメソッドにより初期化されます。

参照　ImageTypeプロパティ、ImageTypeListプロパティ

CapStorage プロパティ

形式　LONG CapStorage;

説明　デバイスがイメージデータファイルを書き込むことが可能な場所を示す列挙体です。次のいずれかの値を保持します。

値	説明
GDSP_CST_HARDTOTALS_ONLY	関連するハードトータルデバイスのみをサポートします。
GDSP_CST_HOST_ONLY	ホストファイルシステムのみをサポートします。
GDSP_CST_ALL	関連するハードトータルデバイスとホストファイルシステムの両方をサポートします。

このプロパティは、**Open**メソッドにより初期化されます。

デバイスがハードトータルデバイスをサポートする場合、**Storage**プロパティの値はGDSP_CST_HARDTOTALS_ONLY、またはGDSP_CST_ALLにする必要があり、**CapAssociatedHardTotalsDevice**プロパティは関連付けられたハードトータルデバイスのオープン名を保持します。

参照　**Storage**プロパティ、**CapAssociatedHardTotalsDevice**プロパティ

CapURLBack プロパティ

形式　BOOL CapURLBack;

説明　TRUE：閲覧履歴に前のページが存在します。アプリケーションはGoURLBackメソッドで前のページに戻ることができます。
FALSE：閲覧履歴に前のページが存在しません。

このプロパティは、**Open**メソッドによりFALSEで初期化されます。また、Webページの読み込み状態が変わることにより、デバイスコントロールによって設定されます。

参照　**GoURLBack**メソッド

CapURLForward プロパティ

型式　BOOL CapURLForward;

説明　TRUE：閲覧履歴に次ページが存在します。**GoURLForward**メソッドで次ページに進むことができます。
FALSE：閲覧履歴に次ページが存在しません。

このプロパティは、**Open**メソッドによりFALSEで初期化されます。また、Webページの読み込み状態が変わることにより、デバイスコントロールによって設定されます。

参照　**GoURLForward**メソッド

CapVideoType プロパティ

形式　BOOL CapVideoType;

説明　TRUE：**VideoType**プロパティの値が対象のグラフィックディスプレイデバイスで使用可能な動画形式の値を示します。
FALSE：それ以外の場合

このプロパティは、**Open**メソッドにより初期化されます。

参照　**VideoType**プロパティ、**VideoTypeList**プロパティ

CapVolume プロパティ

形式　BOOL CapVolume;

説明　TRUE：アプリケーションは動画の音量の変更が可能です。
FALSE：アプリケーションは動画の音量の変更ができません。

このプロパティは、**Open**メソッドにより初期化されます。

参照　**Volume**プロパティ

Brightness プロパティ R/W

形式　LONG Brightness;

説明　画面明るさを示します。有効な値は0から100の範囲です。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値　本プロパティ設定時、次の値がResultCodeプロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**CapBrightness**プロパティ

DisplayMode プロパティ R/W

形式　LONG DisplayMode;

説明　画像、及び動画の表示モードを示します。

サポートしている表示モードは以下の通りです。

値	説明
GDSP_DMODE_HIDDEN	画面を非表示にします。
GDSP_DMODE_IMAGE_FIT	画像を表示するモードです。 表示される画像は、アスペクトを維持し、画面内に丁度入るサイズに拡大・縮小されます。
GDSP_DMODE_IMAGE_FILL	画像を表示するモードです。 表示される画像は、アスペクトを維持し、画面全体を覆うサイズに拡大・縮小されます。
GDSP_DMODE_IMAGE_CENTER	画像を表示するモードです。 表示される画像は、サイズを変更せずに画面中央に表示されます。
GDSP_DMODE_VIDEO_NORMAL	動画を表示するモードです。 表示される動画は、サイズ変更せずに画面中央に表示されます。
GDSP_DMODE_VIDEO_FULL	動画を表示するモードです。 表示される動画は、フルスクリーンで表示されます。
GDSP_DMODE_WEB	Web画面を表示します。

画像、動画、Web表示中に別のモードや画面を非表示にすると、表示内容はすべてクリアされます。動画再生中は動画が停止されます。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。

OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照 **LoadImage**メソッド、**PlayVideo**メソッド

ImageType プロパティ R/W

形式 **BSTR ImageType;**

説明 **CapImageType**プロパティがTRUEの場合、デバイスでサポートされている画像形式を示します。例えば、デバイスがBMP形式をサポートしている場合、このプロパティは"BMP"に設定する必要があります。このプロパティは**LoadImage**メソッドを実行する前に設定する必要があります。対応するすべてのファイル形式は**ImageTypeList**プロパティにリストされます。(表記内容はデバイスによって異なる場合があります。)

このプロパティは、**Open**メソッドにより初期化されます。

戻り値 本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照 **CapImageList**プロパティ、**ImageTypeList**プロパティ、**LoadImage**メソッド

ImageTypeList プロパティ

形式　BSTR ImageTypeList;

説明　デバイスでサポートされている画像ファイル形式をカンマ区切りのリストで保持します。例えば、BMP形式とJPEG形式をサポートしている場合このプロパティは”BMP,JPEG”に設定する必要があります。**CapImageType**プロパティがTRUEの場合、このプロパティの画像形式のうち一つを**LoadImage**メソッドを実行する前に**ImageType**プロパティに設定する必要があります。(表記内容はデバイスによって異なる場合があります。)

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CapImageType**プロパティ、**ImageType**プロパティ、**LoadImage**メソッド

LoadStatus プロパティ

形式　LONG LoadStatus;

説明　Webページの読み込み状態が保持されます。

以下の値が設定されます。

値	説明
GDSP_LSTATUS_START	Webページの読み込みを開始します。
GDSP_LSTATUS_FINISH	Webページの読み込みを完了しました。
GDSP_LSTATUS_CANCEL	Webページの読み込みをキャンセルしました。

本プロパティは、**StatusUpdateEvent**がアプリケーションに通知される直前に設定されます。

Storage プロパティ R/W

形式 LONG Storage;

説明 列挙体であり、デバイスが記録された画像ファイルを書き込む場所を指定します。メソッドを呼び出し前に設定する必要があります。
このプロパティには以下のいずれかの値を設定します。

値	説明
GDSP_ST_HARDTOTALS	画像ファイルは関連付けられたハードトータルデバイスに書き込まれます。
GDSP_ST_HOST	画像ファイルはホストファイルシステムに保存されます。
GDSP_ST_HOST_HARDTOTALS	画像ファイルは関連付けられたハードトータルデバイスとホストファイルシステムに書き込まれます。

CapAssociatedHardTotalsDeviceプロパティは関連付けられたハードトータルデバイスのオープン名を保持します。

CapStorageプロパティの値がGDSP_CST_ALLの場合、このプロパティはGDSP_ST_HOST_HARDTOTALSに設定されます。

戻り値 本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定された、もしくは既に記録中です。
その他	**ResultCode**の項目を参照してください。

参照 **CapStorage**プロパティ、**CapAssociatedHardTotalsDevice**メソッド

URL プロパティ

形式　BSTR URL;

説明　**LoadStatus**プロパティがGDSP_LSTATUS_STARTの時、ロード開始するWebページのURLが設定されます。
LoadStatusプロパティがGDSP_LSTATUS_FINISHの時、ロード完了したWebページのURLが設定されます。
LoadStatusプロパティがGDSP_LSTATUS_CANCELの時、ロード中断したWebページのURLが設定されます。

本プロパティは、**StatusUpdateEvent**が通知される直前にコントロールによって設定されます。

参照　**LoadStatus**プロパティ

VideoType プロパティ R/W

形式　BSTR VideoType;

説明　**CapVideoType**プロパティがTRUEの場合、デバイスで再生可能な動画形式値を示します。例えば、AVI MJPG型式をサポートする場合、このプロパティは“AVI MJPG”に設定する必要があります。このプロパティは**PlayVideo**メソッドを実行する前に設定する必要があります。再生可能なすべての動画形式は**ImageTypeList**プロパティにリストされます。(表記内容はデバイスによって異なる場合があります。)

このプロパティは、**Open**メソッドにより初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**CapVideoType**プロパティ、**VideoTypeList**プロパティ、**PlayVideo**メソッド

VideoTypeList プロパティ

形式　BSTR VideoTypeList;

説明　デバイスでサポートされている動画ファイル形式をカンマ区切りのリストで保持します。例えば、デバイスがAVI_IYUV形式とAVI_MJPG形式をサポートしている場合、このプロパティは" AVI_IYUV, AVI_MJPG"に設定する必要があります。
CapVideoTypeプロパティがTRUEの場合、このプロパティの動画形式のうち一つを**PlayVideo**メソッドを実行する前に**VideoType**プロパティに
設定する必要があります。(表記内容はデバイスによって異なる場合があります。)

このプロパティは、**Open**メソッドにより初期化されます。

参照　**CapVideoType**プロパティ、**VideoType**プロパティ、**PlayVideo**メソッド

Volume プロパティ R/W

形式　LONG Volume;

説明　動画再生時の音量を示します。有効な値は0から100の範囲です。

このプロパティは、**Open**メソッドにより初期化されます。

戻り値　本プロパティ設定時、次の値が**ResultCode**プロパティに格納されます。

値	説明
OPOS_SUCCESS	プロパティは正常に設定されました。
OPOS_E_ILLEGAL	無効な値が指定されました。
その他	**ResultCode**の項目を参照してください。

参照　**CapVolume**プロパティ、**PlayVideo**メソッド

メソッド

CancelURLLoading メソッド

形式　LONG CancelURLLoading();

説明　Webページの読み込みを中断します。

このメソッドは非同期的に実行されます。読み込み状態は**StatusUpdateEvent**、**OutputCompleteEvent**または**ErrorEvent**で通知されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	読み込み中ではありません。
その他	**ResultCode**の項目を参照してください。

GoURLBack メソッド

形式　LONG GoURLBack();

説明　閲覧履歴の前ページに戻ります。

このメソッドは非同期的に実行されます。読み込み状態は**StatusUpdateEvent**、**OutputCompleteEvent**または**ErrorEvent**で通知されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	閲覧履歴に前ページが存在しません。
その他	**ResultCode**の項目を参照してください。

参照　**CapURLBack**プロパティ

GoURLForward メソッド

形式 LONG GoForward();

説明 閲覧履歴の次ページに進みます。

このメソッドは非同期的に実行されます。読み込み状態は**StatusUpdateEvent**、**OutputCompleteEvent**または**ErrorEvent**で通知されます。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	閲覧履歴に次ページが存在しません。
その他	**ResultCode**の項目を参照してください。

参照 **CapURLForward**プロパティ

LoadImage メソッド

形式　LONG LoadImage(BSTR *FileName*);

パラメータ	説明
FileName	読み込む画像ファイル名を指定します。

説明　指定した画像を読み込みます。

DisplayModeプロパティの値が、GDSP_DMODE_IMAGE_FIT、GDSP_DMODE_IMAGE_FILL、GDSP_DMODE_IMAGE_CENTERに設定されていない場合、このメソッドは失敗します。

対象の画像ファイルは**Storage**プロパティに設定された領域に配置されます。

このメソッドは非同期的に実行されます。画像の読み込み状態は**StatusUpdateEvent**、**OutputCompleteEvent**または**ErrorEvent**で通知されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が渡されました。またはサポートしていない画像ファイルが指定されました。
OPOS_E_NOEXIST	*FileName*で指定したファイルが存在しません。
その他	**ResultCode**の項目を参照してください。

参照　**DisplayMode**プロパティ

LoadURL メソッド

形式 LONG LoadURL(BSTR *URL*);

パラメータ	説明
URL	読み込むWebページのURLを指定します。

説明 指定したURLのWebページを読み込みます。

このメソッドは非同期的に実行されます。読み込み状態は**StatusUpdateEvent**、**OutputCompleteEvent**または**ErrorEvent**で通知されます。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が渡されました。
その他	**ResultCode**の項目を参照してください。

PlayVideo メソッド

形式　LONG PlayVideo(BSTR *FileName*, BOOL *Loop*);

パラメータ	説明
FileName	再生する動画のファイル名を指定します。
Loop	TRUEならばループ再生し、FALSEならループ再生しません。

説明　**VideoType**プロパティで設定した動画形式のファイルを再生します。**CapVideoType**プロパティがTRUEの場合、全ての動画ファイルの値は**VideoTypeList**プロパティに列挙されているファイル形式です。**DisplayMode**プロパティの値が、GDSP_DMODE_VIDEO_NORMAL、またはGDSP_DMODE_VIDEO_FULLに設定されていない場合、このメソッドは失敗します。

このメソッドは非同期的に実行されます。動画再生を途中で終了するには、**StopVideo**メソッドを呼びます。

対象の動画ファイルは**Storage**プロパティに設定された領域に配置されます。

動画の再生状況は**StatusUpdateEvent**にて通知されます。

このメソッドは非同期的に実行されます。動画の再生状態は**StatusUpdateEvent**、**OutputCompleteEvent**または**ErrorEvent**で通知されます。

戻り値　次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	無効な値が渡されました。またはサポートしていない動画ファイルが指定されました。
OPOS_E_NOEXIST	*FileName*で指定したファイルが存在しません。
その他	**ResultCode**の項目を参照してください。

参照　**DisplayMode**プロパティ

StopVideo メソッド

形式 LONG StopVideo();

説明 再生中の動画を停止します。

このメソッドは非同期的に実行されます。動画の再生状態は**StatusUpdateEvent**、**OutputCompleteEvent**または**ErrorEvent**で通知されます。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	動画は再生されていません。
その他	**ResultCode**の項目を参照してください。

参照 **PlayVideo**メソッド

UpdateURLPage メソッド

形式 LONG UpdateURLPage();

説明 現在のWebページを再読み込みします。

このメソッドは非同期的に実行されます。読み込み状態は**StatusUpdateEvent**、**OutputCompleteEvent**または**ErrorEvent**で通知されます。

戻り値 次の値のいずれかが戻され、**ResultCode**プロパティにもその値が格納されます。

値	説明
OPOS_SUCCESS	メソッドは正常終了しました。
OPOS_E_ILLEGAL	読み込み中です。
その他	**ResultCode**の項目を参照してください。

イベント

DirectIOEvent イベント

形式　**void DirectIOEvent (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。イベント番号とサービスオブジェクトで変化する特定値です。
pString	文字列のポインタ。イベント番号とサービスオブジェクトで変化する特定値です。

説明　このイベントにより、コントロールオブジェクトではサポートされないイベントを、サービスオブジェクトがアプリケーションに提供することが可能となります。

参照　「イベント、DirectIOメソッド」

ErrorEvent イベント

形式　**void ErrorEvent (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**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**ResultCodeExtended**の項を参照してください。
ErrorLocus	エラーの位置です。EL_OUTPUTが指定されている場合は、非同期処理中にエラーが発生したことを示します。
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。

もし、*ResultCode*がOPOS_E_EXTENDEDであれば*ResultCodeExtended*は次に示す値のいずれかに設定されます。

値	説明
OPOS_EGDSP_NOROOM	画像データファイルの対象デバイスの保存容量が不足しています。

ErrorLocus パラメータは次の値です。

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。

アプリケーションエラーハンドラーは*pErrorResponse*に対して以下のいずれかに設定できます。

値	意味
OPOS_ER_RETRY	非同期処理を再試行します。エラー状態は解除されます。これがデフォルトの応答です。
OPOS_ER_CLEAR	全てのバッファリングされている出力データ（非同期出力を含む）をクリアします(ClearOutputメソッドを呼び出した場合と同等です)。エラー状態は解除されます。

説明　エラーを検知し、エラーが状態に遷移したことを通知します。

参照　「**ステータス、結果コード、状態モデル**」、「出力モデルについて」

OutputCompleteEvent イベント

形式　void OutputCompleteEvent (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　前に開始された非同期出力リクエストが正常に終了したときに通知します。

参照　「出力モデルについて、OutputIDプロパティ」

StatusUpdateEvent イベント

形式　void StatusUpdateEvent (LONG *Status*);

パラメータ	説明

Status　　グラフィックディスプレイデバイスの状態の変化を示します。

第1.3版から電源状態通知を通知するStatusUpdateEvent値が追加されていることに注意してください。

ファームウェアの更新機能では、非同期でのファームウェア更新処理の状態と進行状況を通知するためのStatus値が追加されました。詳細は「第2章 共通プロパティ、メソッド、イベント」のStatusUpdateEventの記述を参照。

値	意味
GDSP_SUE_START_IMAGE_LOAD	画像読み込みを開始しました。
GDSP_SUE_END_IMAGE_LOAD	画像読み込みが完了しました。
GDSP_SUE_START_LOAD_WEBPAGE	WEBページの読み込みが開始されました。
GDSP_SUE_FINISH_LOAD_WEBPAGE	WEBページの読み込みが完了しました。
GDSP_SUE_CANCEL_LOAD_WEBPAGE	WEBページの読み込みをキャンセルしました。
GDSP_SUE_START_PLAY_VIDEO	動画再生を開始しました。
GDSP_SUE_STOP_PLAY_VIDEO	動画再生を停止しました。

説明　　グラフィックディスプレイデバイスの状態が変化したことを通知します。

参照　　「**イベント**」

付　録　A

OPOSレジストリの利用法

コントロールオブジェクトが正確にサービスオブジェクトと結びつき、対応するデバイスを初期化するためには、OPOSコントロールはシステムレジストリにデータを必要とします。

レジストリは階層構造に構成されていて、各レベルを「キー」と言います。各キーは以下のものからなります。

追加キー（「サブキー」ということもあります）

命名された「値」（ないこともあります）。値にはストリング、バイナリまたはダブルワードタイプの「データ」が定義されます。

ストリングタイプのデータで定義された1つの「デフォルト値」

OPOSはストリングデータしか定義しません。

サービスオブジェクトルートレジストリキー

すべてのOPOSサービスオブジェクトエントリは、次のメインキーの配下に配置されます。

HKEY_LOCAL_MACHINE¥SOFTWARE¥OLEforRetail¥ServiceOPOS

PCにローカルなソフトウェアコンフィギュレーションに対して推奨されるキーは"HKEY_LOCAL_MACHINE¥SOFTWARE"キーです。"OLEforRetail"キーはOLE関連コンフィギュレーション情報のすべてをまとめます。"ServiceOPOS"キーはOPOSサービスオブジェクト用コンフィギュレーション情報を保持します。

デバイスクラスキー

各クラスは、メインOPOSキー配下の固有デバイスクラスサブキーを持ち、以下のキー名が設定されています。

Belt
BillAcceptor
BillDispenser
Biometrics
BumpBar
CashChanger
CashDrawer
CAT
CheckScanner
CoinAcceptor
CoinDispenser

ElectronicJournal
ElectronicValueReaderWriter
FiscalPrinter
Gate
HardTotals
ImageScanner
ItemDispenser
Keylock
Lights
LineDisplay
MICR
MotionSensor
MSR
PINPad
PointCardRW
POSKeyBoard
POSPower
POSPrinter
RemoteOrderDisplay
RFIDScanner
Scale
Scanner
SignatureCapture
SmartCardRW
ToneIndicator

Video Capture
Individual Recognition
Sound Recorder
Voice Recognition
Sound Player
Speech Synthesis
Gesture Control
Device Monitor
Graphic Display

デバイス名キーと値

クラスに含まれる各デバイスは、そのクラスのキー配下にデバイス名サブキーが定義されます。これは、サービスオブジェクトのインストール処理で行わなければなりません。このデバイス名キーをアプリケーションがコントロールオブジェクトの**Open**メソッドに渡します。デバイス名には制約はありませんが、ただ、デバイスクラス配下の名前の中でユニークでなければなりません。

デバイス名キーのデフォルト値はサービスオブジェクトのプログラマティックID[10]です。このストリングをコントロールオブジェクトが必要とします。サービスオブジェクトをロードし、OLEオートメーションインタフェースをコントロールオブジェクトとサービスオブジェクトの間で確立できるようにするためです。

値 – 必須	データ
(Default)	SOのOLEプログラマティックID.

デバイスユニットキーの値とそのデータで、端末やPC上の実デバイスの特性を記述します。インストレーションエンジニア、サポートエンジニア用には次の値を強くお勧めします。

値 – 推奨	データ
Service	サービスオブジェクトのファイル名.
Description	サービスオブジェクトを示すストリング
Version	サービスオブジェクトのバージョン番号を示すストリング 一般的フォーマットは次のとおりです。 MajorVersion.MinorVersion.BuildVersion.

サービスオブジェクトの必要に応じてその他の値も指定できます。次のような情報を含めることができます。

通信ポート
ボーレート
シリアル回線特性
割込みリクエスト(IRQ)値
入出力(I/O)ポート

論理デバイス名の値

アプリケーションは、デバイス名キーを**Open**メソッドに渡すことによって、コントロールをオープンします。しかし、多くの場合、アプリケーションは、デバイス名に相当する「論理デバイス名」を指定できるレベルを必要とします。

---

[10] プログラマティックID("Prog ID")は、レジストリの"HKEY_CLASSES_ROOT"部になければならないキーの名前です。このキーには名前が"CLSID"のサブキー（Prog IDに関連のクラスID）がなければなりません。クラスIDは"HKEY_CLASSES_ROOT＼CLSID"レジストリ部の中のキーでなければなりません。このキーに、OLEオートメーションサーバタイプを指定するサブキーと、サーバを開始する方法をOLEに指示するサブキーが入っています。

論理デバイス名は、デバイスクラスキー内の値としてレジストリに定義します。論理デバイス名とした値は、同じデバイスクラスに入っているデバイス名キーと一致しなければなりません。

論理デバイス名をレジストリに追加するのは、アプリケーションインテグレータの役割です。（サービスオブジェクトのインストール処理では論理デバイス名を追加しません。）

サービスプロバイダルートレジストリキー

SOサービスプロバイダは、サービスオブジェクトの一部または全部に共通の情報を、レジストリに格納する必要があります。このデータには、インストレーションディレクトリ、インストール日、インストール解除情報などがあります。サービスプロバイダ情報は次のメインキー配下に配置します。

HKEY_LOCAL_MACHINE¥SOFTWARE¥OLEforRetail¥ServiceInfo

このキー配下のサブキーはサービスプロバイダの会社名にします。各サービスプロバイダの会社サブキー内のサブキーと値は、プロバイダによって異なります。

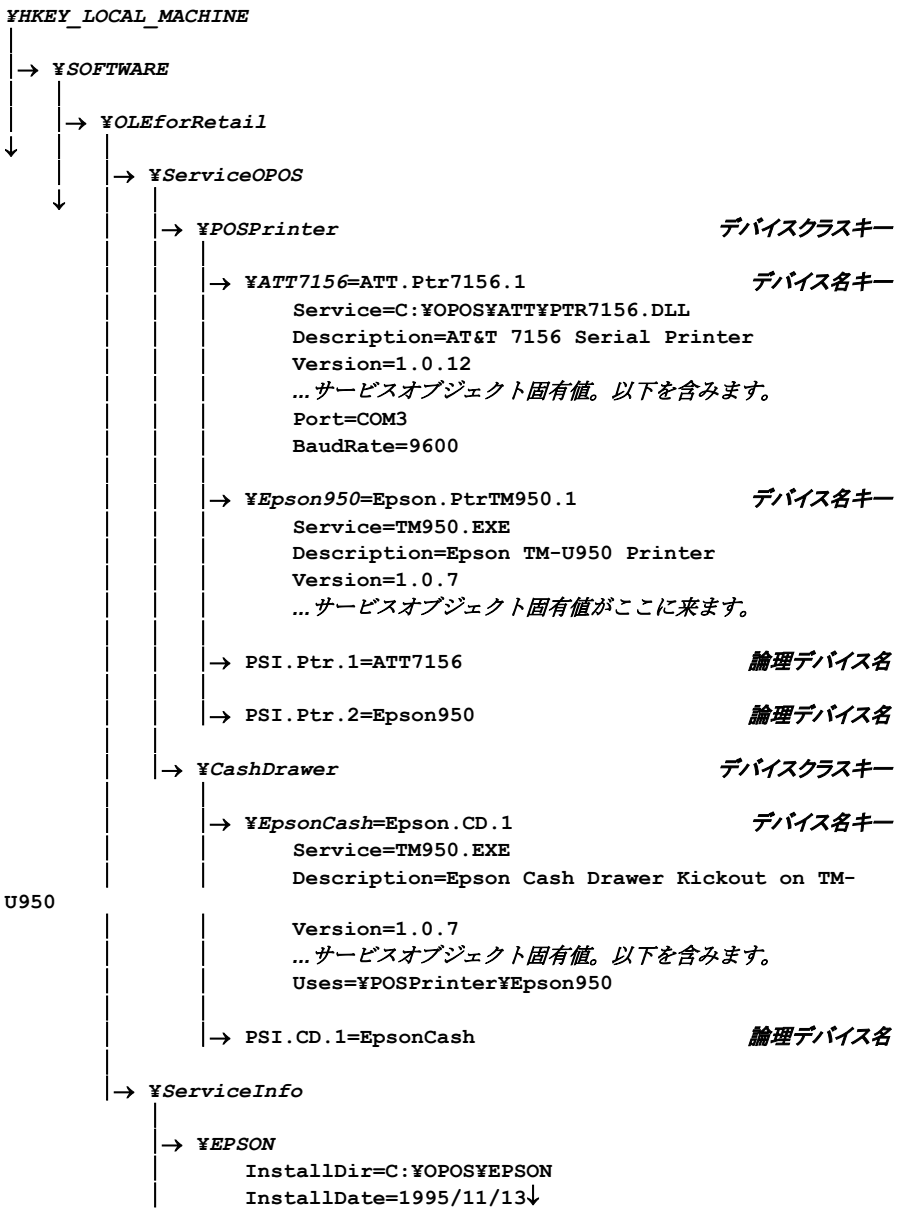
例

この例では、キーをイタリック体で示し、コメントも記載しています。

デバイスクラスはPOSPrinterとCashDrawerの2つです。

POSPrinterにはデバイス名が2つあります。また、論理デバイス名も2つあり、それらがデバイス名を指します。

CashDrawerクラスにはデバイス名1つと論理デバイス名1つが入っています。サービスオブジェクトはユニークなProg IDを持ちますが、プリンタの1つと同じ実行ファイルを指定しています。このサービスオブジェクトは、"Uses"を使用して、ドロワーパラメータ用に使用できる、プリンタデバイスのレジストリ値を指定しています。


¥HKEY_LOCAL_MACHINE
¥SOFTWARE
¥OLEforRetail
¥ServiceOPOS
¥POSPrinter
デバイスクラスキー
¥ATT7156=ATT.Ptr7156.1
デバイス名キー
Service=C:¥OPOS¥ATT¥PTR7156.DLL
Description=AT&T 7156 Serial Printer
Version=1.0.12
...サービスオブジェクト固有値。以下を含みます。
Port=COM3
BaudRate=9600
¥Epson950=Epson.PtrTM950.1
デバイス名キー
Service=TM950.EXE
Description=Epson TM-U950 Printer
Version=1.0.7
...サービスオブジェクト固有値がここに来ます。
PSI.Ptr.1=ATT7156
論理デバイス名
PSI.Ptr.2=Epson950
論理デバイス名
¥CashDrawer
デバイスクラスキー
¥EpsonCash=Epson.CD.1
デバイス名キー
Service=TM950.EXE
Description=Epson Cash Drawer Kickout on TM-U950
Version=1.0.7
...サービスオブジェクト固有値。以下を含みます。
Uses=¥POSPrinter¥Epson950
PSI.CD.1=EpsonCash
論理デバイス名
¥ServiceInfo
¥EPSON
InstallDir=C:¥OPOS¥EPSON
InstallDate=1995/11/13↓

付　録　B

OPOSで定義するヘッダーファイル

ヘッダーファイルは、アルファベット順に一覧にしてあります。
デバイスクラス名とヘッダーファイルの対応は、以下の通りです。

共通 –	Opos.h
Bill Acceptor	OposBacc.h
Bump Bar	OposBb.h
Bill Dispenser	OposBdsp.h
Belt	OposBelt.h
Biometrics	OposBio.h
Coin Acceptor	OposCacc.h
Cash Drawer	OposCash.h
CAT	OposCat.h
Cash Changer	OposChan.h
Check Scanner	OposChk.h
Coin Dispenser	OposCoin.h
Line Display	OposDisp.h
Device Monitor	OposDmon.h
Electronic Journal	OposEj.h
Electronic Value Reader Writer	OposEvrw.h
Fiscal Printer	OposFptr.h
Gate	OposGate.h
Gesture Control	OposGctl.h
Graphic Display	OposGdsp.h
Image Scanner	OposImg.h
Individual Recognition	OposIrcg.h
Item Dispenser	OposItem.h
POS Keyboard	OposKbd.h
Lights	OposLgt.h
Keylock	OposLock.h
MICR	OposMicr.h
Motion Sensor	OposMotion.h
MSR	OposMsr.h
PIN Pad	OposPpad.h
Point Card Reader Writer	OposPcrw.h
POS Printer	OposPtr.h
POS Power	OposPwr.h
Remote Order Display	OposRod.h
RFID Scanner	OposRfid.h
Scale	OposScal.h
Scanner	OposScan.h
Smart Card Reader Writer	OposScrw.h

Signature Capture	OposSig.h
Speech Synthesis	OposSpch.h
Sound Player	OposSply.h
Sound Recorder	OposSrec.h
Statistic name	OposStat.h
Tone Indicator	OposTone.h
Hard Totals	OposTot.h
Video Capture	OposVcap.h
Voice Recognition	OposVrcg.h

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。

1.14.1版以前　http://monroecs.com/

1.15版以後　　https://github.com/kunif/OPOS-CCO

Opos.h：　OPOS基本ヘッダーファイル

```
//////////////////////////////////////////////////////////////////////
//
// Opos.h
//
//  General header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 1995-12-08 OPOS Release 1.0                                  CRM
// 1997-06-04 OPOS Release 1.2                                  CRM
//  Add OPOS_FOREVER.
//  Add BinaryConversion constants.
// 1998-03-06 OPOS Release 1.3                                  CRM
//  Add CapPowerReporting, PowerState, and PowerNotify constants.
//  Add power reporting constants for StatusUpdateEvent.
// 2000-09-24 OPOS Release 1.5                                  CRM
//  Add OpenResult status constants.
// 2004-10-26 OPOS Release 1.8                                  CRM
//  Add "ResultCodeExtended" statistics constant.
// 2005-04-29 OPOS Release 1.9                                  CRM
//  Add CompareFirmwareVersion's Result constants.
//  Add StatusUpdateEvent and ResultCodeExtended constants
//    for firmware update.
// 2006-03-15 OPOS Release 1.10                                 CRM
//  Add another constant for firmware update.
// 2007-01-30 OPOS Release 1.11                                 CRM
//  Add ResultCode constant for deprecation.
// 2008-08-30 OPOS Release 1.12                                 CRM
//  Comment updates.
//
//////////////////////////////////////////////////////////////////////

#if !defined(OPOS_H)
#define     OPOS_H


//////////////////////////////////////////////////////////////////////
// OPOS "State" Property Constants
//////////////////////////////////////////////////////////////////////

const LONG OPOS_S_CLOSED       = 1;
const LONG OPOS_S_IDLE         = 2;
const LONG OPOS_S_BUSY         = 3;
const LONG OPOS_S_ERROR        = 4;


//////////////////////////////////////////////////////////////////////
// OPOS "ResultCode" Property Constants
//////////////////////////////////////////////////////////////////////

const LONG OPOS_SUCCESS        =   0;
const LONG OPOS_E_CLOSED       = 101;
const LONG OPOS_E_CLAIMED      = 102;
const LONG OPOS_E_NOTCLAIMED   = 103;
const LONG OPOS_E_NOSERVICE    = 104;
const LONG OPOS_E_DISABLED     = 105;
const LONG OPOS_E_ILLEGAL      = 106;
const LONG OPOS_E_NOHARDWARE   = 107;
const LONG OPOS_E_OFFLINE      = 108;
const LONG OPOS_E_NOEXIST      = 109;
const LONG OPOS_E_EXISTS       = 110;
const LONG OPOS_E_FAILURE      = 111;
const LONG OPOS_E_TIMEOUT      = 112;
const LONG OPOS_E_BUSY         = 113;
const LONG OPOS_E_EXTENDED     = 114;
const LONG OPOS_E_DEPRECATED   = 115; // (added in 1.11)

const LONG OPOSERR    = 100; // Base for ResultCode errors.
const LONG OPOSERREXT = 200; // Base for ResultCodeExtendedErrors.
```

```
////////////////////////////////////////////////////////////////////
// OPOS "ResultCodeExtended" Property Constants
////////////////////////////////////////////////////////////////////

// The following applies to ResetStatistics and UpdateStatistics.
const LONG OPOS_ESTATS_ERROR      = 280; // (added in 1.8)
const LONG OPOS_ESTATS_DEPENDENCY  = 282; // (added in 1.10)
// The following applies to CompareFirmwareVersion and UpdateFirmware.
const LONG OPOS_EFIRMWARE_BAD_FILE = 281; // (added in 1.9)


////////////////////////////////////////////////////////////////////
// OPOS "OpenResult" Property Constants (added in 1.5)
////////////////////////////////////////////////////////////////////

// The following can be set by the control object.
//  - Control Object already open.
const LONG OPOS_OR_ALREADYOPEN  = 301;
//   - The registry does not contain a key for the specified device
name.
const LONG OPOS_OR_REGBADNAME   = 302;
//  - Could not read the device name key's default value, or
//    could not convert this Prog ID to a valid Class ID.
const LONG OPOS_OR_REGPROGID    = 303;
//  - Could not create a service object instance, or
//    could not get its IDispatch interface.
const LONG OPOS_OR_CREATE       = 304;
//  - The service object does not support one or more of the
//    method required by its release.
const LONG OPOS_OR_BADIF        = 305;
//  - The service object returned a failure status from its
//    open call, but doesn't have a more specific failure code.
const LONG OPOS_OR_FAILEDOPEN   = 306;
//  - The service object major version number is not 1.
const LONG OPOS_OR_BADVERSION   = 307;

// The following can be returned by the service object if it
// returns a failure status from its open call.
//  - Port access required at open, but configured port
//    is invalid or inaccessible.
const LONG OPOS_ORS_NOPORT      = 401;
//  - Service Object does not support the specified device.
const LONG OPOS_ORS_NOTSUPPORTED= 402;
//  - Configuration information error.
const LONG OPOS_ORS_CONFIG      = 403;
//  - Errors greater than this value are SO-specific.
const LONG OPOS_ORS_SPECIFIC    = 450;


////////////////////////////////////////////////////////////////////
// OPOS "BinaryConversion" Property Constants (added in 1.2)
////////////////////////////////////////////////////////////////////

const LONG OPOS_BC_NONE         = 0;
const LONG OPOS_BC_NIBBLE       = 1;
const LONG OPOS_BC_DECIMAL      = 2;


////////////////////////////////////////////////////////////////////
// "CheckHealth" Method: "Level" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_CH_INTERNAL     = 1;
const LONG OPOS_CH_EXTERNAL     = 2;
const LONG OPOS_CH_INTERACTIVE  = 3;

////////////////////////////////////////////////////////////////////
// OPOS "CapPowerReporting", "PowerState", "PowerNotify" Property
//   Constants (added in 1.3)
////////////////////////////////////////////////////////////////////

const LONG OPOS_PR_NONE         = 0;
const LONG OPOS_PR_STANDARD     = 1;
const LONG OPOS_PR_ADVANCED     = 2;
```

```
const LONG OPOS_PN_DISABLED    = 0;
const LONG OPOS_PN_ENABLED     = 1;

const LONG OPOS_PS_UNKNOWN     = 2000;
const LONG OPOS_PS_ONLINE      = 2001;
const LONG OPOS_PS_OFF         = 2002;
const LONG OPOS_PS_OFFLINE     = 2003;
const LONG OPOS_PS_OFF_OFFLINE  = 2004;

////////////////////////////////////////////////////////////////////
// "CompareFirmwareVersion" Method: "Result" Parameter Constants
//   (added in 1.9)
////////////////////////////////////////////////////////////////////

const LONG OPOS_CFV_FIRMWARE_OLDER     = 1;
const LONG OPOS_CFV_FIRMWARE_SAME      = 2;
const LONG OPOS_CFV_FIRMWARE_NEWER     = 3;
const LONG OPOS_CFV_FIRMWARE_DIFFERENT  = 4;
const LONG OPOS_CFV_FIRMWARE_UNKNOWN   = 5;

////////////////////////////////////////////////////////////////////
// "ErrorEvent" Event: "ErrorLocus" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_EL_OUTPUT      = 1;
const LONG OPOS_EL_INPUT       = 2;
const LONG OPOS_EL_INPUT_DATA   = 3;

////////////////////////////////////////////////////////////////////
// "ErrorEvent" Event: "ErrorResponse" Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_ER_RETRY       = 11;
const LONG OPOS_ER_CLEAR       = 12;
const LONG OPOS_ER_CONTINUEINPUT= 13;

////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: Common "Status" Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_SUE_POWER_ONLINE                = 2001; // (added in
1.3)
const LONG OPOS_SUE_POWER_OFF                   = 2002; // (added in
1.3)
const LONG OPOS_SUE_POWER_OFFLINE               = 2003; // (added in
1.3)
const LONG OPOS_SUE_POWER_OFF_OFFLINE           = 2004; // (added in
1.3)

const LONG OPOS_SUE_UF_PROGRESS                 = 2100; // (added in
1.9)
const LONG OPOS_SUE_UF_COMPLETE                 = 2200; // (added in
1.9)
const LONG OPOS_SUE_UF_COMPLETE_DEV_NOT_RESTORED    = 2205; // (added
in 1.9)
const LONG OPOS_SUE_UF_FAILED_DEV_OK            = 2201; // (added in
1.9)
const LONG OPOS_SUE_UF_FAILED_DEV_UNRECOVERABLE    = 2202; // (added
in 1.9)
const LONG OPOS_SUE_UF_FAILED_DEV_NEEDS_FIRMWARE   = 2203; // (added
in 1.9)
const LONG OPOS_SUE_UF_FAILED_DEV_UNKNOWN       = 2204; // (added in
1.9)

////////////////////////////////////////////////////////////////////
// General Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_FOREVER       = -1; // (added in 1.2)

#endif               // !defined(OPOS_H)
```

OposBacc.h：　紙幣入金機ヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposBacc.h
//
//  Bill Acceptor header file for OPOS Applications.
//
// Modification history
// ------------------------------------------------------------------
// 2007-01-30 OPOS Release 1.11                                  CRM
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSBACC_H)
#define     OPOSBACC_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "FullStatus" Property Constants
// "StatusUpdateEvent" Event Constants
////////////////////////////////////////////////////////////////////

const LONG BACC_STATUS_OK             = 0;    // FullStatus

const LONG BACC_STATUS_FULL           = 21;   // FullStatus,
StatusUpdateEvent
const LONG BACC_STATUS_NEARFULL       = 22;   // FullStatus,
StatusUpdateEvent
const LONG BACC_STATUS_FULLOK         = 23;   // StatusUpdateEvent

const LONG BACC_STATUS_JAM            = 31;   // StatusUpdateEvent
const LONG BACC_STATUS_JAMOK          = 32;   // StatusUpdateEvent


////////////////////////////////////////////////////////////////////
// "DepositStatus" Property Constants
////////////////////////////////////////////////////////////////////

const LONG BACC_STATUS_DEPOSIT_START   = 1;
const LONG BACC_STATUS_DEPOSIT_END     = 2;
const LONG BACC_STATUS_DEPOSIT_COUNT   = 4;
const LONG BACC_STATUS_DEPOSIT_JAM     = 5;


////////////////////////////////////////////////////////////////////
// "EndDeposit" Method: "Success" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG BACC_DEPOSIT_COMPLETE      = 11;


////////////////////////////////////////////////////////////////////
// "PauseDeposit" Method: "Control" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG BACC_DEPOSIT_PAUSE         = 11;
const LONG BACC_DEPOSIT_RESTART       = 12;


#endif               // !defined(OPOSBACC_H)
```

OposBd.h :　バンプバーヘッダーファイル

```
///////////////////////////////////////////////////////////////////////
//
// OposBb.h
//
//  Bump Bar header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 1998-03-06 OPOS Release 1.3                                    BB
//
///////////////////////////////////////////////////////////////////////

#if !defined(OPOSBB_H)
#define     OPOSBB_H


#include "Opos.h"


///////////////////////////////////////////////////////////////////////
// "CurrentUnitID" and "UnitsOnline" Properties
//  and "Units" Parameter Constants
///////////////////////////////////////////////////////////////////////

#define BB_UID(Unit) (1 << (Unit-1))

const LONG BB_UID_1           = 0x00000001;
const LONG BB_UID_2           = 0x00000002;
const LONG BB_UID_3           = 0x00000004;
const LONG BB_UID_4           = 0x00000008;
const LONG BB_UID_5           = 0x00000010;
const LONG BB_UID_6           = 0x00000020;
const LONG BB_UID_7           = 0x00000040;
const LONG BB_UID_8           = 0x00000080;
const LONG BB_UID_9           = 0x00000100;
const LONG BB_UID_10          = 0x00000200;
const LONG BB_UID_11          = 0x00000400;
const LONG BB_UID_12          = 0x00000800;
const LONG BB_UID_13          = 0x00001000;
const LONG BB_UID_14          = 0x00002000;
const LONG BB_UID_15          = 0x00004000;
const LONG BB_UID_16          = 0x00008000;
const LONG BB_UID_17          = 0x00010000;
const LONG BB_UID_18          = 0x00020000;
const LONG BB_UID_19          = 0x00040000;
const LONG BB_UID_20          = 0x00080000;
const LONG BB_UID_21          = 0x00100000;
const LONG BB_UID_22          = 0x00200000;
const LONG BB_UID_23          = 0x00400000;
const LONG BB_UID_24          = 0x00800000;
const LONG BB_UID_25          = 0x01000000;
const LONG BB_UID_26          = 0x02000000;
const LONG BB_UID_27          = 0x04000000;
const LONG BB_UID_28          = 0x08000000;
const LONG BB_UID_29          = 0x10000000;
const LONG BB_UID_30          = 0x20000000;
const LONG BB_UID_31          = 0x40000000;
const LONG BB_UID_32          = 0x80000000;


///////////////////////////////////////////////////////////////////////
// "DataEvent" Event: "Status" Parameter Constants
///////////////////////////////////////////////////////////////////////

const LONG BB_DE_KEY          = 0x01;


#endif              // !defined(OPOSBB_H)
```

OposBdsp.h :　紙幣出金機ヘッダーファイル

```
/////////////////////////////////////////////////////////////////////
//
// OposBdsp.h
//
//   Bill Dispenser header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 2007-01-30 OPOS Release 1.11                                  CRM
//
/////////////////////////////////////////////////////////////////////

#if !defined(OPOSBDSP_H)
#define     OPOSBDSP_H


#include "Opos.h"


/////////////////////////////////////////////////////////////////////
// "DeviceStatus" Property Constants
// "StatusUpdateEvent" Event Constants
/////////////////////////////////////////////////////////////////////

const LONG BDSP_STATUS_OK             = 0;    // DeviceStatus

const LONG BDSP_STATUS_EMPTY           = 11;   // DeviceStatus,
StatusUpdateEvent
const LONG BDSP_STATUS_NEAREMPTY       = 12;   // DeviceStatus,
StatusUpdateEvent
const LONG BDSP_STATUS_EMPTYOK         = 13;   // StatusUpdateEvent

const LONG BDSP_STATUS_JAM            = 31;   // DeviceStatus,
StatusUpdateEvent
const LONG BDSP_STATUS_JAMOK           = 32;   // StatusUpdateEvent

const LONG BDSP_STATUS_ASYNC           = 91;   // StatusUpdateEvent


/////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
/////////////////////////////////////////////////////////////////////

const LONG OPOS_EBDSP_OVERDISPENSE     = 201; // DispenseCash


#endif               // !defined(OPOSBDSP_H)
```

OposBelt.h :　ベルトヘッダーファイル

```
///////////////////////////////////////////////////////////////////////
//
// OposBelt.h
//
//  Belt Drawer file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 2008-08-30 OPOS Release 1.12                                  CRM
//
///////////////////////////////////////////////////////////////////////

#if !defined(OPOSBELT_H)
#define     OPOSBELT_H


#include "Opos.h"


///////////////////////////////////////////////////////////////////////
// "MotionStatus" Property Constants
///////////////////////////////////////////////////////////////////////

const LONG BELT_MT_FORWARD            = 1;
const LONG BELT_MT_BACKWARD           = 2;
const LONG BELT_MT_STOPPED            = 3;
const LONG BELT_MT_EMERGENCY          = 4;
const LONG BELT_MT_MOTOR_FAULT        = 5;


///////////////////////////////////////////////////////////////////////
// "AdjustItemCount" Method: "Direction" Parameter Constants
///////////////////////////////////////////////////////////////////////

const LONG BELT_AIC_BACKWARD          = 1;
const LONG BELT_AIC_FORWARD           = 2;


///////////////////////////////////////////////////////////////////////
// "ResetItemCount" Method: "Direction" Parameter Constants
///////////////////////////////////////////////////////////////////////

const LONG BELT_RIC_BACKWARD          = 1;
const LONG BELT_RIC_FORWARD           = 2;


///////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constants
///////////////////////////////////////////////////////////////////////

const LONG BELT_SUE_AUTO_STOP                       = 11;
const LONG BELT_SUE_EMERGENCY_STOP                   = 12;
const LONG BELT_SUE_SAFETY_STOP                     = 13;
const LONG BELT_SUE_TIMEOUT_STOP                    = 14;
const LONG BELT_SUE_MOTOR_OVERHEATING                = 15;
const LONG BELT_SUE_MOTOR_FUSE_DEFECT                = 16;
const LONG BELT_SUE_LIGHT_BARRIER_BACKWARD_INTERRUPTED  = 17;
const LONG BELT_SUE_LIGHT_BARRIER_BACKWARD_OK         = 18;
const LONG BELT_SUE_LIGHT_BARRIER_FORWARD_INTERRUPTED   = 19;
const LONG BELT_SUE_LIGHT_BARRIER_FORWARD_OK          = 20;
const LONG BELT_SUE_SECURITY_FLAP_BACKWARD_OPENED      = 21;
const LONG BELT_SUE_SECURITY_FLAP_BACKWARD_CLOSED      = 22;
const LONG BELT_SUE_SECURITY_FLAP_FORWARD_OPENED       = 23;
const LONG BELT_SUE_SECURITY_FLAP_FORWARD_CLOSED       = 24;


#endif               // !defined(OPOSBELT_H)
```

OposBio.h :　バイオメトリクスヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposBio.h
//
//  Biometrics header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 2006-03-15 OPOS Release 1.10                                  CRM
// 2007-01-30 OPOS Release 1.11                                  CRM
//  Add constants to StatusUpdateEvent.
//  Corrected spelling of "keystroke dynamics" constants.
// 2008-08-30 OPOS Release 1.12                                  CRM
//  Add constant to StatusUpdateEvent.
// 2009-10-02 OPOS Release 1.13                                  CRM
//  Add constants to StatusUpdateEvent.
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSBIO_H)
#define     OPOSBIO_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "CapSensorColor" Property Constants
////////////////////////////////////////////////////////////////////

const LONG BIO_CSC_MONO               = 0x00000001;
const LONG BIO_CSC_GRAYSCALE           = 0x00000002;
const LONG BIO_CSC_16                 = 0x00000004;
const LONG BIO_CSC_256                = 0x00000008;
const LONG BIO_CSC_FULL               = 0x00000010;


////////////////////////////////////////////////////////////////////
// "CapSensorOrientation" Property Constants
////////////////////////////////////////////////////////////////////

const LONG BIO_CSO_NORMAL             = 0x00000001;
const LONG BIO_CSO_RIGHT              = 0x00000002;
const LONG BIO_CSO_INVERTED           = 0x00000004;
const LONG BIO_CSO_LEFT               = 0x00000008;


////////////////////////////////////////////////////////////////////
// "CapSensorType" Property Constants
////////////////////////////////////////////////////////////////////

const LONG BIO_CST_FACIAL_FEATURES     = 0x00000001;
const LONG BIO_CST_VOICE              = 0x00000002;
const LONG BIO_CST_FINGERPRINT         = 0x00000004;
const LONG BIO_CST_IRIS               = 0x00000008;
const LONG BIO_CST_RETINA             = 0x00000010;
const LONG BIO_CST_HAND_GEOMETRY       = 0x00000020;
const LONG BIO_CST_SIGNATURE_DYNAMICS   = 0x00000040;
const LONG BIO_CST_KEYSTROKE_DYNAMICS   = 0x00000080;
const LONG BIO_CST_LIP_MOVEMENT        = 0x00000100;
const LONG BIO_CST_THERMAL_FACE_IMAGE   = 0x00000200;
const LONG BIO_CST_THERMAL_HAND_IMAGE   = 0x00000400;
const LONG BIO_CST_GAIT               = 0x00000800;
const LONG BIO_CST_PASSWORD            = 0x00001000;


////////////////////////////////////////////////////////////////////
// "SensorColor" Property Constants
////////////////////////////////////////////////////////////////////

const LONG BIO_SC_MONO               = 1;
```

```
const LONG BIO_SC_GRAYSCALE          = 2;
const LONG BIO_SC_16                 = 3;
const LONG BIO_SC_256                = 4;
const LONG BIO_SC_FULL               = 5;

////////////////////////////////////////////////////////////////////
// "SensorOrientation" Property Constants
////////////////////////////////////////////////////////////////////

const LONG BIO_SO_NORMAL             = 1;
const LONG BIO_SO_RIGHT              = 2;
const LONG BIO_SO_INVERTED           = 3;
const LONG BIO_SO_LEFT               = 4;

////////////////////////////////////////////////////////////////////
// "SensorType" Property Constants
////////////////////////////////////////////////////////////////////

const LONG BIO_ST_FACIAL_FEATURES     =  1;
const LONG BIO_ST_VOICE              =  2;
const LONG BIO_ST_FINGERPRINT         =  3;
const LONG BIO_ST_IRIS               =  4;
const LONG BIO_ST_RETINA             =  5;
const LONG BIO_ST_HAND_GEOMETRY       =  6;
const LONG BIO_ST_SIGNATURE_DYNAMICS   =  7;
const LONG BIO_ST_KEYSTROKE_DYNAMICS   =  8;
const LONG BIO_ST_LIP_MOVEMENT        =  9;
const LONG BIO_ST_THERMAL_FACE_IMAGE   = 10;
const LONG BIO_ST_THERMAL_HAND_IMAGE   = 11;
const LONG BIO_ST_GAIT               = 12;
const LONG BIO_ST_PASSWORD           = 13;

////////////////////////////////////////////////////////////////////
// "DataEvent" "Status" Property Constants
////////////////////////////////////////////////////////////////////

const LONG BIO_DATA_ENROLL           = 1;
const LONG BIO_DATA_VERIFY           = 2;

////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" "Status" Property Constants
////////////////////////////////////////////////////////////////////

const LONG BIO_SUE_RAW_DATA          = 1;
const LONG BIO_SUE_MOVE_LEFT         = 2;
const LONG BIO_SUE_MOVE_RIGHT        = 3;
const LONG BIO_SUE_MOVE_DOWN         = 4;
const LONG BIO_SUE_MOVE_UP           = 5;
const LONG BIO_SUE_MOVE_CLOSER       = 6;
const LONG BIO_SUE_MOVE_AWAY         = 7;
const LONG BIO_SUE_MOVE_BACKWARD      = 8;
const LONG BIO_SUE_MOVE_FORWARD      = 9;
const LONG BIO_SUE_MOVE_SLOWER       = 10; // (added in 1.11)
const LONG BIO_SUE_MOVE_FASTER       = 11; // (added in 1.11)
const LONG BIO_SUE_SENSOR_DIRTY      = 12; // (added in 1.11)
const LONG BIO_SUE_FAILED_READ       = 13; // (added in 1.12)
const LONG BIO_SUE_SENSOR_READY      = 14; // (added in 1.13)
const LONG BIO_SUE_SENSOR_COMPLETE    = 15; // (added in 1.13)

#endif              // !defined(OPOSBIO_H)
```

OposCacc.h :　硬貨入金機ヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposCacc.h
//
//  Coin Acceptor header file for OPOS Applications.
//
// Modification history
// ------------------------------------------------------------------
// 2007-01-30 OPOS Release 1.11                                 CRM
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSCACC_H)
#define     OPOSCACC_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "FullStatus" Property Constants
// "StatusUpdateEvent" Event Constants
////////////////////////////////////////////////////////////////////

const LONG CACC_STATUS_OK             = 0;

const LONG CACC_STATUS_FULL           = 21;   // FullStatus,
StatusUpdateEvent
const LONG CACC_STATUS_NEARFULL       = 22;   // FullStatus,
StatusUpdateEvent
const LONG CACC_STATUS_FULLOK         = 23;   // StatusUpdateEvent

const LONG CACC_STATUS_JAM            = 31;   // StatusUpdateEvent
const LONG CACC_STATUS_JAMOK          = 32;   // StatusUpdateEvent


////////////////////////////////////////////////////////////////////
// "DepositStatus" Property Constants
////////////////////////////////////////////////////////////////////

const LONG CACC_STATUS_DEPOSIT_START   = 1;
const LONG CACC_STATUS_DEPOSIT_END     = 2;
const LONG CACC_STATUS_DEPOSIT_COUNT   = 4;
const LONG CACC_STATUS_DEPOSIT_JAM     = 5;


////////////////////////////////////////////////////////////////////
// "EndDeposit" Method: "Success" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG CACC_DEPOSIT_COMPLETE      = 11;


////////////////////////////////////////////////////////////////////
// "PauseDeposit" Method: "Control" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG CACC_DEPOSIT_PAUSE         = 11;
const LONG CACC_DEPOSIT_RESTART       = 12;


#endif               // !defined(OPOSCACC_H)
```

OposCash.h :　ドロワーヘッダーファイル

```
///////////////////////////////////////////////////////////////////////
//
// OposCash.h
//
//  Cash Drawer header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 1995-12-08 OPOS Release 1.0                                  CRM
// 1998-03-06 OPOS Release 1.3                                  CRM
//
///////////////////////////////////////////////////////////////////////

#if !defined(OPOSCASH_H)
#define     OPOSCASH_H


#include "Opos.h"


///////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event Constants
///////////////////////////////////////////////////////////////////////

const LONG CASH_SUE_DRAWERCLOSED = 0;
const LONG CASH_SUE_DRAWEROPEN   = 1;


#endif               // !defined(OPOSCASH_H)
```

OposCat.h :　CATヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposCAT.h
//
//  CAT header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 1998-06-01 OPOS Release 1.4                                OPOS-J
// 2000-09-24 OPOS Release 1.5                                  BKS
//  Added the following constants:
//    CAT_PAYMENT_DEBIT, CAT_MEDIA_...
// 2005-04-29 OPOS Release 1.9                                  CRM
//  Added constants to PaymentCondition, PaymentMedia,
//    TransactionType, and ResultCodeExtended.
//  Added LogStatus constants.
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSCAT_H)
#define     OPOSCAT_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
//  "CapDailyLog"  Property  and  "AccessDailyLog"  Type  Parameter
Constants
////////////////////////////////////////////////////////////////////

const LONG CAT_DL_NONE                  = 0;
const LONG CAT_DL_REPORTING              = 1;
const LONG CAT_DL_SETTLEMENT             = 2;
const LONG CAT_DL_REPORTING_SETTLEMENT     = 3;


////////////////////////////////////////////////////////////////////
// "LogStatus" Property Constants (added in 1.9)
////////////////////////////////////////////////////////////////////

const LONG CAT_LOGSTATUS_OK              = 1;
const LONG CAT_LOGSTATUS_NEARFULL         = 2;
const LONG CAT_LOGSTATUS_FULL             = 3;


////////////////////////////////////////////////////////////////////
// "PaymentCondition" Property Constants
////////////////////////////////////////////////////////////////////

const LONG CAT_PAYMENT_LUMP              =  10;
const LONG CAT_PAYMENT_BONUS_1            =  21;
const LONG CAT_PAYMENT_BONUS_2            =  22;
const LONG CAT_PAYMENT_BONUS_3            =  23;
const LONG CAT_PAYMENT_BONUS_4            =  24;
const LONG CAT_PAYMENT_BONUS_5            =  25;
const LONG CAT_PAYMENT_INSTALLMENT_1       =  61;
const LONG CAT_PAYMENT_INSTALLMENT_2       =  62;
const LONG CAT_PAYMENT_INSTALLMENT_3       =  63;
const LONG CAT_PAYMENT_BONUS_COMBINATION_1  =  31;
const LONG CAT_PAYMENT_BONUS_COMBINATION_2  =  32;
const LONG CAT_PAYMENT_BONUS_COMBINATION_3  =  33;
const LONG CAT_PAYMENT_BONUS_COMBINATION_4  =  34;
const LONG CAT_PAYMENT_REVOLVING          =  80;
const LONG CAT_PAYMENT_DEBIT              = 110; // (added in 1.5)
const LONG CAT_PAYMENT_ELECTRONIC_MONEY    = 111; // (added in 1.9)


////////////////////////////////////////////////////////////////////
// "PaymentMedia" Property Constants (added in 1.5)
////////////////////////////////////////////////////////////////////
```

```
const LONG CAT_MEDIA_UNSPECIFIED          = 0;
const LONG CAT_MEDIA_NONDEFINE            = 0; // (deprecated)
const LONG CAT_MEDIA_CREDIT               = 1;
const LONG CAT_MEDIA_DEBIT                = 2;
const LONG CAT_MEDIA_ELECTRONIC_MONEY      = 3; // (added in 1.9)


////////////////////////////////////////////////////////////////////
// "TransactionType" Property Constants
////////////////////////////////////////////////////////////////////

const LONG CAT_TRANSACTION_SALES          = 10;
const LONG CAT_TRANSACTION_VOID           = 20;
const LONG CAT_TRANSACTION_REFUND         = 21;
const LONG CAT_TRANSACTION_VOIDPRESALES    = 29;
const LONG CAT_TRANSACTION_COMPLETION      = 30;
const LONG CAT_TRANSACTION_PRESALES        = 40;
const LONG CAT_TRANSACTION_CHECKCARD       = 41;
const LONG CAT_TRANSACTION_CASHDEPOSIT     = 50; // (added in 1.9)


////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_ECAT_CENTERERROR          =  01;
const LONG OPOS_ECAT_COMMANDERROR         =  90;
const LONG OPOS_ECAT_RESET                =  91;
const LONG OPOS_ECAT_COMMUNICATIONERROR    =  92;
const LONG OPOS_ECAT_DAILYLOGOVERFLOW      = 200;
const LONG OPOS_ECAT_DEFICIENT            = 201; // (added in 1.9)
const LONG OPOS_ECAT_OVERDEPOSIT          = 202; // (added in 1.9)

#endif               // !defined(OPOSCAT_H)
```

OposChan.h：　自動釣り銭機ヘッダーファイル

```
///////////////////////////////////////////////////////////////////////
//
// OposChan.h
//
//   Cash Changer header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 97-06-04 OPOS Release 1.2                                    CRM
// 00-09-24 OPOS Release 1.5                                  OPOS-J
//   Add DepositStatus constants.
//   Add EndDeposit constants.
//   Add PauseDeposit constants.
//
///////////////////////////////////////////////////////////////////////

#if !defined(OPOSCHAN_H)
#define     OPOSCHAN_H


#include "Opos.h"


///////////////////////////////////////////////////////////////////////
// "DeviceStatus" and "FullStatus" Property Constants
// "StatusUpdateEvent" Event Constants
///////////////////////////////////////////////////////////////////////

const LONG CHAN_STATUS_OK      =  0; // DeviceStatus, FullStatus

const LONG CHAN_STATUS_EMPTY    = 11; // DeviceStatus,
StatusUpdateEvent
const LONG CHAN_STATUS_NEAREMPTY= 12; // DeviceStatus,
StatusUpdateEvent
const LONG CHAN_STATUS_EMPTYOK  = 13; // StatusUpdateEvent

const LONG CHAN_STATUS_FULL    = 21; // FullStatus,
StatusUpdateEvent
const LONG CHAN_STATUS_NEARFULL = 22; // FullStatus,
StatusUpdateEvent
const LONG CHAN_STATUS_FULLOK   = 23; // StatusUpdateEvent

const LONG CHAN_STATUS_JAM     = 31; // DeviceStatus,
StatusUpdateEvent
const LONG CHAN_STATUS_JAMOK    = 32; // StatusUpdateEvent

const LONG CHAN_STATUS_ASYNC    = 91; // StatusUpdateEvent


///////////////////////////////////////////////////////////////////////
// "DepositStatus" Property Constants
///////////////////////////////////////////////////////////////////////

const LONG CHAN_STATUS_DEPOSIT_START= 1;
const LONG CHAN_STATUS_DEPOSIT_END  = 2;
const LONG CHAN_STATUS_DEPOSIT_NONE = 3;
const LONG CHAN_STATUS_DEPOSIT_COUNT= 4;
const LONG CHAN_STATUS_DEPOSIT_JAM  = 5;


///////////////////////////////////////////////////////////////////////
// "EndDeposit" Method: "Success" Parameter Constants
```

```
////////////////////////////////////////////////////////////////////////

const LONG CHAN_DEPOSIT_CHANGE  = 1;
const LONG CHAN_DEPOSIT_NOCHANGE= 2;
const LONG CHAN_DEPOSIT_REPAY   = 3;


////////////////////////////////////////////////////////////////////////
// "PauseDeposit" Method: "Control" Parameter Constants
////////////////////////////////////////////////////////////////////////

const LONG CHAN_DEPOSIT_PAUSE   = 11;
const LONG CHAN_DEPOSIT_RESTART = 12;


////////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
////////////////////////////////////////////////////////////////////////

const LONG OPOS_ECHAN_OVERDISPENSE = 201;


#endif               // !defined(OPOSCHAN_H)
```

OposChk.h：　チェックスキャナヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposChk.h
//
//  Check Scanner header file for OPOS Applications.
//
// Modification history
// ------------------------------------------------------------------
// 2002-08-17 OPOS Release 1.7                                  CRM
// 2005-04-29 OPOS Release 1.9                                  CRM
//  Added Constrast constants.
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSCHK_H)
#define     OPOSCHK_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "CapColor" Capability Constants
////////////////////////////////////////////////////////////////////

const LONG CHK_CCL_MONO              = 0x00000001;
const LONG CHK_CCL_GRAYSCALE          = 0x00000002;
const LONG CHK_CCL_16                = 0x00000004;
const LONG CHK_CCL_256               = 0x00000008;
const LONG CHK_CCL_FULL              = 0x00000010;


////////////////////////////////////////////////////////////////////
// "CapImageFormat" Capability Constants
////////////////////////////////////////////////////////////////////

const LONG CHK_CIF_NATIVE             = 0x00000001;
const LONG CHK_CIF_TIFF              = 0x00000002;
const LONG CHK_CIF_BMP               = 0x00000004;
const LONG CHK_CIF_JPEG              = 0x00000008;
const LONG CHK_CIF_GIF               = 0x00000010;


////////////////////////////////////////////////////////////////////
// "Color" Property Constants
////////////////////////////////////////////////////////////////////

const LONG CHK_CL_MONO               = 1;
const LONG CHK_CL_GRAYSCALE           = 2;
const LONG CHK_CL_16                 = 3;
const LONG CHK_CL_256                = 4;
const LONG CHK_CL_FULL               = 5;


////////////////////////////////////////////////////////////////////
// "Constrast" Property Constants (added in 1.9)
////////////////////////////////////////////////////////////////////

const LONG CHK_AUTOMATIC_CONTRAST      = -1;


////////////////////////////////////////////////////////////////////
```

```
// "ImageFormat" Property Constants
////////////////////////////////////////////////////////////////////////

const LONG CHK_IF_NATIVE             = 1;
const LONG CHK_IF_TIFF               = 2;
const LONG CHK_IF_BMP                = 3;
const LONG CHK_IF_JPEG               = 4;
const LONG CHK_IF_GIF                = 5;


////////////////////////////////////////////////////////////////////////
// "ImageMemoryStatus" Property Constants
////////////////////////////////////////////////////////////////////////

const LONG CHK_IMS_EMPTY             = 1;
const LONG CHK_IMS_OK                = 2;
const LONG CHK_IMS_FULL              = 3;


////////////////////////////////////////////////////////////////////////
// "MapMode" Property Constants
////////////////////////////////////////////////////////////////////////

const LONG CHK_MM_DOTS               = 1;
const LONG CHK_MM_TWIPS              = 2;
const LONG CHK_MM_ENGLISH            = 3;
const LONG CHK_MM_METRIC             = 4;


////////////////////////////////////////////////////////////////////////
// "ClearImage" Method Constants:
////////////////////////////////////////////////////////////////////////

//   "By" Parameter
const LONG CHK_CLR_ALL               = 1;
const LONG CHK_CLR_BY_FILEID         = 2;
const LONG CHK_CLR_BY_FILEINDEX      = 3;
const LONG CHK_CLR_BY_IMAGETAGDATA   = 4;


////////////////////////////////////////////////////////////////////////
// "DefineCropArea" Method Constants:
////////////////////////////////////////////////////////////////////////

// "CropAreaID" Parameter or index number
const LONG CHK_CROP_AREA_ENTIRE_IMAGE  = -1;
const LONG CHK_CROP_AREA_RESET_ALL     = -2;

// "CX" Parameter or integer width
const LONG CHK_CROP_AREA_RIGHT       = -1;

// "CY" Parameter or integer height
const LONG CHK_CROP_AREA_BOTTOM      = -1;


////////////////////////////////////////////////////////////////////////
// "RetrieveMemory" Method Constants:
////////////////////////////////////////////////////////////////////////

// "By" Parameter
const LONG CHK_LOCATE_BY_FILEID      = 1;
const LONG CHK_LOCATE_BY_FILEINDEX   = 2;
const LONG CHK_LOCATE_BY_IMAGETAGDATA  = 3;


////////////////////////////////////////////////////////////////////////
```

```
// "RetrieveImage" and "StoreImage" Method Constant:
//////////////////////////////////////////////////////////////////////

// "CropAreaID" Parameter or index number
//const LONG CHK_CROP_AREA_ENTIRE_IMAGE   = -1; //(Defined above)


//////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constant
//////////////////////////////////////////////////////////////////////

const LONG CHK_SUE_SCANCOMPLETE        = 11;


//////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
//////////////////////////////////////////////////////////////////////

const LONG OPOS_ECHK_NOCHECK           = 201; // endInsertion
const LONG OPOS_ECHK_CHECK             = 202; // endRemoval
const LONG OPOS_ECHK_NOROOM            = 203; // storeImage


#endif               // !defined(OPOSCHK_H)
```

OposCoin.h :　コインディスペンサヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposCoin.h
//
//  Coin Dispenser header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------
// 1995-12-08 OPOS Release 1.0                              CRM
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSCOIN_H)
#define     OPOSCOIN_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "DispenserStatus" Property Constants
// "StatusUpdateEvent" Event: "Data" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG COIN_STATUS_OK      = 1;
const LONG COIN_STATUS_EMPTY    = 2;
const LONG COIN_STATUS_NEAREMPTY= 3;
const LONG COIN_STATUS_JAM     = 4;


#endif               // !defined(OPOSCOIN_H)
```

OposDisp.h：　ラインディスプレイヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposDisp.h
//
//  Line Display header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 1995-12-08 OPOS Release 1.0                                  CRM
// 1996-03-18 OPOS Release 1.01                                 CRM
//  Add DISP_MT_INIT constant and MarqueeFormat constants.
// 1996-04-22 OPOS Release 1.1                                  CRM
//  Add CapCharacterSet constants for Kana and Kanji.
// 2000-09-24 OPOS Release 1.5                                  BKS
//  Add CapCharacterSet and CharacterSet constants for Unicode
// 2001-07-15 OPOS Release 1.6                                  BKS
//  Add CapCursorType, CapReadBack, CapReverse, CursorType
//    property constants.
//  Add DefineGlyph, DisplayText and DisplayTextAt parameter
//    constants.
// 2002-08-17 OPOS Release 1.7                                  CRM
//  Add DisplayBitmap and SetBitmap parameter constants.
// 2004-03-22 OPOS Release 1.8                                  CRM
//  Add more constants for CapCursorType and CursorType.
// 2004-10-26 Add "CharacterSet" ANSI constant (from 1.5).      CRM
//  Add ResultCodeExtended constants (from 1.7).
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSDISP_H)
#define     OPOSDISP_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "CapBlink" Property Constants
////////////////////////////////////////////////////////////////////

const LONG DISP_CB_NOBLINK     = 0;
const LONG DISP_CB_BLINKALL    = 1;
const LONG DISP_CB_BLINKEACH   = 2;


////////////////////////////////////////////////////////////////////
// "CapCharacterSet" Property Constants
////////////////////////////////////////////////////////////////////

const LONG DISP_CCS_NUMERIC    =   0;
const LONG DISP_CCS_ALPHA      =   1;
const LONG DISP_CCS_ASCII      = 998;
const LONG DISP_CCS_KANA       =  10; // (added in 1.1)
const LONG DISP_CCS_KANJI      =  11; // (added in 1.1)
const LONG DISP_CCS_UNICODE    = 997; // (added in 1.5)


////////////////////////////////////////////////////////////////////
// "CapCursorType" Property Constants (added in 1.6)
////////////////////////////////////////////////////////////////////

const LONG DISP_CCT_NONE       =   0x00000000;
const LONG DISP_CCT_FIXED      =   0x00000001;
const LONG DISP_CCT_BLOCK      =   0x00000002;
const LONG DISP_CCT_HALFBLOCK   =   0x00000004;
const LONG DISP_CCT_UNDERLINE   =   0x00000008;
const LONG DISP_CCT_REVERSE    =   0x00000010;
const LONG DISP_CCT_OTHER      =   0x00000020;
const LONG DISP_CCT_BLINK      =   0x00000040;  // (added in 1.8)
```

```
////////////////////////////////////////////////////////////////////
// "CapReadBack" Property Constants (added in 1.6)
////////////////////////////////////////////////////////////////////

const LONG DISP_CRB_NONE       =   0x00000000;
const LONG DISP_CRB_SINGLE     =   0x00000001;


////////////////////////////////////////////////////////////////////
// "CapReverse" Property Constants (added in 1.6)
////////////////////////////////////////////////////////////////////

const LONG DISP_CR_NONE        =   0x00000000;
const LONG DISP_CR_REVERSEALL   =   0x00000001;
const LONG DISP_CR_REVERSEEACH  =   0x00000002;


////////////////////////////////////////////////////////////////////
// "CharacterSet" Property Constants
////////////////////////////////////////////////////////////////////

const LONG DISP_CS_UNICODE     = 997; // (added in 1.5)
const LONG DISP_CS_ASCII       = 998;
const LONG DISP_CS_WINDOWS     = 999;
const LONG DISP_CS_ANSI        = 999; // (added in 1.5)


////////////////////////////////////////////////////////////////////
// "CursorType" Property Constants (added in 1.6)
////////////////////////////////////////////////////////////////////

const LONG DISP_CT_NONE        =   0;
const LONG DISP_CT_FIXED       =   1;
const LONG DISP_CT_BLOCK       =   2;
const LONG DISP_CT_HALFBLOCK    =   3;
const LONG DISP_CT_UNDERLINE    =   4;
const LONG DISP_CT_REVERSE     =   5;
const LONG DISP_CT_OTHER       =   6;
const LONG DISP_CT_BLINK       =   0x10000000;  // (added in 1.8)


////////////////////////////////////////////////////////////////////
// "MarqueeType" Property Constants
////////////////////////////////////////////////////////////////////

const LONG DISP_MT_NONE        = 0;
const LONG DISP_MT_UP          = 1;
const LONG DISP_MT_DOWN        = 2;
const LONG DISP_MT_LEFT        = 3;
const LONG DISP_MT_RIGHT       = 4;
const LONG DISP_MT_INIT        = 5; // (added in 1.01)


////////////////////////////////////////////////////////////////////
// "MarqueeFormat" Property Constants (added in 1.01)
////////////////////////////////////////////////////////////////////

const LONG DISP_MF_WALK        = 0;
const LONG DISP_MF_PLACE       = 1;


////////////////////////////////////////////////////////////////////
// "DefineGlyph" Method: "GlyphType" Parameter Constants (added in
1.6)
////////////////////////////////////////////////////////////////////

const LONG DISP_GT_SINGLE      = 1;


////////////////////////////////////////////////////////////////////
// "DisplayText" Method: "Attribute" Property Constants (added in 1.6)
// "DisplayTextAt" Method: "Attribute" Property Constants (added in
1.6)
////////////////////////////////////////////////////////////////////
```

```
const LONG DISP_DT_NORMAL       = 0;
const LONG DISP_DT_BLINK        = 1;
const LONG DISP_DT_REVERSE      = 2;
const LONG DISP_DT_BLINK_REVERSE = 3;


////////////////////////////////////////////////////////////////////
// "ScrollText" Method: "Direction" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG DISP_ST_UP          = 1;
const LONG DISP_ST_DOWN         = 2;
const LONG DISP_ST_LEFT         = 3;
const LONG DISP_ST_RIGHT        = 4;


////////////////////////////////////////////////////////////////////
// "SetDescriptor" Method: "Attribute" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG DISP_SD_OFF          = 0;
const LONG DISP_SD_ON           = 1;
const LONG DISP_SD_BLINK        = 2;


////////////////////////////////////////////////////////////////////
// "DisplayBitmap" and "SetBitmap" Method Constants (added in 1.7)
////////////////////////////////////////////////////////////////////

//   "Width" Parameter

const LONG DISP_BM_ASIS         = -11;

//   "AlignmentX" Parameter

const LONG DISP_BM_LEFT         = -1;
const LONG DISP_BM_CENTER       = -2;
const LONG DISP_BM_RIGHT        = -3;

//   "AlignmentY" Parameter

const LONG DISP_BM_TOP          = -1;
//const LONG DISP_BM_CENTER     = -2;
const LONG DISP_BM_BOTTOM       = -3;


////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_EDISP_TOOBIG    = 201; // DisplayBitmap (added in 1.7)
const LONG OPOS_EDISP_BADFORMAT = 202; // DisplayBitmap (added in 1.7)


#endif                // !defined(OPOSDISP_H)
```

OposDmon.h :　デバイスモニタヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposDmon.h
//
//  Device Monitor header file for OPOS Applications.
//
// Modification history
// ------------------------------------------------------------------
// 2021-01-26 OPOS Release 1.16                              KF
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSDMON_H)
#define     OPOSDMON_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "AddMonitoringDevice" Methods: "MonitoringMode" Parameter
Constants
////////////////////////////////////////////////////////////////////

const LONG DMON_MMODE_UPDATE         = 1;
const LONG DMON_MMODE_STRADDLED      = 2;
const LONG DMON_MMODE_HIGH           = 3;
const LONG DMON_MMODE_LOW            = 4;
const LONG DMON_MMODE_WITHIN         = 5;
const LONG DMON_MMODE_OUTSIDE        = 6;
const LONG DMON_MMODE_POLLING        = 7;


////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG DMON_SUE_START_MONITORING  = 11;
const LONG DMON_SUE_STOP_MONITORING   = 12;


#endif               // !defined(OPOSDMON_H)
```

OposEj.h :　電子ジャーナルヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposEj.h
//
//  Electronic Journal header file for OPOS Applications.
//
// Modification history
// ------------------------------------------------------------------
// 2006-03-15 OPOS Release 1.10                                 CRM
// 2008-08-30 OPOS Release 1.12                                 CRM
//  Add constant to StatusUpdateEvent.
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSEJ_H)
#define     OPOSEJ_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "CapStation", "Station" Property Constants
////////////////////////////////////////////////////////////////////

const LONG EJ_S_RECEIPT              = 0x00000001;
const LONG EJ_S_SLIP                = 0x00000002;
const LONG EJ_S_JOURNAL              = 0x00000004;


////////////////////////////////////////////////////////////////////
// "retrieveCurrentMarker" Method, "markerType" Parameter Constants
// "retrieveMarker" Method, "markerType" Parameter Constants
// "retrieveMarkerByDateTime" Method, "markerType" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG EJ_MT_SESSION_BEG          = 1;
const LONG EJ_MT_SESSION_END          = 2;
const LONG EJ_MT_DOCUMENT             = 3;
const LONG EJ_MT_HEAD                 = 4;
const LONG EJ_MT_TAIL                 = 5;


////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property and "ErrorEvent" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_EEJ_EXISTING          = 201;
const LONG OPOS_EEJ_MEDIUM_FULL       = 202;
const LONG OPOS_EEJ_MULTIPLE_MARKER   = 203;
const LONG OPOS_EEJ_UNINITIALIZED_MEDIUM= 204;
const LONG OPOS_EEJ_CORRUPTED_MEDIUM  = 205;
const LONG OPOS_EEJ_UNKNOWN_DATAFORMAT = 206;
const LONG OPOS_EEJ_NOT_ENOUGH_SPACE  = 207;
const LONG OPOS_EEJ_MULTIPLE_MARKERS  = 208;

////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" "Status" Property Constants
////////////////////////////////////////////////////////////////////

const LONG EJ_SUE_MEDIUM_NEAR_FULL    = 1;
const LONG EJ_SUE_MEDIUM_FULL         = 2;
const LONG EJ_SUE_MEDIUM_REMOVED      = 3;
const LONG EJ_SUE_MEDIUM_INSERTED     = 4;
const LONG EJ_SUE_SUSPENDED           = 5;
const LONG EJ_SUE_IDLE                = 1001; // (added in 1.12)

#endif              // !defined(OPOSEJ_H)
```

OposEvrw.h : 電子バリューリーダライタヘッダーファイル

```
//////////////////////////////////////////////////////////////////////
//
// OposEvrw.h
//
//  Electronic Value Reader Writer header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 2008-08-30 OPOS Release 1.12                                  CRM
// 2013-10-03 OPOS Release 1.14                                  CRM
//  Added PINEntry constants.
//  Added TrainingModeState constants.
//  Added RetrieveResultInformation's Value constants.
//  Added TransitionEvent's EventNumber constants.
// 2015-01-19 OPOS Release 1.14.001                               CRM
//  Added ServiceType constants.
//  Added AccessData's DataType constants.
//  Additional RetrieveResultInformation's Value constants.
//  Additional TransitionEvent's EventNumber constants.
// 2020-06-01 OPOS Release 1.15                                  KF
//  Added CapDailyLog and AccessDailyLog constants.
//  Added PaymentCondition constants.
//  Added PaymentMedia constants.
//  Added TransactionType constants.
//  Additional RetrieveResultInformation's Value constants.
//  Additional ServiceType's Value constants.
//  Additional TransitionEvent's EventNumber constants.
// 2025-09-17 OPOS Release 1.15.1                                 KF
//  Additional RetrieveResultInformation's Value constants.
//  Update version comment of EVRW_ST_CAT.
//
//////////////////////////////////////////////////////////////////////

#if !defined(OPOSEVRW_H)
#define     OPOSEVRW_H


#include "Opos.h"


//////////////////////////////////////////////////////////////////////
// "CapCardSensor" Property Constants
//////////////////////////////////////////////////////////////////////

const LONG EVRW_CCS_ENTRY             = 0x00000001;
const LONG EVRW_CCS_DETECT            = 0x00000002;
const LONG EVRW_CCS_CAPTURE           = 0x00000004;


//////////////////////////////////////////////////////////////////////
// "CapDetectionControl" Property Constants
//////////////////////////////////////////////////////////////////////

const LONG EVRW_CDC_RWCONTROL         = 0x00000001;
const LONG EVRW_CDC_APPLICATIONCONTROL  = 0x00000002;


//////////////////////////////////////////////////////////////////////
// "DetectionStatus" Property Constants
//////////////////////////////////////////////////////////////////////

const LONG EVRW_DS_NOCARD             = 1;
const LONG EVRW_DS_DETECTED           = 2;
const LONG EVRW_DS_ENTERED            = 3;
const LONG EVRW_DS_CAPTURED           = 4;


//////////////////////////////////////////////////////////////////////
//  "CapDailyLog"  Property  and  "AccessDailyLog"  Type  Parameter
Constants
// (added in 1.15)
```

```
////////////////////////////////////////////////////////////////////

const LONG EVRW_DL_NONE               = 0;
const LONG EVRW_DL_REPORTING           = 1;
const LONG EVRW_DL_SETTLEMENT          = 2;
const LONG EVRW_DL_REPORTING_SETTLEMENT = 3;


////////////////////////////////////////////////////////////////////
// "LogStatus" Property Constants
////////////////////////////////////////////////////////////////////

const LONG EVRW_LS_OK                 = 1;
const LONG EVRW_LS_NEARFULL            = 2;
const LONG EVRW_LS_FULL               = 3;


////////////////////////////////////////////////////////////////////
// "PaymentCondition" Property Constants (added in 1.15)
////////////////////////////////////////////////////////////////////

const LONG EVRW_PAYMENT_LUMP              =  10;
const LONG EVRW_PAYMENT_BONUS_1            =  21;
const LONG EVRW_PAYMENT_BONUS_2            =  22;
const LONG EVRW_PAYMENT_BONUS_3            =  23;
const LONG EVRW_PAYMENT_BONUS_4            =  24;
const LONG EVRW_PAYMENT_BONUS_5            =  25;
const LONG EVRW_PAYMENT_INSTALLMENT_1       =  61;
const LONG EVRW_PAYMENT_INSTALLMENT_2       =  62;
const LONG EVRW_PAYMENT_INSTALLMENT_3       =  63;
const LONG EVRW_PAYMENT_BONUS_COMBINATION_1 =  31;
const LONG EVRW_PAYMENT_BONUS_COMBINATION_2 =  32;
const LONG EVRW_PAYMENT_BONUS_COMBINATION_3 =  33;
const LONG EVRW_PAYMENT_BONUS_COMBINATION_4 =  34;
const LONG EVRW_PAYMENT_REVOLVING          =  80;
const LONG EVRW_PAYMENT_DEBIT             = 110;
const LONG EVRW_PAYMENT_ELECTRONIC_MONEY    = 111;


////////////////////////////////////////////////////////////////////
// "PaymentMedia" Property Constants (added in 1.15)
////////////////////////////////////////////////////////////////////

const LONG EVRW_MEDIA_UNSPECIFIED          = 0;
const LONG EVRW_MEDIA_CREDIT              = 1;
const LONG EVRW_MEDIA_DEBIT               = 2;
const LONG EVRW_MEDIA_ELECTRONIC_MONEY      = 3;


////////////////////////////////////////////////////////////////////
// "TransactionType" Property Constants (added in 1.15)
////////////////////////////////////////////////////////////////////

const LONG EVRW_TRANSACTION_SALES          = 10;
const LONG EVRW_TRANSACTION_VOID           = 20;
const LONG EVRW_TRANSACTION_REFUND         = 21;
const LONG EVRW_TRANSACTION_VOIDPRESALES    = 29;
const LONG EVRW_TRANSACTION_COMPLETION      = 30;
const LONG EVRW_TRANSACTION_PRESALES        = 40;
const LONG EVRW_TRANSACTION_CHECKCARD       = 41;
const LONG EVRW_TRANSACTION_CASHDEPOSIT     = 50;


////////////////////////////////////////////////////////////////////
// "PINEntry" Property Constants (added in 1.14)
////////////////////////////////////////////////////////////////////

const LONG EVRW_PIN_ENTRY_NONE         = 1;
const LONG EVRW_PIN_ENTRY_EXTERNAL      = 2;
const LONG EVRW_PIN_ENTRY_INTERNAL      = 3;
const LONG EVRW_PIN_ENTRY_UNKNOWN       = 4;


////////////////////////////////////////////////////////////////////
// "ServiceType" Property Constants (added in 1.14.1)
```

```
////////////////////////////////////////////////////////////////////////

const LONG EVRW_ST_UNSPECIFIED         = 0;
const LONG EVRW_ST_ELECTRONIC_MONEY    = 1;
const LONG EVRW_ST_POINT               = 2;
const LONG EVRW_ST_VOUCHER             = 3;
const LONG EVRW_ST_MEMBERSHIP          = 4;
const LONG EVRW_ST_CAT                 = 5; // (added in 1.15.1)

////////////////////////////////////////////////////////////////////////
// "TrainingModeState" Property Constants (added in 1.14)
////////////////////////////////////////////////////////////////////////

const LONG EVRW_TM_FALSE               = 1;
const LONG EVRW_TM_TRUE                = 2;
const LONG EVRW_TM_UNKNOWN             = 3;


////////////////////////////////////////////////////////////////////////
// "AccessData" Method: "DataType" Parameter Constants
//    (added in 1.14.1)
////////////////////////////////////////////////////////////////////////

const LONG EVRW_AD_KEY                 = 1;
const LONG EVRW_AD_NEGATIVE_LIST       = 2;
const LONG EVRW_AD_OTHERS              = 3;


////////////////////////////////////////////////////////////////////////
// "AccessLog" Method: "Type" Parameter Constants
////////////////////////////////////////////////////////////////////////

const LONG EVRW_AL_REPORTING           = 1;
const LONG EVRW_AL_SETTLEMENT          = 2;


////////////////////////////////////////////////////////////////////////
// "BeginDetection" Method: "Type" Parameter Constants
////////////////////////////////////////////////////////////////////////

const LONG EVRW_BD_ANY                 = 1;
const LONG EVRW_BD_SPECIFIC            = 2;


////////////////////////////////////////////////////////////////////////
// "RetrieveResultInformation" Method: "Value" Parameter Constants
//   (added in 1.14)
////////////////////////////////////////////////////////////////////////

const LONG EVRW_TAG_AS_AUTHENTICATED   = 1;
const LONG EVRW_TAG_AS_UNAUTHENTICATED  = 2;

const LONG EVRW_TAG_CTT_CANCEL         = 1;
const LONG EVRW_TAG_CTT_CHARGE         = 2;
const LONG EVRW_TAG_CTT_RETURN         = 3;
const LONG EVRW_TAG_CTT_SALES          = 4;

const LONG EVRW_TAG_CM_CASH            = 1;
const LONG EVRW_TAG_CM_CREDIT          = 2;
const LONG EVRW_TAG_CM_POINT           = 3;

const LONG EVRW_TAG_NIT_ALL            = 1; // (added in 1.14)
const LONG EVRW_TAG_NIT_UPDATED        = 2; // (added in 1.14)

const LONG EVRW_TAG_PC_INSTALLMENT_1      =  1;
const LONG EVRW_TAG_PC_INSTALLMENT_2      =  2;
const LONG EVRW_TAG_PC_INSTALLMENT_3      =  3;
const LONG EVRW_TAG_PC_BONUS_1            =  4;
const LONG EVRW_TAG_PC_BONUS_2            =  5;
const LONG EVRW_TAG_PC_BONUS_3            =  6;
const LONG EVRW_TAG_PC_BONUS_4            =  7;
const LONG EVRW_TAG_PC_BONUS_5            =  8;
const LONG EVRW_TAG_PC_BONUS_COMBINATION_1 =  9;
const LONG EVRW_TAG_PC_BONUS_COMBINATION_2 = 10;
const LONG EVRW_TAG_PC_BONUS_COMBINATION_3 = 11;
```

```
const LONG EVRW_TAG_PC_BONUS_COMBINATION_4 = 12;
const LONG EVRW_TAG_PC_LUMP              = 13;
const LONG EVRW_TAG_PC_REVOLVING         = 14;

const LONG EVRW_TAG_PM_COMBINED        = 1;
const LONG EVRW_TAG_PM_FULL_SETTLEMENT  = 2;

const LONG EVRW_TAG_PMFP_CASH          = 1;
const LONG EVRW_TAG_PMFP_CREDIT        = 2;
const LONG EVRW_TAG_PMFP_EM            = 3;
const LONG EVRW_TAG_PMFP_OTHER         = 4;

const LONG EVRW_TAG_ROS_NG             = 1;
const LONG EVRW_TAG_ROS_OK             = 2;
const LONG EVRW_TAG_ROS_UNKNOWN        = 3;

const LONG EVRW_TAG_STT_1              = 1;
const LONG EVRW_TAG_STT_2              = 2;
const LONG EVRW_TAG_STT_3              = 3;

const LONG EVRW_TAG_TT_ADD             = 1;
const LONG EVRW_TAG_TT_CANCEL_CHARGE    = 2;
const LONG EVRW_TAG_TT_CANCEL_RETURN    = 3;
const LONG EVRW_TAG_TT_CANCEL_SALES     = 4;
const LONG EVRW_TAG_TT_GET_LOG          = 5;
const LONG EVRW_TAG_TT_READ            = 6;
const LONG EVRW_TAG_TT_RETURN           = 7;
const LONG EVRW_TAG_TT_SUBTRACT         = 8;
const LONG EVRW_TAG_TT_WRITE           = 9;
const LONG EVRW_TAG_TT_COMPLETION       = 10; // (added in 1.15)
const LONG EVRW_TAG_TT_PRE_SALES        = 11; // (added in 1.15)

const LONG EVRW_TAG_VR_VOID            = 1; // (added in 1.15.1)
const LONG EVRW_TAG_VR_RETURN           = 2; // (added in 1.15.1)

const LONG EVRW_TAG_VTT_SALES           = 1; // (added in 1.15.1)
const LONG EVRW_TAG_VTT_CHARGE          = 2; // (added in 1.15.1)
const LONG EVRW_TAG_VTT_CANCEL          = 3; // (added in 1.15.1)
const LONG EVRW_TAG_VTT_RETURN          = 4; // (added in 1.15.1)


////////////////////////////////////////////////////////////////////////
// "TransactionAccess" Method: "Control" Parameter Constants
////////////////////////////////////////////////////////////////////////

const LONG EVRW_TA_TRANSACTION          = 11;
const LONG EVRW_TA_NORMAL              = 12;


////////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constants
////////////////////////////////////////////////////////////////////////

const LONG EVRW_SUE_LS_OK                = 11;
const LONG EVRW_SUE_LS_NEARFULL           = 12;
const LONG EVRW_SUE_LS_FULL              = 13;
const LONG EVRW_SUE_DS_NOCARD            = 21;
const LONG EVRW_SUE_DS_DETECTED           = 22;
const LONG EVRW_SUE_DS_ENTERED           = 23;
const LONG EVRW_SUE_DS_CAPTURED           = 24;


////////////////////////////////////////////////////////////////////////
// "TransitionEvent" Event: "EventNumber" Parameter Constants
////////////////////////////////////////////////////////////////////////

const LONG EVRW_TE_NOTIFY_TOUCH_RETRY             =  1;
const LONG EVRW_TE_NOTIFY_TOUCH_RETRY_CANCELABLE    =  2;
const LONG EVRW_TE_CONFIRM_TOUCH_RETRY            =  3;
const LONG EVRW_TE_CONFIRM_CANCEL                =  4;
const LONG EVRW_TE_NOTIFY_INVALID_OPERATION        =  5;
const LONG EVRW_TE_CONFIRM_INVALID_OPERATION       =  6;
const LONG EVRW_TE_CONFIRM_REMAINDER_SUBTRACTION    =  7;
const LONG EVRW_TE_CONFIRM_CENTER_CHECK           =  8;
const LONG EVRW_TE_CONFIRM_TOUCH_TIMEOUT           =  9;
```

```
const LONG EVRW_TE_CONFIRM_AUTO_CHARGE             = 10;
const LONG EVRW_TE_NOTIFY_CAPTURE_CARD             = 11;
const LONG EVRW_TE_NOTIFY_PIN                     = 12;
const LONG EVRW_TE_NOTIFY_CENTER_CHECK             = 13;
const LONG EVRW_TE_NOTIFY_COMPLETE                 = 14;
const LONG EVRW_TE_NOTIFY_TOUCH                   = 15;
const LONG EVRW_TE_NOTIFY_BUSY                    = 16;
const LONG EVRW_TE_CONFIRM_CENTER_CHECK_COMPLETE    = 17;
const LONG EVRW_TE_CONFIRM_SELECT                  = 18;
const LONG EVRW_TE_NOTIFY_LOCK                    = 19;
const LONG EVRW_TE_NOTIFY_CENTER_CHECK_COMPLETE     = 20;
const LONG EVRW_TE_CONFIRM_PIN_ENTRY_BY_OUTER_PINPAD = 21;
const LONG EVRW_TE_NOTIFY_PROGRESS_1_TO_100         = 22; // (added in
1.14.1)
const LONG EVRW_TE_CONFIRM_DEVICE_DATA              = 23; // (added in
1.14.1)
const LONG EVRW_TE_CONFIRM_SEARCH_TABLE             = 24; // (added in
1.15)
const LONG EVRW_TE_CONFIRM_PAYMENT_CONDITION        = 25; // (added in
1.15)
const LONG EVRW_TE_CONFIRM_AUTHORIZE                = 26; // (added in
1.15)
const LONG EVRW_TE_NOTIFY_CHECK_CARD                = 27; // (added in
1.15)
const LONG EVRW_TE_NOTIFY_SELECT_PAYMENT_CONDITION   = 28; // (added
in 1.15)


////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_EVRW_NOCARD              = 201;
const LONG OPOS_EVRW_RELEASE             = 202;
const LONG OPOS_EVRW_CENTERERROR          = 203;
const LONG OPOS_EVRW_COMMANDERROR         = 204;
const LONG OPOS_EVRW_RESET               = 205;
const LONG OPOS_EVRW_COMMUNICATIONERROR    = 206;
const LONG OPOS_EVRW_LOGOVERFLOW          = 207;
const LONG OPOS_EVRW_DAILYLOGOVERFLOW      = 208;
const LONG OPOS_EVRW_DEFICIENT           = 209;
const LONG OPOS_EVRW_OVERDEPOSIT          = 210;


#endif               // !defined(OPOSEVRW_H)
```

OposFptr.h :　フィスカルプリンタヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposFptr.h
//
//  Fiscal Printer header file for OPOS Applications.
//
// Modification history
// ------------------------------------------------------------------
// 1998-03-06 OPOS Release 1.3                                  PDU
// 2001-07-15 OPOS Release 1.6                                  TNN
//  Add values for all 1.6 added properties and method
//  parameters
// 2004-03-22 OPOS Release 1.8                                  CRM
//  Add more values for StatusUpdateEvent.
// 2007-01-30 OPOS Release 1.11                                 CRM
//  Add values for 1.11.
// 2008-08-30 OPOS Release 1.12                                 CRM
//  Add values for ActualCurrency and CountryCode.
// 2020-06-01 OPOS Release 1.15                                 KF
//  Add values for CountryCode and DateType.
// 2025-09-17 OPOS Release 1.15.1                               KF
//  Add value for GetData.
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSFPTR_H)
#define     OPOSFPTR_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// Fiscal Printer Station Constants
////////////////////////////////////////////////////////////////////

const LONG FPTR_S_JOURNAL                = 1;
const LONG FPTR_S_RECEIPT                = 2;
const LONG FPTR_S_SLIP                   = 4;

const LONG FPTR_S_JOURNAL_RECEIPT          = 3;


////////////////////////////////////////////////////////////////////
// "ActualCurrency" Property Constants
////////////////////////////////////////////////////////////////////

const LONG FPTR_AC_BRC                   =  1;
const LONG FPTR_AC_BGL                   =  2;
const LONG FPTR_AC_EUR                   =  3;
const LONG FPTR_AC_GRD                   =  4;
const LONG FPTR_AC_HUF                   =  5;
const LONG FPTR_AC_ITL                   =  6;
const LONG FPTR_AC_PLZ                   =  7;
const LONG FPTR_AC_ROL                   =  8;
const LONG FPTR_AC_RUR                   =  9;
const LONG FPTR_AC_TRL                   =  10;
const LONG FPTR_AC_CZK                   =  11;  // (added in 1.11)
const LONG FPTR_AC_UAH                   =  12;  // (added in 1.11)
const LONG FPTR_AC_SEK                   =  13;  // (added in 1.12)
const LONG FPTR_AC_OTHER                  =  200; // (added in 1.11)
```

```
////////////////////////////////////////////////////////////////////////
// "ContractorId" Property Constants
////////////////////////////////////////////////////////////////////////

const LONG FPTR_CID_FIRST                = 1;
const LONG FPTR_CID_SECOND               = 2;
const LONG FPTR_CID_SINGLE               = 3;


////////////////////////////////////////////////////////////////////////
// "CountryCode" Property Constants
////////////////////////////////////////////////////////////////////////

const LONG FPTR_CC_BRAZIL                = 0x00000001;
const LONG FPTR_CC_GREECE                = 0x00000002;
const LONG FPTR_CC_HUNGARY               = 0x00000004;
const LONG FPTR_CC_ITALY                 = 0x00000008;
const LONG FPTR_CC_POLAND                = 0x00000010;
const LONG FPTR_CC_TURKEY                = 0x00000020;
const LONG FPTR_CC_RUSSIA                = 0x00000040;
const LONG FPTR_CC_BULGARIA              = 0x00000080;
const LONG FPTR_CC_ROMANIA               = 0x00000100;
const LONG FPTR_CC_CZECH_REPUBLIC        = 0x00000200; // (added in
1.11)
const LONG FPTR_CC_UKRAINE               = 0x00000400; // (added in
1.11)
const LONG FPTR_CC_SWEDEN                = 0x00000800; // (added in
1.12)
const LONG FPTR_CC_GERMANY               = 0x00001000; // (added in
1.15)
const LONG FPTR_CC_OTHER                 = 0x40000000; // (added in
1.11)


////////////////////////////////////////////////////////////////////////
// "DateType" Property Constants
////////////////////////////////////////////////////////////////////////

const LONG FPTR_DT_CONF                  = 1;
const LONG FPTR_DT_EOD                   = 2;
const LONG FPTR_DT_RESET                 = 3;
const LONG FPTR_DT_RTC                   = 4;
const LONG FPTR_DT_VAT                   = 5;
const LONG FPTR_DT_START                 = 6; // (added in 1.11)
const LONG FPTR_DT_TICKET_START          = 7; // (added in 1.15)
const LONG FPTR_DT_TICKET_END            = 8; // (added in 1.15)


////////////////////////////////////////////////////////////////////////
// "ErrorLevel" Property Constants
////////////////////////////////////////////////////////////////////////

const LONG FPTR_EL_NONE                  = 1;
const LONG FPTR_EL_RECOVERABLE           = 2;
const LONG FPTR_EL_FATAL                 = 3;
const LONG FPTR_EL_BLOCKED               = 4;


////////////////////////////////////////////////////////////////////////
// "ErrorState", "PrinterState" Property Constants
////////////////////////////////////////////////////////////////////////

const LONG FPTR_PS_MONITOR               = 1;
const LONG FPTR_PS_FISCAL_RECEIPT        = 2;
const LONG FPTR_PS_FISCAL_RECEIPT_TOTAL  = 3;
```

```
const LONG FPTR_PS_FISCAL_RECEIPT_ENDING   =  4;
const LONG FPTR_PS_FISCAL_DOCUMENT         =  5;
const LONG FPTR_PS_FIXED_OUTPUT            =  6;
const LONG FPTR_PS_ITEM_LIST               =  7;
const LONG FPTR_PS_LOCKED                  =  8;
const LONG FPTR_PS_NONFISCAL               =  9;
const LONG FPTR_PS_REPORT                  = 10;


////////////////////////////////////////////////////////////////////
// "FiscalReceiptStation" Property Constants
////////////////////////////////////////////////////////////////////

const LONG FPTR_RS_RECEIPT                 =  1;
const LONG FPTR_RS_SLIP                    =  2;


////////////////////////////////////////////////////////////////////
// "FiscalReceiptType" Property Constants
////////////////////////////////////////////////////////////////////

const LONG FPTR_RT_CASH_IN                 =  1;
const LONG FPTR_RT_CASH_OUT                =  2;
const LONG FPTR_RT_GENERIC                 =  3;
const LONG FPTR_RT_SALES                   =  4;
const LONG FPTR_RT_SERVICE                 =  5;
const LONG FPTR_RT_SIMPLE_INVOICE          =  6;
const LONG FPTR_RT_REFUND                  =  7; // (added in 1.11)

////////////////////////////////////////////////////////////////////
// "MessageType" Property Constants
////////////////////////////////////////////////////////////////////

const LONG FPTR_MT_ADVANCE                 =  1;
const LONG FPTR_MT_ADVANCE_PAID            =  2;
const LONG FPTR_MT_AMOUNT_TO_BE_PAID       =  3;
const LONG FPTR_MT_AMOUNT_TO_BE_PAID_BACK  =  4;
const LONG FPTR_MT_CARD                    =  5;
const LONG FPTR_MT_CARD_NUMBER             =  6;
const LONG FPTR_MT_CARD_TYPE               =  7;
const LONG FPTR_MT_CASH                    =  8;
const LONG FPTR_MT_CASHIER                 =  9;
const LONG FPTR_MT_CASH_REGISTER_NUMBER    = 10;
const LONG FPTR_MT_CHANGE                  = 11;
const LONG FPTR_MT_CHEQUE                  = 12;
const LONG FPTR_MT_CLIENT_NUMBER           = 13;
const LONG FPTR_MT_CLIENT_SIGNATURE        = 14;
const LONG FPTR_MT_COUNTER_STATE           = 15;
const LONG FPTR_MT_CREDIT_CARD             = 16;
const LONG FPTR_MT_CURRENCY                = 17;
const LONG FPTR_MT_CURRENCY_VALUE          = 18;
const LONG FPTR_MT_DEPOSIT                 = 19;
const LONG FPTR_MT_DEPOSIT_RETURNED        = 20;
const LONG FPTR_MT_DOT_LINE                = 21;
const LONG FPTR_MT_DRIVER_NUMB             = 22;
const LONG FPTR_MT_EMPTY_LINE              = 23;
const LONG FPTR_MT_FREE_TEXT               = 24;
const LONG FPTR_MT_FREE_TEXT_WITH_DAY_LIMIT = 25;
const LONG FPTR_MT_GIVEN_DISCOUNT          = 26;
const LONG FPTR_MT_LOCAL_CREDIT            = 27;
const LONG FPTR_MT_MILEAGE_KM              = 28;
const LONG FPTR_MT_NOTE                    = 29;
const LONG FPTR_MT_PAID                    = 30;
const LONG FPTR_MT_PAY_IN                  = 31;
const LONG FPTR_MT_POINT_GRANTED           = 32;
const LONG FPTR_MT_POINTS_BONUS            = 33;
```

```
const LONG FPTR_MT_POINTS_RECEIPT         =  34;
const LONG FPTR_MT_POINTS_TOTAL           =  35;
const LONG FPTR_MT_PROFITED               =  36;
const LONG FPTR_MT_RATE                   =  37;
const LONG FPTR_MT_REGISTER_NUMB          =  38;
const LONG FPTR_MT_SHIFT_NUMBER           =  39;
const LONG FPTR_MT_STATE_OF_AN_ACCOUNT    =  40;
const LONG FPTR_MT_SUBSCRIPTION           =  41;
const LONG FPTR_MT_TABLE                  =  42;
const LONG FPTR_MT_THANK_YOU_FOR_LOYALTY  =  43;
const LONG FPTR_MT_TRANSACTION_NUMB       =  44;
const LONG FPTR_MT_VALID_TO               =  45;
const LONG FPTR_MT_VOUCHER                =  46;
const LONG FPTR_MT_VOUCHER_PAID           =  47;
const LONG FPTR_MT_VOUCHER_VALUE          =  48;
const LONG FPTR_MT_WITH_DISCOUNT          =  49;
const LONG FPTR_MT_WITHOUT_UPLIFT         =  50;


////////////////////////////////////////////////////////////////////////
// "SlipSelection" Property Constants
////////////////////////////////////////////////////////////////////////

const LONG FPTR_SS_FULL_LENGTH            =  1;
const LONG FPTR_SS_VALIDATION             =  2;


////////////////////////////////////////////////////////////////////////
// "TotalizerType" Property Constants
////////////////////////////////////////////////////////////////////////

const LONG FPTR_TT_DOCUMENT               =  1;
const LONG FPTR_TT_DAY                    =  2;
const LONG FPTR_TT_RECEIPT                =  3;
const LONG FPTR_TT_GRAND                  =  4;


////////////////////////////////////////////////////////////////////////
// "GetData" Method Constants
////////////////////////////////////////////////////////////////////////

// - "DataItem" Parameter.
const LONG FPTR_GD_FIRMWARE               = 10; // 1:
IdentificationData.
const LONG FPTR_GD_PRINTER_ID             =  9; // 1:
IdentificationData.
const LONG FPTR_GD_VAT_ID_LIST            = 29; // 1:
IdentificationData. (added in 1.15.1)

const LONG FPTR_GD_CURRENT_TOTAL          =  1; // 2: Total.
const LONG FPTR_GD_DAILY_TOTAL            =  2; // 2: Total.
const LONG FPTR_GD_GRAND_TOTAL            =  8; // 2: Total.
const LONG FPTR_GD_MID_VOID               =  6; // 2: Total.
const LONG FPTR_GD_NOT_PAID               =  5; // 2: Total.
const LONG FPTR_GD_RECEIPT_NUMBER         =  3; // 2: Total.
const LONG FPTR_GD_REFUND                 =  4; // 2: Total.
const LONG FPTR_GD_REFUND_VOID            = 12; // 2: Total.

const LONG FPTR_GD_NUMB_CONFIG_BLOCK      = 13; // 3:
FiscalMemoryCount.
const LONG FPTR_GD_NUMB_CURRENCY_BLOCK    = 14; // 3:
FiscalMemoryCount.
const LONG FPTR_GD_NUMB_HDR_BLOCK         = 15; // 3:
FiscalMemoryCount.
const LONG FPTR_GD_NUMB_RESET_BLOCK       = 16; // 3:
FiscalMemoryCount.
```

```
const LONG FPTR_GD_NUMB_VAT_BLOCK         = 17; // 3:
FiscalMemoryCount.

const LONG FPTR_GD_FISCAL_DOC             = 18; // 4: Counter.
const LONG FPTR_GD_FISCAL_DOC_VOID         = 19; // 4: Counter.
const LONG FPTR_GD_FISCAL_REC             = 20; // 4: Counter.
const LONG FPTR_GD_FISCAL_REC_VOID         = 21; // 4: Counter.
const LONG FPTR_GD_NONFISCAL_DOC           = 22; // 4: Counter.
const LONG FPTR_GD_NONFISCAL_DOC_VOID       = 23; // 4: Counter.
const LONG FPTR_GD_NONFISCAL_REC           = 24; // 4: Counter.
const LONG FPTR_GD_RESTART                = 11; // 4: Counter.
const LONG FPTR_GD_SIMP_INVOICE            = 25; // 4: Counter.
const LONG FPTR_GD_Z_REPORT                =  7; // 4: Counter.

const LONG FPTR_GD_TENDER                 = 26; // 5:
FixedFiscalPrinterText.

const LONG FPTR_GD_LINECOUNT              = 27; // 6: Linecounter.

const LONG FPTR_GD_DESCRIPTION_LENGTH      = 28; // 7:
DescriptionLength.

// - "OptArgs" Parameter, when "DataItem" is FPTR_GD_TENDER.
const LONG FPTR_PDL_CASH                   =  1;
const LONG FPTR_PDL_CHEQUE                 =  2;
const LONG FPTR_PDL_CHITTY                 =  3;
const LONG FPTR_PDL_COUPON                 =  4;
const LONG FPTR_PDL_CURRENCY               =  5;
const LONG FPTR_PDL_DRIVEN_OFF             =  6;
const LONG FPTR_PDL_EFT_IMPRINTER          =  7;
const LONG FPTR_PDL_EFT_TERMINAL           =  8;
const LONG FPTR_PDL_TERMINAL_IMPRINTER      =  9;
const LONG FPTR_PDL_FREE_GIFT              = 10;
const LONG FPTR_PDL_GIRO                   = 11;
const LONG FPTR_PDL_HOME                   = 12;
const LONG FPTR_PDL_IMPRINTER_WITH_ISSUER   = 13;
const LONG FPTR_PDL_LOCAL_ACCOUNT          = 14;
const LONG FPTR_PDL_LOCAL_ACCOUNT_CARD      = 15;
const LONG FPTR_PDL_PAY_CARD               = 16;
const LONG FPTR_PDL_PAY_CARD_MANUAL         = 17;
const LONG FPTR_PDL_PREPAY                 = 18;
const LONG FPTR_PDL_PUMP_TEST              = 19;
const LONG FPTR_PDL_SHORT_CREDIT           = 20;
const LONG FPTR_PDL_STAFF                  = 21;
const LONG FPTR_PDL_VOUCHER                = 22;

// - "OptArgs" Parameter, when "DataItem" is FPTR_GD_LINECOUNT.
const LONG FPTR_LC_ITEM                    =  1;
const LONG FPTR_LC_ITEM_VOID               =  2;
const LONG FPTR_LC_DISCOUNT                =  3;
const LONG FPTR_LC_DISCOUNT_VOID           =  4;
const LONG FPTR_LC_SURCHARGE               =  5;
const LONG FPTR_LC_SURCHARGE_VOID          =  6;
const LONG FPTR_LC_REFUND                  =  7;
const LONG FPTR_LC_REFUND_VOID             =  8;
const LONG FPTR_LC_SUBTOTAL_DISCOUNT        =  9;
const LONG FPTR_LC_SUBTOTAL_DISCOUNT_VOID    = 10;
const LONG FPTR_LC_SUBTOTAL_SURCHARGE       = 11;
const LONG FPTR_LC_SUBTOTAL_SURCHARGE_VOID   = 12;
const LONG FPTR_LC_COMMENT                 = 13;
const LONG FPTR_LC_SUBTOTAL                = 14;
const LONG FPTR_LC_TOTAL                   = 15;

// - "OptArgs" Parameter, when "DataItem" is
FPTR_GD_DESCRIPTION_LENGTH.
const LONG FPTR_DL_ITEM                    =  1;
```

```
const LONG FPTR_DL_ITEM_ADJUSTMENT          =  2;
const LONG FPTR_DL_ITEM_FUEL                =  3;
const LONG FPTR_DL_ITEM_FUEL_VOID           =  4;
const LONG FPTR_DL_NOT_PAID                 =  5;
const LONG FPTR_DL_PACKAGE_ADJUSTMENT       =  6;
const LONG FPTR_DL_REFUND                   =  7;
const LONG FPTR_DL_REFUND_VOID              =  8;
const LONG FPTR_DL_SUBTOTAL_ADJUSTMENT      =  9;
const LONG FPTR_DL_TOTAL                    = 10;
const LONG FPTR_DL_VOID                     = 11;
const LONG FPTR_DL_VOID_ITEM                = 12;


/////////////////////////////////////////////////////////////////////
// "GetTotalizer" Method Constants
/////////////////////////////////////////////////////////////////////

const LONG FPTR_GT_GROSS                  =  1;
const LONG FPTR_GT_NET                    =  2;
const LONG FPTR_GT_DISCOUNT               =  3;
const LONG FPTR_GT_DISCOUNT_VOID          =  4;
const LONG FPTR_GT_ITEM                   =  5;
const LONG FPTR_GT_ITEM_VOID              =  6;
const LONG FPTR_GT_NOT_PAID               =  7;
const LONG FPTR_GT_REFUND                 =  8;
const LONG FPTR_GT_REFUND_VOID            =  9;
const LONG FPTR_GT_SUBTOTAL_DISCOUNT      = 10;
const LONG FPTR_GT_SUBTOTAL_DISCOUNT_VOID = 11;
const LONG FPTR_GT_SUBTOTAL_SURCHARGES    = 12;
const LONG FPTR_GT_SUBTOTAL_SURCHARGES_VOID = 13;
const LONG FPTR_GT_SURCHARGE              = 14;
const LONG FPTR_GT_SURCHARGE_VOID         = 15;
const LONG FPTR_GT_VAT                    = 16;
const LONG FPTR_GT_VAT_CATEGORY           = 17;


/////////////////////////////////////////////////////////////////////
// "AdjustmentType" arguments in diverse methods
/////////////////////////////////////////////////////////////////////

const LONG FPTR_AT_AMOUNT_DISCOUNT          =  1;
const LONG FPTR_AT_AMOUNT_SURCHARGE         =  2;
const LONG FPTR_AT_PERCENTAGE_DISCOUNT      =  3;
const LONG FPTR_AT_PERCENTAGE_SURCHARGE     =  4;
const LONG FPTR_AT_COUPON_AMOUNT_DISCOUNT   =  5; // (added in 1.11)
const LONG FPTR_AT_COUPON_PERCENTAGE_DISCOUNT =  6; // (added in
1.11)


/////////////////////////////////////////////////////////////////////
// "ReportType" argument in "PrintReport" method
/////////////////////////////////////////////////////////////////////

const LONG FPTR_RT_ORDINAL                =  1;
const LONG FPTR_RT_DATE                   =  2;
const LONG FPTR_RT_EOD_ORDINAL            =  3; // (added in 1.11)

/////////////////////////////////////////////////////////////////////
// "NewCurrency" argument in "SetCurrency" method
/////////////////////////////////////////////////////////////////////

const LONG FPTR_SC_EURO                   =  1;


/////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constants
```

```
////////////////////////////////////////////////////////////////////

const LONG FPTR_SUE_COVER_OPEN           =  11;
const LONG FPTR_SUE_COVER_OK             =  12;
const LONG FPTR_SUE_JRN_COVER_OPEN        =  60;  // (added in 1.8)
const LONG FPTR_SUE_JRN_COVER_OK          =  61;  // (added in 1.8)
const LONG FPTR_SUE_REC_COVER_OPEN        =  62;  // (added in 1.8)
const LONG FPTR_SUE_REC_COVER_OK          =  63;  // (added in 1.8)
const LONG FPTR_SUE_SLP_COVER_OPEN        =  64;  // (added in 1.8)
const LONG FPTR_SUE_SLP_COVER_OK          =  65;  // (added in 1.8)

const LONG FPTR_SUE_JRN_EMPTY            =  21;
const LONG FPTR_SUE_JRN_NEAREMPTY         =  22;
const LONG FPTR_SUE_JRN_PAPEROK          =  23;

const LONG FPTR_SUE_REC_EMPTY            =  24;
const LONG FPTR_SUE_REC_NEAREMPTY         =  25;
const LONG FPTR_SUE_REC_PAPEROK          =  26;

const LONG FPTR_SUE_SLP_EMPTY            =  27;
const LONG FPTR_SUE_SLP_NEAREMPTY         =  28;
const LONG FPTR_SUE_SLP_PAPEROK          =  29;

const LONG FPTR_SUE_IDLE                =1001;


////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_EFPTR_COVER_OPEN             = 201;
const LONG OPOS_EFPTR_JRN_EMPTY              = 202;
const LONG OPOS_EFPTR_REC_EMPTY              = 203;
const LONG OPOS_EFPTR_SLP_EMPTY              = 204;
const LONG OPOS_EFPTR_SLP_FORM               = 205;
const LONG OPOS_EFPTR_MISSING_DEVICES         = 206;
const LONG OPOS_EFPTR_WRONG_STATE            = 207;
const LONG OPOS_EFPTR_TECHNICAL_ASSISTANCE     = 208;
const LONG OPOS_EFPTR_CLOCK_ERROR            = 209;
const LONG OPOS_EFPTR_FISCAL_MEMORY_FULL      = 210;
const LONG OPOS_EFPTR_FISCAL_MEMORY_DISCONNECTED = 211;
const LONG OPOS_EFPTR_FISCAL_TOTALS_ERROR      = 212;
const LONG OPOS_EFPTR_BAD_ITEM_QUANTITY       = 213;
const LONG OPOS_EFPTR_BAD_ITEM_AMOUNT         = 214;
const LONG OPOS_EFPTR_BAD_ITEM_DESCRIPTION     = 215;
const LONG OPOS_EFPTR_RECEIPT_TOTAL_OVERFLOW    = 216;
const LONG OPOS_EFPTR_BAD_VAT                = 217;
const LONG OPOS_EFPTR_BAD_PRICE              = 218;
const LONG OPOS_EFPTR_BAD_DATE               = 219;
const LONG OPOS_EFPTR_NEGATIVE_TOTAL          = 220;
const LONG OPOS_EFPTR_WORD_NOT_ALLOWED        = 221;
const LONG OPOS_EFPTR_BAD_LENGTH             = 222;
const LONG OPOS_EFPTR_MISSING_SET_CURRENCY     = 223;
const LONG OPOS_EFPTR_DAY_END_REQUIRED        = 224; // (added in
1.11)


#endif               // !defined(OPOSFPTR_H)
```

OposGate.h :　ゲートヘッダーファイル

```
///////////////////////////////////////////////////////////////////////
//
// OposGate.h
//
//  Gate header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 2008-08-30 OPOS Release 1.12                                CRM
//
///////////////////////////////////////////////////////////////////////

#if !defined(OPOSGATE_H)
#define     OPOSGATE_H


#include "Opos.h"


///////////////////////////////////////////////////////////////////////
// "GateStatus" Property Constants
///////////////////////////////////////////////////////////////////////

const LONG GATE_GS_CLOSED            = 1;
const LONG GATE_GS_OPEN              = 2;
const LONG GATE_GS_BLOCKED           = 3;
const LONG GATE_GS_MALFUNCTION       = 4;


///////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constants
///////////////////////////////////////////////////////////////////////

const LONG GATE_SUE_CLOSED           = 11;
const LONG GATE_SUE_OPEN             = 12;
const LONG GATE_SUE_BLOCKED          = 13;
const LONG GATE_SUE_MALFUNCTION      = 14;


#endif              // !defined(OPOSGATE_H)
```

OposGctl.h :　ジェスチャーコントロールヘッダーファイル

```
//////////////////////////////////////////////////////////////////////
//
// OposGctl.h
//
//  Gesture Control header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 2021-01-26 OPOS Release 1.16                                 KF
//
//////////////////////////////////////////////////////////////////////

#if !defined(OPOSGCTL_H)
#define     OPOSGCTL_H


#include "Opos.h"


//////////////////////////////////////////////////////////////////////
// "CapStorage", "Storage" Property Constants
//////////////////////////////////////////////////////////////////////

const LONG GCTL_CST_HOST_ONLY      = 1;
const LONG GCTL_CST_HARDTOTALS_ONLY = 2;
const LONG GCTL_CST_ALL            = 3;

const LONG GCTL_ST_HOST            = 1;
const LONG GCTL_ST_HARDTOTALS      = 2;
const LONG GCTL_ST_HOST_HARDTOTALS  = 3;


//////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constants
//////////////////////////////////////////////////////////////////////

const LONG GCTL_SUE_START_MOTION    = 11;
const LONG GCTL_SUE_STOP_MOTION     = 12;


//////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
//////////////////////////////////////////////////////////////////////

const LONG OPOS_EGCTL_NOROOM       = 201; // (Several)


#endif               // !defined(OPOSGCTL_H)
```

OposGdsp.h： グラフィックディスプレイヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposGdsp.h
//
//  Graphic Display header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 2021-01-26 OPOS Release 1.16                                  KF
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSGDSP_H)
#define     OPOSGDSP_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "CapStorage", "Storage" Property Constants
////////////////////////////////////////////////////////////////////

const LONG GDSP_CST_HOST_ONLY      = 1;
const LONG GDSP_CST_HARDTOTALS_ONLY = 2;
const LONG GDSP_CST_ALL            = 3;

const LONG GDSP_ST_HOST            = 1;
const LONG GDSP_ST_HARDTOTALS      = 2;
const LONG GDSP_ST_HOST_HARDTOTALS  = 3;


////////////////////////////////////////////////////////////////////
// "DisplayMode" Property Constants
////////////////////////////////////////////////////////////////////

const LONG GDSP_DMODE_HIDDEN       = 1;
const LONG GDSP_DMODE_IMAGE_FIT    = 2;
const LONG GDSP_DMODE_IMAGE_FILL   = 3;
const LONG GDSP_DMODE_IMAGE_CENTER  = 4;
const LONG GDSP_DMODE_VIDEO_NORMAL  = 5;
const LONG GDSP_DMODE_VIDEO_FULL   = 6;
const LONG GDSP_DMODE_WEB          = 7;


////////////////////////////////////////////////////////////////////
// "LoadStatus" Property Constants
////////////////////////////////////////////////////////////////////

const LONG GDSP_LSTATUS_START      = 1;
const LONG GDSP_LSTATUS_FINISH     = 2;
const LONG GDSP_LSTATUS_CANCEL     = 3;


////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG GDSP_SUE_START_IMAGE_LOAD   = 11;
const LONG GDSP_SUE_END_IMAGE_LOAD     = 12;
const LONG GDSP_SUE_START_LOAD_WEBPAGE  = 21;
const LONG GDSP_SUE_FINISH_LOAD_WEBPAGE = 22;
```

```
const LONG GDSP_SUE_CANCEL_LOAD_WEBPAGE = 23;
const LONG GDSP_SUE_START_PLAY_VIDEO    = 31;
const LONG GDSP_SUE_STOP_PLAY_VIDEO     = 32;


////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_EGDSP_NOROOM       = 201; // (Several)


#endif               // !defined(OPOSGDSP_H)
```

OposImg.h :　イメージスキャナヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposImg.h
//
//   Image Scanner header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 2007-01-30 OPOS Release 1.11                                  CRM
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSIMG_H)
#define     OPOSIMG_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "FrameType" Property Constants
////////////////////////////////////////////////////////////////////

const LONG IMG_FRAME_STILL            = 1;
const LONG IMG_FRAME_VIDEO            = 2;


////////////////////////////////////////////////////////////////////
// "ImageMode" Property Constants
////////////////////////////////////////////////////////////////////

const LONG IMG_ALL                   = 0xFF;
const LONG IMG_DECODE_ONLY            = 0x01;
const LONG IMG_STILL_ONLY             = 0x02;
const LONG IMG_STILL_DECODE           = 0x03;
const LONG IMG_VIDEO_DECODE           = 0x05;
const LONG IMG_VIDEO_STILL            = 0x06;


////////////////////////////////////////////////////////////////////
// "ImageQuality" Property Constants
////////////////////////////////////////////////////////////////////

const LONG IMG_QUAL_LOW               = 0x01;
const LONG IMG_QUAL_MED               = 0x02;
const LONG IMG_QUAL_HIGH              = 0x03;


////////////////////////////////////////////////////////////////////
// "ImageType" Property Constants
////////////////////////////////////////////////////////////////////

const LONG IMG_TYP_BMP                = 1;
const LONG IMG_TYP_JPEG               = 2;
const LONG IMG_TYP_GIF                = 3;
const LONG IMG_TYP_PNG                = 4;
const LONG IMG_TYP_TIFF               = 5;


#endif              // !defined(OPOSIMG_H)
```

OposIrcg.h :　個体認識ヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposIrcg.h
//
//   Individual Recognition header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------
// 2021-01-26 OPOS Release 1.16                              KF
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSIRCG_H)
#define     OPOSIRCG_H


#include "Opos.h"


#endif               // !defined(OPOSIRCG_H)
```

OposItem.h：　アイテムディスペンサヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposItem.h
//
//   Item Dispenser header file for OPOS Applications.
//
// Modification history
// ------------------------------------------------------------------
// 2008-08-30 OPOS Release 1.12                              CRM
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSITEM_H)
#define     OPOSITEM_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "DispenserStatus" Property Constants
////////////////////////////////////////////////////////////////////

const LONG ITEM_DS_OK               = 1;
const LONG ITEM_DS_EMPTY             = 2;
const LONG ITEM_DS_NEAREMPTY          = 3;
const LONG ITEM_DS_JAM               = 4;


////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG ITEM_SUE_OK               = 11;
const LONG ITEM_SUE_EMPTY             = 12;
const LONG ITEM_SUE_NEAREMPTY         = 13;
const LONG ITEM_SUE_JAM              = 14;


#endif               // !defined(OPOSITEM_H)
```

OposKbd.h :　POSキーボードヘッダーファイル

```
///////////////////////////////////////////////////////////////////////
//
// OposKbd.h
//
//  POS Keyboard header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 1996-04-22 OPOS Release 1.1                                  CRM
// 1997-06-04 OPOS Release 1.2                                  CRM
//  Add "EventTypes" and "POSKeyEventType" values.
//
///////////////////////////////////////////////////////////////////////

#if !defined(OPOSKBD_H)
#define     OPOSKBD_H


#include "Opos.h"


///////////////////////////////////////////////////////////////////////
// "EventTypes" Property Constants
///////////////////////////////////////////////////////////////////////

const LONG KBD_ET_DOWN        =  1;
const LONG KBD_ET_DOWN_UP     =  2;


///////////////////////////////////////////////////////////////////////
// "POSKeyEventType" Property Constants
///////////////////////////////////////////////////////////////////////

const LONG KBD_KET_KEYDOWN    =  1;
const LONG KBD_KET_KEYUP      =  2;


#endif               // !defined(OPOSKBD_H)
```

OposLgt.h :　ライトヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposLights.h
//
//  Lights header file for OPOS Applications.
//
// Modification history
// ------------------------------------------------------------------
// 2008-08-30 OPOS Release 1.12                              CRM
// 2021-01-26 OPOS Release 1.16                              KF
//  Added CapPattern, Pattern constants.
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSLGT_H)
#define     OPOSLGT_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "CapAlarm" Property Constants
////////////////////////////////////////////////////////////////////

const LONG LGT_ALARM_NOALARM                = 0x00000001;
const LONG LGT_ALARM_SLOW                   = 0x00000010;
const LONG LGT_ALARM_MEDIUM                 = 0x00000020;
const LONG LGT_ALARM_FAST                   = 0x00000040;
const LONG LGT_ALARM_CUSTOM1                = 0x00010000;
const LONG LGT_ALARM_CUSTOM2                = 0x00020000;


////////////////////////////////////////////////////////////////////
// "CapColor" Property Constants
////////////////////////////////////////////////////////////////////

const LONG LGT_COLOR_PRIMARY                = 0x00000001;
const LONG LGT_COLOR_CUSTOM1                = 0x00010000;
const LONG LGT_COLOR_CUSTOM2                = 0x00020000;
const LONG LGT_COLOR_CUSTOM3                = 0x00040000;
const LONG LGT_COLOR_CUSTOM4                = 0x00080000;
const LONG LGT_COLOR_CUSTOM5                = 0x00100000;


////////////////////////////////////////////////////////////////////
// "CapPattern" Property Constants
// "SwitchOnPattern" Method: "Pattern" Paremeter Constants
////////////////////////////////////////////////////////////////////

const LONG LGT_PATTERN_NOPATTERN            = 0x00000000;
const LONG LGT_PATTERN_CUSTOM1              = 0x00000001;
const LONG LGT_PATTERN_CUSTOM2              = 0x00000002;
const LONG LGT_PATTERN_CUSTOM3              = 0x00000004;
const LONG LGT_PATTERN_CUSTOM4              = 0x00000008;
const LONG LGT_PATTERN_CUSTOM5              = 0x00000010;
const LONG LGT_PATTERN_CUSTOM6              = 0x00000020;
const LONG LGT_PATTERN_CUSTOM7              = 0x00000040;
const LONG LGT_PATTERN_CUSTOM8              = 0x00000080;
const LONG LGT_PATTERN_CUSTOM9              = 0x00000100;
const LONG LGT_PATTERN_CUSTOM10             = 0x00000200;
const LONG LGT_PATTERN_CUSTOM11             = 0x00000400;
```

```
const LONG LGT_PATTERN_CUSTOM12             = 0x00000800;
const LONG LGT_PATTERN_CUSTOM13             = 0x00001000;
const LONG LGT_PATTERN_CUSTOM14             = 0x00002000;
const LONG LGT_PATTERN_CUSTOM15             = 0x00004000;
const LONG LGT_PATTERN_CUSTOM16             = 0x00008000;
const LONG LGT_PATTERN_CUSTOM17             = 0x00010000;
const LONG LGT_PATTERN_CUSTOM18             = 0x00020000;
const LONG LGT_PATTERN_CUSTOM19             = 0x00040000;
const LONG LGT_PATTERN_CUSTOM20             = 0x00080000;
const LONG LGT_PATTERN_CUSTOM21             = 0x00100000;
const LONG LGT_PATTERN_CUSTOM22             = 0x00200000;
const LONG LGT_PATTERN_CUSTOM23             = 0x00400000;
const LONG LGT_PATTERN_CUSTOM24             = 0x00800000;
const LONG LGT_PATTERN_CUSTOM25             = 0x01000000;
const LONG LGT_PATTERN_CUSTOM26             = 0x02000000;
const LONG LGT_PATTERN_CUSTOM27             = 0x04000000;
const LONG LGT_PATTERN_CUSTOM28             = 0x08000000;
const LONG LGT_PATTERN_CUSTOM29             = 0x10000000;
const LONG LGT_PATTERN_CUSTOM30             = 0x20000000;
const LONG LGT_PATTERN_CUSTOM31             = 0x40000000;
const LONG LGT_PATTERN_CUSTOM32             = 0x80000000;


#endif              // !defined(OPOSLGT_H)
```

OposLock.h：　キーロックヘッダーファイル

```
///////////////////////////////////////////////////////////////////////
//
// OposLock.h
//
//  Keylock header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 1995-12-08 OPOS Release 1.0                                  CRM
// 2007-01-30 OPOS Release 1.11                                 CRM
//
///////////////////////////////////////////////////////////////////////

#if !defined(OPOSLOCK_H)
#define     OPOSLOCK_H


#include "Opos.h"


///////////////////////////////////////////////////////////////////////
// "KeyPosition" Property Constants
// "WaitForKeylockChange" Method: "KeyPosition" Parameter
// "StatusUpdateEvent" Event: "Data" Parameter
///////////////////////////////////////////////////////////////////////

const LONG LOCK_KP_ANY         = 0;    // WaitForKeylockChange Only
const LONG LOCK_KP_ELECTRONIC   = 0;    // StatusUpdateEvent Only
(added in 1.11)
const LONG LOCK_KP_LOCK        = 1;
const LONG LOCK_KP_NORM        = 2;
const LONG LOCK_KP_SUPR        = 3;


///////////////////////////////////////////////////////////////////////
// "CapKeylockType" Property Constants
//  Added in 1.11.
///////////////////////////////////////////////////////////////////////

const LONG LOCK_KT_STANDARD    = 1;
const LONG LOCK_KT_ELECTRONIC   = 2;


#endif               // !defined(OPOSLOCK_H)
```

OposMicr.h :　MICR(磁気インクリーダ)ヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposMicr.h
//
//  MICR header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------
// 1995-12-08 OPOS Release 1.0                              CRM
// 2002-08-17 OPOS Release 1.7                              CRM
//  Add new ResultCodeExtended constants.
// 2009-10-02 OPOS Release 1.13                             CRM
//  Add CountryCode constants.
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSMICR_H)
#define     OPOSMICR_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "CheckType" Property Constants
////////////////////////////////////////////////////////////////////

const LONG MICR_CT_PERSONAL    =  1;
const LONG MICR_CT_BUSINESS    =  2;
const LONG MICR_CT_UNKNOWN     = 99;


////////////////////////////////////////////////////////////////////
// "CountryCode" Property Constants
////////////////////////////////////////////////////////////////////

const LONG MICR_CC_USA        =  1;
const LONG MICR_CC_CANADA      =  2;
const LONG MICR_CC_MEXICO      =  3;
const LONG MICR_CC_UNKNOWN     = 99;
const LONG MICR_CC_CMC7        =  4; // Added in 1.13
const LONG MICR_CC_OTHER       =  5; // Added in 1.13


////////////////////////////////////////////////////////////////////
/
// "ResultCodeExtended" Property Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_EMICR_NOCHECK      = 201; // EndInsertion
const LONG OPOS_EMICR_CHECK        = 202; // EndRemoval

// The following were added in 1.7
const LONG OPOS_EMICR_BADDATA      = 203;
const LONG OPOS_EMICR_NODATA       = 204;
const LONG OPOS_EMICR_BADSIZE      = 205;
const LONG OPOS_EMICR_JAM          = 206;
const LONG OPOS_EMICR_CHECKDIGIT    = 207;
const LONG OPOS_EMICR_COVEROPEN     = 208;

#endif               // !defined(OPOSMICR_H)
```

OposMotion.h :　モーションセンサヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposMotion.h
//
//  Motion Sensor header file for OPOS Applications.
//
// Modification history
// ------------------------------------------------------------------
// 2002-08-17 OPOS Release 1.7                                CRM
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSMOTION_H)
#define     OPOSMOTION_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// Status Update Event "Status" Constants
////////////////////////////////////////////////////////////////////

const LONG MOTION_M_PRESENT    = 1;
const LONG MOTION_M_ABSENT     = 2;


#endif               // !defined(OPOSMOTION_H)
```

OposMsr.h :　磁気ストライプリーダヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposMsr.h
//
//  Magnetic Stripe Reader header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 1995-12-08 OPOS Release 1.0                                  CRM
// 1997-06-04 OPOS Release 1.2                                  CRM
//  Add ErrorReportingType constants.
// 2000-09-24 OPOS Release 1.5                                  BKS
//  Add constants relating to Track 4 Data.
//   (01-07-15 Added omitted MSR_TR_1_3_4 property)
// 2006-03-15 OPOS Release 1.10                                 CRM
//  Add values for the following properties:
//    CapWritableTracks, EncodingMaxLength, TracksToWrite
// 2008-08-30 OPOS Release 1.12                                 CRM
//  Add values for new properties, plus SUE and RCExt values.
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSMSR_H)
#define     OPOSMSR_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "TracksToRead" Property Constants
////////////////////////////////////////////////////////////////////

const LONG MSR_TR_1           = 1;
const LONG MSR_TR_2           = 2;
const LONG MSR_TR_3           = 4;
const LONG MSR_TR_4           = 8;

const LONG MSR_TR_1_2         = 0x03;
const LONG MSR_TR_1_3         = 0x05;
const LONG MSR_TR_1_4         = 0x09;
const LONG MSR_TR_2_3         = 0x06;
const LONG MSR_TR_2_4         = 0x0A;
const LONG MSR_TR_3_4         = 0x0C;

const LONG MSR_TR_1_2_3       = 0x07;
const LONG MSR_TR_1_2_4       = 0x0B;
const LONG MSR_TR_1_3_4       = 0x0D;
const LONG MSR_TR_2_3_4       = 0x0E;

const LONG MSR_TR_1_2_3_4     = 0x0F;


////////////////////////////////////////////////////////////////////
// "ErrorReportingType" Property Constants
////////////////////////////////////////////////////////////////////

const LONG MSR_ERT_CARD       = 0;
const LONG MSR_ERT_TRACK      = 1;


////////////////////////////////////////////////////////////////////
```

```
// "CapDataEncryption", "DataEncryptionAlgorithm" Property Constants
//   (added in 1.12)
////////////////////////////////////////////////////////////////////////

const LONG MSR_DE_NONE         = 0x00000001;
const LONG MSR_DE_3DEA_DUKPT   = 0x00000002;
// Note: Service-specific values begin at 0x01000000.


////////////////////////////////////////////////////////////////////////
// "CapDeviceAuthentication" Property Constants (added in 1.12)
////////////////////////////////////////////////////////////////////////

const LONG MSR_DA_NOT_SUPPORTED = 0;
const LONG MSR_DA_OPTIONAL     = 1;
const LONG MSR_DA_REQUIRED     = 2;


////////////////////////////////////////////////////////////////////////
// "DeviceAuthenticationProtocol" Property Constants (added in 1.12)
////////////////////////////////////////////////////////////////////////

const LONG MSR_AP_NONE            = 0;
const LONG MSR_AP_CHALLENGERESPONSE = 1;


////////////////////////////////////////////////////////////////////////
// "CardType" Property Constants (added in 1.12)
////////////////////////////////////////////////////////////////////////

#define MSR_CT_AAMVA          "AAMVA"
#define MSR_CT_BANK           "BANK"


////////////////////////////////////////////////////////////////////////
// "RetrieveCardProperty" Method: "Name" Parameter Constants
//   (added in 1.12)
////////////////////////////////////////////////////////////////////////

#define MSR_RCP_AccountNumber   "AccountNumber"
#define MSR_RCP_Address        "Address"
#define MSR_RCP_BirthDate      "BirthDate"
#define MSR_RCP_City           "City"
#define MSR_RCP_Class          "Class"
#define MSR_RCP_Endorsements    "Endorsements"
#define MSR_RCP_ExpirationDate  "ExpirationDate"
#define MSR_RCP_EyeColor       "EyeColor"
#define MSR_RCP_FirstName      "FirstName"
#define MSR_RCP_Gender         "Gender"
#define MSR_RCP_HairColor      "HairColor"
#define MSR_RCP_Height         "Height"
#define MSR_RCP_LicenseNumber   "LicenseNumber"
#define MSR_RCP_MiddleInitial   "MiddleInitial"
#define MSR_RCP_PostalCode     "PostalCode"
#define MSR_RCP_Restrictions    "Restrictions"
#define MSR_RCP_ServiceCode    "ServiceCode"
#define MSR_RCP_State          "State"
#define MSR_RCP_Suffix         "Suffix"
#define MSR_RCP_Surname        "Surname"
#define MSR_RCP_Title          "Title"
#define MSR_RCP_Weight         "Weight"


////////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constants
////////////////////////////////////////////////////////////////////////
```

```
const LONG MSR_SUE_DEVICE_AUTHENTICATED    =  11; // (added in 1.12)
const LONG MSR_SUE_DEVICE_DEAUTHENTICATED   =  12; // (added in 1.12)


////////////////////////////////////////////////////////////////////
// "ErrorEvent" Event: "ResultCodeExtended" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_EMSR_START     = 201;
const LONG OPOS_EMSR_END       = 202;
const LONG OPOS_EMSR_PARITY    = 203;
const LONG OPOS_EMSR_LRC       = 204;
const LONG OPOS_EMSR_DEVICE_AUTHENTICATION_FAILED   = 205; // (added
in 1.12)
const LONG OPOS_EMSR_DEVICE_DEAUTHENTICATION_FAILED = 206; // (added
in 1.12)


#endif               // !defined(OPOSMSR_H)
```

OposPpad.h :　PINパッドヘッダーファイル

```
///////////////////////////////////////////////////////////////////////
//
// OposPpad.h
//
//  PIN Pad header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 1998-03-06 OPOS Release 1.3                                  JDB
// 2025-09-17 OPOS Release 1.15.1                               KF
//  Corrected missing define of DataEvent constant PPAD_TIMEOUT.
//
///////////////////////////////////////////////////////////////////////

#if !defined(OPOSPPAD_H)
#define     OPOSPPAD_H


#include "Opos.h"


///////////////////////////////////////////////////////////////////////
// "CapDisplay" Property Constants
///////////////////////////////////////////////////////////////////////

const LONG PPAD_DISP_UNRESTRICTED         = 1;
const LONG PPAD_DISP_PINRESTRICTED        = 2;
const LONG PPAD_DISP_RESTRICTED_LIST      = 3;
const LONG PPAD_DISP_RESTRICTED_ORDER     = 4;
const LONG PPAD_DISP_NONE                 = 5;


///////////////////////////////////////////////////////////////////////
// "AvailablePromptsList" and "Prompt" Property Constants
///////////////////////////////////////////////////////////////////////

const LONG PPAD_MSG_ENTERPIN              = 1;
const LONG PPAD_MSG_PLEASEWAIT            = 2;
const LONG PPAD_MSG_ENTERVALIDPIN         = 3;
const LONG PPAD_MSG_RETRIESEXCEEDED       = 4;
const LONG PPAD_MSG_APPROVED              = 5;
const LONG PPAD_MSG_DECLINED              = 6;
const LONG PPAD_MSG_CANCELED              = 7;
const LONG PPAD_MSG_AMOUNTOK              = 8;
const LONG PPAD_MSG_NOTREADY              = 9;
const LONG PPAD_MSG_IDLE                  = 10;
const LONG PPAD_MSG_SLIDE_CARD            = 11;
const LONG PPAD_MSG_INSERTCARD            = 12;
const LONG PPAD_MSG_SELECTCARDTYPE        = 13;


///////////////////////////////////////////////////////////////////////
// "CapLanguage" Property Constants
///////////////////////////////////////////////////////////////////////

const LONG PPAD_LANG_NONE                 = 1;
const LONG PPAD_LANG_ONE                  = 2;
const LONG PPAD_LANG_PINRESTRICTED        = 3;
const LONG PPAD_LANG_UNRESTRICTED         = 4;

///////////////////////////////////////////////////////////////////////
// "TransactionType" Property Constants
```

```
////////////////////////////////////////////////////////////////////////

const LONG PPAD_TRANS_DEBIT             = 1;
const LONG PPAD_TRANS_CREDIT            = 2;
const LONG PPAD_TRANS_INQ               = 3;
const LONG PPAD_TRANS_RECONCILE          = 4;
const LONG PPAD_TRANS_ADMIN             = 5;


////////////////////////////////////////////////////////////////////////
// "EndEFTTransaction" Method Completion Code Constants
////////////////////////////////////////////////////////////////////////

const LONG PPAD_EFT_NORMAL              = 1;
const LONG PPAD_EFT_ABNORMAL            = 2;


////////////////////////////////////////////////////////////////////////
// "DataEvent" Event Status Constants
////////////////////////////////////////////////////////////////////////
const LONG PPAD_SUCCESS                 = 1;
const LONG PPAD_CANCEL                  = 2;
const LONG PPAD_TIMEOUT                 = 3;


////////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
////////////////////////////////////////////////////////////////////////

const LONG OPOS_EPPAD_BAD_KEY           = 201;


#endif              // !defined(OPOSPPAD_H)
```

OposPcrw.h： ポイントカード機ヘッダーファイル

```
//////////////////////////////////////////////////////////////////////
//
// OposPcrw.H
//
//  Point Card Reader Writer header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 2000-09-24 OPOS Release 1.5                                  BKS
// 2004-10-26 Add "CharacterSet" ANSI constant (from 1.5).      CRM
//
//////////////////////////////////////////////////////////////////////

#if !defined(OPOSPCRW_H)
#define     OPOSPCRW_H


#include "opos.h"


//////////////////////////////////////////////////////////////////////
// "CapCharacterSet" Property Constants
//////////////////////////////////////////////////////////////////////

const LONG PCRW_CCS_ALPHA      = 1;
const LONG PCRW_CCS_ASCII      = 998;
const LONG PCRW_CCS_KANA       = 10;
const LONG PCRW_CCS_KANJI      = 11;
const LONG PCRW_CCS_UNICODE    = 997;


//////////////////////////////////////////////////////////////////////
// "CardState" Property Constants
//////////////////////////////////////////////////////////////////////

const LONG PCRW_STATE_NOCARD   = 1;
const LONG PCRW_STATE_REMAINING = 2;
const LONG PCRW_STATE_INRW     = 4;


//////////////////////////////////////////////////////////////////////
// CapTrackToRead and TrackToWrite Property constants
//////////////////////////////////////////////////////////////////////

const LONG PCRW_TRACK1         = 0x0001;
const LONG PCRW_TRACK2         = 0x0002;
const LONG PCRW_TRACK3         = 0x0004;
const LONG PCRW_TRACK4         = 0x0008;
const LONG PCRW_TRACK5         = 0x0010;
const LONG PCRW_TRACK6         = 0x0020;


//////////////////////////////////////////////////////////////////////
// "CharacterSet" Property Constants
//////////////////////////////////////////////////////////////////////

const LONG PCRW_CS_UNICODE     = 997;
const LONG PCRW_CS_ASCII       = 998;
const LONG PCRW_CS_WINDOWS     = 999;
const LONG PCRW_CS_ANSI        = 999;


//////////////////////////////////////////////////////////////////////
// "MappingMode" Property Constants
//////////////////////////////////////////////////////////////////////

const LONG PCRW_MM_DOTS        = 1;
const LONG PCRW_MM_TWIPS       = 2;
const LONG PCRW_MM_ENGLISH     = 3;
const LONG PCRW_MM_METRIC      = 4;
```

```
/////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
/////////////////////////////////////////////////////////////////////

const LONG OPOS_EPCRW_READ     = 201;
const LONG OPOS_EPCRW_WRITE    = 202;
const LONG OPOS_EPCRW_JAM      = 203;
const LONG OPOS_EPCRW_MOTOR    = 204;
const LONG OPOS_EPCRW_COVER    = 205;
const LONG OPOS_EPCRW_PRINTER  = 206;
const LONG OPOS_EPCRW_RELEASE  = 207;
const LONG OPOS_EPCRW_DISPLAY  = 208;
const LONG OPOS_EPCRW_NOCARD   = 209;


/////////////////////////////////////////////////////////////////////
// Magnetic read/write status Property Constants
/////////////////////////////////////////////////////////////////////

const LONG OPOS_EPCRW_START    = 211;
const LONG OPOS_EPCRW_END      = 212;
const LONG OPOS_EPCRW_PARITY   = 213;
const LONG OPOS_EPCRW_ENCODE   = 214;
const LONG OPOS_EPCRW_LRC      = 215;
const LONG OPOS_EPCRW_VERIFY   = 216;


/////////////////////////////////////////////////////////////////////
// "RotatedPrint" Method: "Rotation" Parameter Constants
// "RotateSpecial" Property Constants
/////////////////////////////////////////////////////////////////////

const LONG PCRW_RP_NORMAL      = 0x0001;
const LONG PCRW_RP_RIGHT90     = 0x0101;
const LONG PCRW_RP_LEFT90      = 0x0102;
const LONG PCRW_RP_ROTATE180   = 0x0103;


/////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" "Status" Parameter Constants
/////////////////////////////////////////////////////////////////////

const LONG PCRW_SUE_NOCARD     = 1;
const LONG PCRW_SUE_REMAINING  = 2;
const LONG PCRW_SUE_INRW       = 4;


#endif               // !defined(OPOSPCRW_H)
```

OposPtr.h： POSプリンタヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposPtr.h
//
//  POS Printer header file for OPOS Applications.
//
// Modification history
// -----------------------------------------------------------------
// 1995-12-08 OPOS Release 1.0                                  CRM
// 1996-04-22 OPOS Release 1.1                                  CRM
//  Add CapCharacterSet constants.
//  Add ErrorLevel constants.
//  Add TransactionPrint's Control constants.
// 1997-06-04 OPOS Release 1.2                                  CRM
//  Remove PTR_RP_NORMAL_ASYNC.
//  Add more barcode symbologies.
// 1998-03-06 OPOS Release 1.3                                  CRM
//  Add more PrintTwoNormal constants.
// 2000-09-24 OPOS Release 1.5                            EPSON/BKS
//  Add CapRecMarkFeed and MarkFeed's Type constants.
//  Add ChangePrintSide constants.
//  Add StatusUpdateEvent constants.
//  Add ResultCodeExtended constants.
//  Add CapXxxCartridgeSensor and XxxCartridgeState constants.
//  Add CartridgeNotify constants.
//  Add CapCharacterset and CharacterSet constants for UNICODE.
// 2003-05-29 OPOS Release 1.7                                  CRM
//  Add more RotatePrint's Rotation PTR_RP_* constants.
// 2004-03-22 OPOS Release 1.8                                  CRM
//  Add more constants for PrintBarCode method and StatusUpdateEvent.
// 2004-10-26 Add "CharacterSet" ANSI constant (from 1.5).      CRM
// 2005-04-29 OPOS Release 1.9                                  CRM
//  Add PageModeDescriptor and PageModePrintDirection constants.
//  Add PageModePrint's Control constants.
// 2006-03-15 OPOS Release 1.10                                 CRM
//  Add PrintMemoryBitmap constants.
// 2008-08-30 OPOS Release 1.12                                 CRM
//  Add more constants for PrintBarCode method.
// 2009-10-02 OPOS Release 1.13                                 CRM
//  Add CapRecRuledLine and CapSlpRuledLine constants.
//  Add DrawRuledLine constants.
//  Add more constants for PrintBarCode method.
// 2025-09-17 OPOS Release 1.15.1                               KF
//  Add more constants for PrintBarCode method.
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSPTR_H)
#define     OPOSPTR_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// Printer Station Constants
////////////////////////////////////////////////////////////////////

const LONG PTR_S_JOURNAL       = 1;
const LONG PTR_S_RECEIPT       = 2;
const LONG PTR_S_SLIP          = 4;

const LONG PTR_S_JOURNAL_RECEIPT    = 0x0003;
const LONG PTR_S_JOURNAL_SLIP       = 0x0005;
const LONG PTR_S_RECEIPT_SLIP       = 0x0006;

const LONG PTR_TWO_RECEIPT_JOURNAL  = 0x8003; // (added in 1.3)
const LONG PTR_TWO_SLIP_JOURNAL     = 0x8005; // (added in 1.3)
const LONG PTR_TWO_SLIP_RECEIPT     = 0x8006; // (added in 1.3)
```

```
////////////////////////////////////////////////////////////////////
// "CapCharacterSet" Property Constants (added in 1.1)
////////////////////////////////////////////////////////////////////

const LONG PTR_CCS_ALPHA       =   1;
const LONG PTR_CCS_ASCII       = 998;
const LONG PTR_CCS_KANA        =  10;
const LONG PTR_CCS_KANJI       =  11;
const LONG PTR_CCS_UNICODE     = 997; // (added in 1.5)


////////////////////////////////////////////////////////////////////
// "CharacterSet" Property Constants
////////////////////////////////////////////////////////////////////

const LONG PTR_CS_UNICODE      = 997; // (added in 1.5)
const LONG PTR_CS_ASCII        = 998;
const LONG PTR_CS_WINDOWS      = 999;
const LONG PTR_CS_ANSI         = 999;


////////////////////////////////////////////////////////////////////
// "ErrorLevel" Property Constants (added in 1.1)
////////////////////////////////////////////////////////////////////

const LONG PTR_EL_NONE         = 1;
const LONG PTR_EL_RECOVERABLE   = 2;
const LONG PTR_EL_FATAL        = 3;


////////////////////////////////////////////////////////////////////
// "MapMode" Property Constants
////////////////////////////////////////////////////////////////////

const LONG PTR_MM_DOTS         = 1;
const LONG PTR_MM_TWIPS        = 2;
const LONG PTR_MM_ENGLISH      = 3;
const LONG PTR_MM_METRIC       = 4;


////////////////////////////////////////////////////////////////////
// "CapXxxColor" Property Constants
////////////////////////////////////////////////////////////////////

const LONG PTR_COLOR_PRIMARY     = 0x00000001;
const LONG PTR_COLOR_CUSTOM1     = 0x00000002;
const LONG PTR_COLOR_CUSTOM2     = 0x00000004;
const LONG PTR_COLOR_CUSTOM3     = 0x00000008;
const LONG PTR_COLOR_CUSTOM4     = 0x00000010;
const LONG PTR_COLOR_CUSTOM5     = 0x00000020;
const LONG PTR_COLOR_CUSTOM6     = 0x00000040;
const LONG PTR_COLOR_CYAN        = 0x00000100;
const LONG PTR_COLOR_MAGENTA     = 0x00000200;
const LONG PTR_COLOR_YELLOW      = 0x00000400;
const LONG PTR_COLOR_FULL        = 0x80000000;


////////////////////////////////////////////////////////////////////
// "CapXxxCartridgeSensor" and  "XxxCartridgeState" Property Constants
//   (added in 1.5)
////////////////////////////////////////////////////////////////////

const LONG PTR_CART_UNKNOWN       = 0x10000000;
const LONG PTR_CART_OK            = 0x00000000;
const LONG PTR_CART_REMOVED       = 0x00000001;
const LONG PTR_CART_EMPTY         = 0x00000002;
const LONG PTR_CART_NEAREND       = 0x00000004;
const LONG PTR_CART_CLEANING      = 0x00000008;


////////////////////////////////////////////////////////////////////
// "CartridgeNotify" Property Constants (added in 1.5)
////////////////////////////////////////////////////////////////////

const LONG PTR_CN_DISABLED      = 0x00000000;
```

```
const LONG PTR_CN_ENABLED       = 0x00000001;


////////////////////////////////////////////////////////////////////
// "PageModeDescriptor" Property Constants (added in 1.9)
////////////////////////////////////////////////////////////////////

const LONG PTR_PM_BITMAP        = 0x00000001;
const LONG PTR_PM_BARCODE       = 0x00000002;
const LONG PTR_PM_BM_ROTATE     = 0x00000004;
const LONG PTR_PM_BC_ROTATE     = 0x00000008;
const LONG PTR_PM_OPAQUE        = 0x00000010;


////////////////////////////////////////////////////////////////////
// "PageModePrintDirection" Property Constants (added in 1.9)
////////////////////////////////////////////////////////////////////

const LONG PTR_PD_LEFT_TO_RIGHT = 1;
const LONG PTR_PD_BOTTOM_TO_TOP = 2;
const LONG PTR_PD_RIGHT_TO_LEFT = 3;
const LONG PTR_PD_TOP_TO_BOTTOM = 4;


////////////////////////////////////////////////////////////////////
// "CutPaper" Method Constant
////////////////////////////////////////////////////////////////////

const LONG PTR_CP_FULLCUT      = 100;


////////////////////////////////////////////////////////////////////
// "PageModePrint" Method: "Control" Parameter Constants (added in
1.9)
////////////////////////////////////////////////////////////////////

const LONG PTR_PM_PAGE_MODE     = 1;
const LONG PTR_PM_PRINT_SAVE    = 2;
const LONG PTR_PM_NORMAL        = 3;
const LONG PTR_PM_CANCEL        = 4;


////////////////////////////////////////////////////////////////////
// "PrintBarCode" Method Constants:
////////////////////////////////////////////////////////////////////

//** "Alignment" Parameter
//    Either the distance from the left-most print column to the start
//    of the bar code, or one of the following:

const LONG PTR_BC_LEFT         = -1;
const LONG PTR_BC_CENTER       = -2;
const LONG PTR_BC_RIGHT        = -3;

//** "TextPosition" Parameter

const LONG PTR_BC_TEXT_NONE    = -11;
const LONG PTR_BC_TEXT_ABOVE   = -12;
const LONG PTR_BC_TEXT_BELOW   = -13;

//** "Symbology" Parameter:

//    - One dimensional symbologies
const LONG PTR_BCS_UPCA        = 101;  // Digits
const LONG PTR_BCS_UPCE        = 102;  // Digits
const LONG PTR_BCS_JAN8        = 103;  // = EAN 8
const LONG PTR_BCS_EAN8        = 103;  // = JAN 8 (added in 1.2)
const LONG PTR_BCS_JAN13       = 104;  // = EAN 13
const LONG PTR_BCS_EAN13       = 104;  // = JAN 13 (added in 1.2)
const LONG PTR_BCS_TF          = 105;  // (Discrete 2 of 5) Digits
const LONG PTR_BCS_ITF         = 106;  // (Interleaved 2 of 5) Digits
const LONG PTR_BCS_Codabar     = 107;  // Digits, -, $, :, /, ., +;
                                    //   4 start/stop characters
                                    //   (a, b, c, d)
const LONG PTR_BCS_Code39      = 108;  // Alpha, Digits, Space, -, .,
```

```
                                    //  $, /, +, %; start/stop (*)
                                    // Also has Full ASCII feature
const LONG PTR_BCS_Code93      = 109;  // Same characters as Code 39
const LONG PTR_BCS_Code128     = 110;  // 128 data characters

//    - One dimensional symbologies (added in 1.2)
const LONG PTR_BCS_UPCA_S      = 111;  // UPC-A with supplemental
                                    //  barcode
const LONG PTR_BCS_UPCE_S      = 112;  // UPC-E with supplemental
                                    //  barcode
const LONG PTR_BCS_UPCD1       = 113;  // UPC-D1
const LONG PTR_BCS_UPCD2       = 114;  // UPC-D2
const LONG PTR_BCS_UPCD3       = 115;  // UPC-D3
const LONG PTR_BCS_UPCD4       = 116;  // UPC-D4
const LONG PTR_BCS_UPCD5       = 117;  // UPC-D5
const LONG PTR_BCS_EAN8_S      = 118;  // EAN 8 with supplemental
                                    //  barcode
const LONG PTR_BCS_EAN13_S     = 119;  // EAN 13 with supplemental
                                    //  barcode

//       The following EAN128 constant deprecated in 1.15.1
//       Instead use the GS1128 constant below.
const LONG PTR_BCS_EAN128      = 120;  // EAN 128

// - One dimensional symbologies (added in 1.15.1)
const LONG PTR_BCS_EAN128      = 120;  // EAN 128

const LONG PTR_BCS_OCRA        = 121;  // OCR "A"
const LONG PTR_BCS_OCRB        = 122;  // OCR "B"

//    - One dimensional symbologies (added in 1.8)
const LONG PTR_BCS_Code128_Parsed=123;  // Code 128 with parsing

// - Two dimensional symbologies (added in 1.15.1)
const LONG PTR_BCS_DOTCODE     = 124;  // DotCode
const LONG PTR_BCS_HANXIN      = 125;  // Han Xin Code

//       The following RSS constants deprecated in 1.12.
//       Instead use the GS1DATABAR constants below.
const LONG PTR_BCS_RSS14       = 131;  // Reduced Space Symbology - 14
digit GTIN
const LONG PTR_BCS_RSS_EXPANDED = 132;  // RSS - 14 digit GTIN plus
additional fields

//    - One dimensional symbologies (added in 1.12)
const LONG PTR_BCS_GS1DATABAR   =131;  // GS1 DataBar Omnidirectional
const LONG PTR_BCS_GS1DATABAR_E  =132;  // GS1 DataBar Expanded
const  LONG  PTR_BCS_GS1DATABAR_S    =133;    //  GS1  DataBar  Stacked
Omnidirectional
const LONG PTR_BCS_GS1DATABAR_E_S=134;  // GS1 DataBar Expanded Stacked

//    - Two dimensional symbologies
const LONG PTR_BCS_PDF417      = 201;
const LONG PTR_BCS_MAXICODE    = 202;

//    - Two dimensional symbologies (added in 1.13)
const LONG PTR_BCS_DATAMATRIX   = 203;  // Data Matrix
const LONG PTR_BCS_QRCODE      = 204;  // QR Code
const LONG PTR_BCS_UQRCODE     = 205;  // Micro QR Code
const LONG PTR_BCS_AZTEC       = 206;  // Aztec
const LONG PTR_BCS_UPDF417     = 207;  // Micro PDF 417

// - Two dimensional symbologies (added in 1.15.1)
const LONG PTR_BCS_GS1DATAMATRIX   = 208;  // GS1 Data Matrix

//    - Start of Printer-Specific bar code symbologies
const LONG PTR_BCS_OTHER       = 501;


////////////////////////////////////////////////////////////////////
// "PrintBitmap" and "PrintMemoryBitmap" Method Constants:
////////////////////////////////////////////////////////////////////

//** "Width" Parameter
//    Either bitmap width or:
```

```
const LONG PTR_BM_ASIS        = -11;  // One pixel per printer dot

//** "Alignment" Parameter
//    Either the distance from the left-most print column to the start
//    of the bitmap, or one of the following:

const LONG PTR_BM_LEFT        = -1;
const LONG PTR_BM_CENTER      = -2;
const LONG PTR_BM_RIGHT       = -3;

//** "Type" Parameter ("PrintMemoryBitmap" only)
const LONG PTR_BMT_BMP        = 1;
const LONG PTR_BMT_JPEG       = 2;
const LONG PTR_BMT_GIF        = 3;


////////////////////////////////////////////////////////////////////
// "RotatePrint" Method: "Rotation" Parameter Constants
// "RotateSpecial" Property Constants
////////////////////////////////////////////////////////////////////

const LONG PTR_RP_NORMAL      = 0x0001;
const LONG PTR_RP_RIGHT90     = 0x0101;
const LONG PTR_RP_LEFT90      = 0x0102;
const LONG PTR_RP_ROTATE180   = 0x0103;

// For "RotatePrint", one or both of the following values may be
// ORed with one of the above values.
const LONG PTR_RP_BARCODE     = 0x1000; // (added in 1.7)
const LONG PTR_RP_BITMAP      = 0x2000; // (added in 1.7)


////////////////////////////////////////////////////////////////////
// "SetLogo" Method: "Location" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG PTR_L_TOP          = 1;
const LONG PTR_L_BOTTOM       = 2;


////////////////////////////////////////////////////////////////////
// "TransactionPrint" Method: "Control" Parameter Constants (added in
1.1)
////////////////////////////////////////////////////////////////////

const LONG PTR_TP_TRANSACTION  = 11;
const LONG PTR_TP_NORMAL       = 12;


////////////////////////////////////////////////////////////////////
// "MarkFeed" Method: "Type" Parameter Constants (added in 1.5)
// "CapRecMarkFeed" Property Constants (added in 1.5)
////////////////////////////////////////////////////////////////////

const LONG PTR_MF_TO_TAKEUP    = 1;
const LONG PTR_MF_TO_CUTTER    = 2;
const LONG PTR_MF_TO_CURRENT_TOF= 4;
const LONG PTR_MF_TO_NEXT_TOF  = 8;


////////////////////////////////////////////////////////////////////
// "ChangePrintSide" Method: "Side" Parameter Constants (added in 1.5)
////////////////////////////////////////////////////////////////////

const LONG PTR_PS_UNKNOWN      = 0;
const LONG PTR_PS_SIDE1        = 1;
const LONG PTR_PS_SIDE2        = 2;
const LONG PTR_PS_OPPOSITE     = 3;


////////////////////////////////////////////////////////////////////
// "CapRecRuledLine" and "CapSlpRuledLine" Property Constants
// "DrawRuledLine" Method: "LineDirection" Parameter Constants
// (added in 1.13)
```

```
////////////////////////////////////////////////////////////////////

const LONG PTR_RL_HORIZONTAL = 1;
const LONG PTR_RL_VERTICAL   = 2;


////////////////////////////////////////////////////////////////////
// "DrawRuledLine" Method: "LineStyle" Parameter Constants
// (added in 1.13)
////////////////////////////////////////////////////////////////////

const LONG PTR_LS_SINGLE_SOLID_LINE = 1;
const LONG PTR_LS_DOUBLE_SOLID_LINE = 2;
const LONG PTR_LS_BROKEN_LINE      = 3;
const LONG PTR_LS_CHAIN_LINE       = 4;


////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG PTR_SUE_COVER_OPEN            = 11;
const LONG PTR_SUE_COVER_OK              = 12;
const LONG PTR_SUE_JRN_COVER_OPEN        = 60;  // (added in 1.8)
const LONG PTR_SUE_JRN_COVER_OK          = 61;  // (added in 1.8)
const LONG PTR_SUE_REC_COVER_OPEN        = 62;  // (added in 1.8)
const LONG PTR_SUE_REC_COVER_OK          = 63;  // (added in 1.8)
const LONG PTR_SUE_SLP_COVER_OPEN        = 64;  // (added in 1.8)
const LONG PTR_SUE_SLP_COVER_OK          = 65;  // (added in 1.8)

const LONG PTR_SUE_JRN_EMPTY             = 21;
const LONG PTR_SUE_JRN_NEAREMPTY         = 22;
const LONG PTR_SUE_JRN_PAPEROK           = 23;

const LONG PTR_SUE_REC_EMPTY             = 24;
const LONG PTR_SUE_REC_NEAREMPTY         = 25;
const LONG PTR_SUE_REC_PAPEROK           = 26;

const LONG PTR_SUE_SLP_EMPTY             = 27;
const LONG PTR_SUE_SLP_NEAREMPTY         = 28;
const LONG PTR_SUE_SLP_PAPEROK           = 29;

const LONG PTR_SUE_JRN_CARTRIDGE_EMPTY    = 41; // (added in 1.5)
const LONG PTR_SUE_JRN_CARTRIDGE_NEAREMPTY = 42; // (added in 1.5)
const LONG PTR_SUE_JRN_HEAD_CLEANING      = 43; // (added in 1.5)
const LONG PTR_SUE_JRN_CARTRIDGE_OK       = 44; // (added in 1.5)

const LONG PTR_SUE_REC_CARTRIDGE_EMPTY    = 45; // (added in 1.5)
const LONG PTR_SUE_REC_CARTRIDGE_NEAREMPTY = 46; // (added in 1.5)
const LONG PTR_SUE_REC_HEAD_CLEANING      = 47; // (added in 1.5)
const LONG PTR_SUE_REC_CARTRIDGE_OK       = 48; // (added in 1.5)

const LONG PTR_SUE_SLP_CARTRIDGE_EMPTY    = 49; // (added in 1.5)
const LONG PTR_SUE_SLP_CARTRIDGE_NEAREMPTY = 50; // (added in 1.5)
const LONG PTR_SUE_SLP_HEAD_CLEANING      = 51; // (added in 1.5)
const LONG PTR_SUE_SLP_CARTRIDGE_OK       = 52; // (added in 1.5)

const LONG PTR_SUE_IDLE                  = 1001;


////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_EPTR_COVER_OPEN          = 201; // (Several)
const LONG OPOS_EPTR_JRN_EMPTY           = 202; // (Several)
const LONG OPOS_EPTR_REC_EMPTY           = 203; // (Several)
const LONG OPOS_EPTR_SLP_EMPTY           = 204; // (Several)
const LONG OPOS_EPTR_SLP_FORM            = 205; // EndRemoval
const LONG OPOS_EPTR_TOOBIG              = 206; // PrintBitmap
const LONG OPOS_EPTR_BADFORMAT           = 207; // PrintBitmap
const LONG OPOS_EPTR_JRN_CARTRIDGE_REMOVED = 208; // (Several) (added
in 1.5)
const LONG OPOS_EPTR_JRN_CARTRIDGE_EMPTY   = 209; // (Several) (added
in 1.5)
```

```
const LONG OPOS_EPTR_JRN_HEAD_CLEANING     = 210; // (Several) (added
in 1.5)
const LONG OPOS_EPTR_REC_CARTRIDGE_REMOVED = 211; // (Several) (added
in 1.5)
const LONG OPOS_EPTR_REC_CARTRIDGE_EMPTY   = 212; // (Several) (added
in 1.5)
const LONG OPOS_EPTR_REC_HEAD_CLEANING     = 213; // (Several) (added
in 1.5)
const LONG OPOS_EPTR_SLP_CARTRIDGE_REMOVED = 214; // (Several) (added
in 1.5)
const LONG OPOS_EPTR_SLP_CARTRIDGE_EMPTY   = 215; // (Several) (added
in 1.5)
const LONG OPOS_EPTR_SLP_HEAD_CLEANING     = 216; // (Several) (added
in 1.5)


#endif               // !defined(OPOSPTR_H)
```

OposPwr.h：　パワーマネージメントヘッダーファイル

```
///////////////////////////////////////////////////////////////////////
//
// OposPwr.h
//
//  POSPower header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 2000-09-24 OPOS Release 1.5                                  TH
// 2005-04-29 OPOS Release 1.9                                  CRM
//  Add PowerSource constants.
//  Add Reason constants for RestartPOS, StandbyPOS, SuspendPOS.
//  Add StatusUpdateEvent constants for battery support.
// 2022-01-18 OPOS Release 1.16                                 KF
//  Add StatusUpdateEvent constants for battery support by in seconds.
//
///////////////////////////////////////////////////////////////////////

#if !defined(OPOSPOWER_H)
#define     OPOSPOWER_H

///////////////////////////////////////////////////////////////////////
// "UPSChargeState" Property Constants
///////////////////////////////////////////////////////////////////////

const LONG PWR_UPS_FULL        = 1;
const LONG PWR_UPS_WARNING     = 2;
const LONG PWR_UPS_LOW         = 4;
const LONG PWR_UPS_CRITICAL    = 8;


///////////////////////////////////////////////////////////////////////
// "PowerSource" Property Constants (added in 1.9)
///////////////////////////////////////////////////////////////////////

const LONG PWR_SOURCE_NA       = 1;
const LONG PWR_SOURCE_AC       = 2;
const LONG PWR_SOURCE_BATTERY   = 3;
const LONG PWR_SOURCE_BACKUP    = 4;


///////////////////////////////////////////////////////////////////////
// "RestartPOS", "StandbyPOS", "SuspendPOS" Methods:
//   "Reason" Parameter Constants (added in 1.9)
///////////////////////////////////////////////////////////////////////

const LONG PWR_REASON_REQUEST   = 1;
const LONG PWR_REASON_ALLOW    = 2;
const LONG PWR_REASON_DENY     = 3;


///////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Status" Constants
///////////////////////////////////////////////////////////////////////

const LONG PWR_SUE_UPS_FULL              = 11;
const LONG PWR_SUE_UPS_WARNING            = 12;
const LONG PWR_SUE_UPS_LOW               = 13;
const LONG PWR_SUE_UPS_CRITICAL           = 14;
const LONG PWR_SUE_FAN_STOPPED            = 15;
const LONG PWR_SUE_FAN_RUNNING            = 16;
const LONG PWR_SUE_TEMPERATURE_HIGH        = 17;
const LONG PWR_SUE_TEMPERATURE_OK         = 18;
const LONG PWR_SUE_SHUTDOWN              = 19;
const LONG PWR_SUE_BAT_LOW               = 20; // (added in 1.9)
const LONG PWR_SUE_BAT_CRITICAL           = 21; // (added in 1.9)
const LONG PWR_SUE_BAT_CAPACITY_REMAINING   = 22; // (added in 1.9)
const LONG PWR_SUE_RESTART               = 23; // (added in 1.9)
const LONG PWR_SUE_STANDBY               = 24; // (added in 1.9)
const LONG PWR_SUE_USER_STANDBY           = 25; // (added in 1.9)
const LONG PWR_SUE_SUSPEND               = 26; // (added in 1.9)
```

```
const LONG PWR_SUE_USER_SUSPEND           = 27; // (added in 1.9)
const LONG PWR_SUE_PWR_SOURCE             = 28; // (added in 1.9)
const LONG PWR_SUE_BAT_CAPACITY_REMAINING_IN_SECONDS = 29; // (added
in 1.16)

#endif               // !defined(OPOSPOWER_H)
```

OposRod.h：リモートオーダーディスプレイヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposRod.h
//
//  Remote Order Display header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 1998-03-06 OPOS Release 1.3                                  BB
// 2000-09-24 OPOS Release 1.5                                  BKS
//  Added CharacterSet constant for UNICODE.
// 2004-10-26 Add "CharacterSet" ANSI constant (from 1.5).      CRM
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSROD_H)
#define     OPOSROD_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "CurrentUnitID" and "UnitsOnline" Properties
//   and "Units" Parameter Constants
////////////////////////////////////////////////////////////////////

#define ROD_UID(Unit) (1 << (Unit-1))

const LONG ROD_UID_1       = 0x00000001;
const LONG ROD_UID_2       = 0x00000002;
const LONG ROD_UID_3       = 0x00000004;
const LONG ROD_UID_4       = 0x00000008;
const LONG ROD_UID_5       = 0x00000010;
const LONG ROD_UID_6       = 0x00000020;
const LONG ROD_UID_7       = 0x00000040;
const LONG ROD_UID_8       = 0x00000080;
const LONG ROD_UID_9       = 0x00000100;
const LONG ROD_UID_10      = 0x00000200;
const LONG ROD_UID_11      = 0x00000400;
const LONG ROD_UID_12      = 0x00000800;
const LONG ROD_UID_13      = 0x00001000;
const LONG ROD_UID_14      = 0x00002000;
const LONG ROD_UID_15      = 0x00004000;
const LONG ROD_UID_16      = 0x00008000;
const LONG ROD_UID_17      = 0x00010000;
const LONG ROD_UID_18      = 0x00020000;
const LONG ROD_UID_19      = 0x00040000;
const LONG ROD_UID_20      = 0x00080000;
const LONG ROD_UID_21      = 0x00100000;
const LONG ROD_UID_22      = 0x00200000;
const LONG ROD_UID_23      = 0x00400000;
const LONG ROD_UID_24      = 0x00800000;
const LONG ROD_UID_25      = 0x01000000;
const LONG ROD_UID_26      = 0x02000000;
const LONG ROD_UID_27      = 0x04000000;
const LONG ROD_UID_28      = 0x08000000;
const LONG ROD_UID_29      = 0x10000000;
const LONG ROD_UID_30      = 0x20000000;
const LONG ROD_UID_31      = 0x40000000;
const LONG ROD_UID_32      = 0x80000000;


////////////////////////////////////////////////////////////////////
// Broadcast Methods: "Attribute" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG ROD_ATTR_BLINK      = 0x80;

const LONG ROD_ATTR_BG_BLACK    = 0x00;
const LONG ROD_ATTR_BG_BLUE     = 0x10;
```

```
const LONG ROD_ATTR_BG_GREEN    = 0x20;
const LONG ROD_ATTR_BG_CYAN     = 0x30;
const LONG ROD_ATTR_BG_RED      = 0x40;
const LONG ROD_ATTR_BG_MAGENTA  = 0x50;
const LONG ROD_ATTR_BG_BROWN    = 0x60;
const LONG ROD_ATTR_BG_GRAY     = 0x70;

const LONG ROD_ATTR_INTENSITY   = 0x08;

const LONG ROD_ATTR_FG_BLACK    = 0x00;
const LONG ROD_ATTR_FG_BLUE     = 0x01;
const LONG ROD_ATTR_FG_GREEN    = 0x02;
const LONG ROD_ATTR_FG_CYAN     = 0x03;
const LONG ROD_ATTR_FG_RED      = 0x04;
const LONG ROD_ATTR_FG_MAGENTA  = 0x05;
const LONG ROD_ATTR_FG_BROWN    = 0x06;
const LONG ROD_ATTR_FG_GRAY     = 0x07;


////////////////////////////////////////////////////////////////////////
// "DrawBox" Method: "BorderType" Parameter Constants
////////////////////////////////////////////////////////////////////////

const LONG ROD_BDR_SINGLE       = 1;
const LONG ROD_BDR_DOUBLE       = 2;
const LONG ROD_BDR_SOLID        = 3;


////////////////////////////////////////////////////////////////////////
// "ControlClock" Method: "Function" Parameter Constants
////////////////////////////////////////////////////////////////////////

const LONG ROD_CLK_START        = 1;
const LONG ROD_CLK_PAUSE        = 2;
const LONG ROD_CLK_RESUME       = 3;
const LONG ROD_CLK_MOVE         = 4;
const LONG ROD_CLK_STOP         = 5;


////////////////////////////////////////////////////////////////////////
// "ControlCursor" Method: "Function" Parameter Constants
////////////////////////////////////////////////////////////////////////

const LONG ROD_CRS_LINE         = 1;
const LONG ROD_CRS_LINE_BLINK   = 2;
const LONG ROD_CRS_BLOCK        = 3;
const LONG ROD_CRS_BLOCK_BLINK  = 4;
const LONG ROD_CRS_OFF          = 5;


////////////////////////////////////////////////////////////////////////
// "SelectCharacterSet" Method: "CharacterSet" Parameter Constants
////////////////////////////////////////////////////////////////////////

const LONG ROD_CS_UNICODE       = 997; // (added in 1.5)
const LONG ROD_CS_ASCII         = 998;
const LONG ROD_CS_WINDOWS       = 999;
const LONG ROD_CS_ANSI          = 999; // (added in 1.5)


////////////////////////////////////////////////////////////////////////
// "TransactionDisplay" Method: "Function" Parameter Constants
////////////////////////////////////////////////////////////////////////

const LONG ROD_TD_TRANSACTION   = 11;
const LONG ROD_TD_NORMAL        = 12;


////////////////////////////////////////////////////////////////////////
// "UpdateVideoRegionAttribute" Method: "Function" Parameter Constants
////////////////////////////////////////////////////////////////////////

const LONG ROD_UA_SET           = 1;
const LONG ROD_UA_INTENSITY_ON  = 2;
const LONG ROD_UA_INTENSITY_OFF = 3;
```

```
const LONG ROD_UA_REVERSE_ON    = 4;
const LONG ROD_UA_REVERSE_OFF   = 5;
const LONG ROD_UA_BLINK_ON      = 6;
const LONG ROD_UA_BLINK_OFF     = 7;


////////////////////////////////////////////////////////////////////
// "EventTypes" Property and "DataEvent" Event: "Status" Parameter
Constants
////////////////////////////////////////////////////////////////////

const LONG ROD_DE_TOUCH_UP      = 0x01;
const LONG ROD_DE_TOUCH_DOWN    = 0x02;
const LONG ROD_DE_TOUCH_MOVE    = 0x04;


////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants for Remote Order Display
////////////////////////////////////////////////////////////////////

const LONG OPOS_EROD_BADCLK     = 201; // ControlClock
const LONG OPOS_EROD_NOCLOCKS   = 202; // ControlClock
const LONG OPOS_EROD_NOREGION   = 203; // RestoreVideo Region
const LONG OPOS_EROD_NOBUFFERS  = 204; // SaveVideoRegion
const LONG OPOS_EROD_NOROOM     = 205; // SaveVideoRegion


#endif              // !defined(OPOSROD_H)
```

OposRfid.h :　RFIDスキャナヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposRfid.h
//
//  RFID Scanner header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------
// 2008-08-30 OPOS Release 1.12                                CRM
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSRFID_H)
#define     OPOSRFID_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "CapWriteTag" Property Constants
////////////////////////////////////////////////////////////////////

const LONG RFID_CWT_NONE                    = 0;
const LONG RFID_CWT_ID                      = 1;
const LONG RFID_CWT_USERDATA                = 2;
const LONG RFID_CWT_ALL                     = 3;


////////////////////////////////////////////////////////////////////
// "CapMultipleProtocols", "CurrentTagProtocol", and
// "ProtocolMask" Property Constants
////////////////////////////////////////////////////////////////////

const LONG RFID_PR_EPC0                     = 0x00000001;
const LONG RFID_PR_0PLUS                    = 0x00000002;
const LONG RFID_PR_EPC1                     = 0x00000004;
const LONG RFID_PR_EPC1G2                   = 0x00000008;
const LONG RFID_PR_EPC2                     = 0x00000010;
const LONG RFID_PR_ISO14443A                = 0x00001000;
const LONG RFID_PR_ISO14443B                = 0x00002000;
const LONG RFID_PR_ISO15693                 = 0x00003000;
const LONG RFID_PR_ISO180006B               = 0x00004000;
const LONG RFID_PR_OTHER                    = 0x01000000;
const LONG RFID_PR_ALL                              = 0x40000000; //
(ProtocolMask only)


////////////////////////////////////////////////////////////////////
// "ReadTags" and "StartReadTags" Methods: "Cmd" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG RFID_RT_ID                       = 0x10;
const LONG RFID_RT_FULLUSERDATA             = 0x01;
const LONG RFID_RT_PARTIALUSERDATA          = 0x02;
const LONG RFID_RT_ID_FULLUSERDATA          = 0x11;
const LONG RFID_RT_ID_PARTIALUSERDATA       = 0x12;


#endif              // !defined(OPOSRFID_H)
```

OposScal.h :　秤ヘッダーファイル

```
///////////////////////////////////////////////////////////////////////
//
// OposScal.h
//
//  Scale header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 1995-12-08 OPOS Release 1.0                                  CRM
// 2005-04-29 OPOS Release 1.9                                  CRM
//  Add StatusNotify constants.
//  Add StatusUpdateEvent constants.
//  Add more ResultCodeExtended constants.
// 2006-03-15 OPOS Release 1.10                                 CRM
//  Corrected names of StatusUpdateEvent constants from 1.9.
// 2013-10-03 OPOS Release 1.14                                 CRM
//  Added FreezeValue's Item constants.
//  Added SetPriceCalculationMode's Mode constants.
//  Added SetSpecialTare's Priority constants.
//  Added SetTarePriority's Mode constants.
// 2025-09-17 OPOS Release 1.15.1                               KF
//      Corrected  missing  define  of  ResultCodeExtended  constant
ESCAL_UNDERWEIHT.
//
///////////////////////////////////////////////////////////////////////

#if !defined(OPOSSCAL_H)
#define     OPOSSCAL_H


#include "Opos.h"


///////////////////////////////////////////////////////////////////////
// "WeightUnit" Property Constants
// "SetUnitPriceWithWeightUnit" Method "WeightUnit" Parameter
//  Constants (added in 1.14)
///////////////////////////////////////////////////////////////////////

const LONG SCAL_WU_GRAM        = 1;
const LONG SCAL_WU_KILOGRAM     = 2;
const LONG SCAL_WU_OUNCE       = 3;
const LONG SCAL_WU_POUND       = 4;


///////////////////////////////////////////////////////////////////////
// "StatusNotify" Property Constants (added in 1.9)
///////////////////////////////////////////////////////////////////////

const LONG SCAL_SN_DISABLED = 1;
const LONG SCAL_SN_ENABLED  = 2;


///////////////////////////////////////////////////////////////////////
// "FreezeValue" Method "Item" Parameter Constants (added in 1.14)
///////////////////////////////////////////////////////////////////////

const LONG SCAL_SFR_MANUAL_TARE        = 0x00000001;
const LONG SCAL_SFR_WEIGHTED_TARE      = 0x00000002;
const LONG SCAL_SFR_PERCENT_TARE       = 0x00000004;
const LONG SCAL_SFR_UNITPRICE          = 0x00000008;


///////////////////////////////////////////////////////////////////////
// "SetPriceCalculationMode" Method "Mode" Parameter Constants
//   (added in 1.14)
///////////////////////////////////////////////////////////////////////

const LONG SCAL_PCM_PRICE_LABELING     = 1;
const LONG SCAL_PCM_SELF_SERVICE       = 2;
const LONG SCAL_PCM_OPERATOR           = 3;
```

```
////////////////////////////////////////////////////////////////////////
// "SetSpecialTare" Method "Mode" Parameter Constants
//   (added in 1.14)
////////////////////////////////////////////////////////////////////////

const LONG SCAL_SST_DEFAULT           = 1;
const LONG SCAL_SST_MANUAL            = 2;
const LONG SCAL_SST_PERCENT           = 3;
const LONG SCAL_SST_WEIGHTED          = 4;


////////////////////////////////////////////////////////////////////////
// "SetTarePriority" Method "Priority" Parameter Constants
//   (added in 1.14)
////////////////////////////////////////////////////////////////////////

const LONG SCAL_STP_FIRST             = 1;
const LONG SCAL_STP_NONE              = 2;


////////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Status" Constants (added in 1.9)
////////////////////////////////////////////////////////////////////////

const LONG SCAL_SUE_STABLE_WEIGHT    = 11;
const LONG SCAL_SUE_WEIGHT_UNSTABLE  = 12;
const LONG SCAL_SUE_WEIGHT_ZERO      = 13;
const LONG SCAL_SUE_WEIGHT_OVERWEIGHT = 14;
const LONG SCAL_SUE_NOT_READY        = 15;
const LONG SCAL_SUE_WEIGHT_UNDER_ZERO = 16;


////////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
////////////////////////////////////////////////////////////////////////

const LONG OPOS_ESCAL_OVERWEIGHT  = 201; // ReadWeight
const LONG OPOS_ESCAL_UNDER_ZERO  = 202; // ReadWeight (added in 1.9)
const LONG OPOS_ESCAL_SAME_WEIGHT = 203; // ReadWeight (added in 1.9)
const LONG OPOS_ESCAL_UNDERWEIGHT = 204; // ReadWeight (added in 1.14)

#endif               // !defined(OPOSSCAL_H)
```

OposScan.h :　スキャナヘッダーファイル

```
////////////////////////////////////////////////////////////////////////
//
// OposScan.h
//
//   Scanner header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 1995-12-08 OPOS Release 1.0                                    CRM
// 1997-06-04 OPOS Release 1.2                                    CRM
// 2004-03-22 OPOS Release 1.8                                    CRM
// 2007-01-30 OPOS Release 1.11                                   CRM
// 2008-08-30 OPOS Release 1.12                                   CRM
//   Add more constants for ScanDataType property.
// 2013-10-03 OPOS Release 1.14                                   CRM
//   Add more constants for ScanDataType property.
// 2025-09-17 OPOS Release 1.15.1                                  KF
//   Add more constants for ScanDataType property.
//
////////////////////////////////////////////////////////////////////////

#if !defined(OPOSSCAN_H)
#define     OPOSSCAN_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////////
// "ScanDataType" Property Constants (added in 1.2)
////////////////////////////////////////////////////////////////////////

// - One dimensional symbologies
const LONG SCAN_SDT_UPCA       = 101;  // Digits
const LONG SCAN_SDT_UPCE       = 102;  // Digits
const LONG SCAN_SDT_JAN8       = 103;  // = EAN 8
const LONG SCAN_SDT_EAN8       = 103;  // = JAN 8 (added in 1.2)
const LONG SCAN_SDT_JAN13      = 104;  // = EAN 13
const LONG SCAN_SDT_EAN13      = 104;  // = JAN 13 (added in 1.2)
const LONG SCAN_SDT_TF         = 105;  // (Discrete 2 of 5) Digits
const LONG SCAN_SDT_ITF        = 106;  // (Interleaved 2 of 5) Digits
const LONG SCAN_SDT_Codabar    = 107;  // Digits, -, $, :, /, ., +;
                                       //   4 start/stop characters
                                       //   (a, b, c, d)
const LONG SCAN_SDT_Code39     = 108;  // Alpha, Digits, Space, -, .,
                                       //   $, /, +, %; start/stop (*)
                                       // Also has Full ASCII feature
const LONG SCAN_SDT_Code93     = 109;  // Same characters as Code 39
const LONG SCAN_SDT_Code128    = 110;  // 128 data characters

const LONG SCAN_SDT_UPCA_S     = 111;  // UPC-A with supplemental
                                       //   barcode
const LONG SCAN_SDT_UPCE_S     = 112;  // UPC-E with supplemental
                                       //   barcode
const LONG SCAN_SDT_UPCD1      = 113;  // UPC-D1
const LONG SCAN_SDT_UPCD2      = 114;  // UPC-D2
const LONG SCAN_SDT_UPCD3      = 115;  // UPC-D3
const LONG SCAN_SDT_UPCD4      = 116;  // UPC-D4
const LONG SCAN_SDT_UPCD5      = 117;  // UPC-D5
const LONG SCAN_SDT_EAN8_S     = 118;  // EAN 8 with supplemental
                                       //   barcode
const LONG SCAN_SDT_EAN13_S    = 119;  // EAN 13 with supplemental
                                       //   barcode

//       The following EAN128 constant deprecated in 1.15.1
//       Instead use the GS1128 constant below.
const LONG SCAN_SDT_EAN128     = 120;  // EAN 128

// - One dimensional symbologies (added in 1.15.1)
const LONG SCAN_SDT_GS1128     = 120;  // GS1 128
```

```
const LONG SCAN_SDT_OCRA       = 121;  // OCR "A"
const LONG SCAN_SDT_OCRB       = 122;  // OCR "B"

// - Two dimensional symbologies (Added in 1.15.1)
const LONG SCAN_SDT_DOTCODE      = 124;  // DotCode

// - One dimensional symbologies (added in 1.8)
//       The following RSS constants deprecated in 1.12.
//       Instead use the GS1DATABAR constants below.
const LONG SCAN_SDT_RSS14      = 131;  // Reduced Space Symbology - 14
digit GTIN
const LONG SCAN_SDT_RSS_EXPANDED= 132;  // RSS - 14 digit GTIN plus
additional fields

// - One dimensional symbologies (added in 1.12)
const LONG SCAN_SDT_GS1DATABAR  = 131;  // GS1 DataBar Omnidirectional
(normal or stacked)
const LONG SCAN_SDT_GS1DATABAR_E= 132;  // GS1 DataBar Expanded (normal
or stacked)

// - One dimensional symbologies (added in Release 1.14)
const LONG SCAN_SDT_ITF_CK              = 133;  // Interleaved 2 of 5
check digit verified and transmitted
const LONG SCAN_SDT_GS1DATABAR_TYPE2    = 134;  // GS1 DataBar Limited
const LONG SCAN_SDT_AMES               = 135;  // Ames Code
const LONG SCAN_SDT_TFMAT              = 136;  // Matrix 2 of 5
const LONG SCAN_SDT_Code39_CK           = 137;  // Code 39 with check
character verified and transmitted
const LONG SCAN_SDT_Code32              = 138;  // Code 39 with Mod 32
check character
const LONG SCAN_SDT_CodeCIP            = 139;  // Code 39 CIP
const LONG SCAN_SDT_TRIOPTIC39         = 140;  // Tri-Optic Code 39
const LONG SCAN_SDT_ISBT128            = 141;  // ISBT-128
const LONG SCAN_SDT_Code11             = 142;  // Code 11
const LONG SCAN_SDT_MSI                = 143;  // MSI Code
const LONG SCAN_SDT_PLESSEY            = 144;  // Plessey Code
const LONG SCAN_SDT_TELEPEN            = 145;  // Telepen

// - Digital Watermarking symbologies (Added in 1.15.1)
const LONG SCAN_SDT_DWCODE             = 146;  // Digital Watermarking

// - Composite Symbologies (added in 1.8)
const LONG SCAN_SDT_CCA        = 151;  // Composite Component A.
const LONG SCAN_SDT_CCB        = 152;  // Composite Component B.
const LONG SCAN_SDT_CCC        = 153;  // Composite Component C.

// - Composite Symbologies (Added in Release 1.14)
const LONG SCAN_SDT_TLC39              = 154;  // TLC-39

// - Two dimensional symbologies
const LONG SCAN_SDT_PDF417     = 201;
const LONG SCAN_SDT_MAXICODE    = 202;

// - Two dimensional symbologies (added in 1.11)
const LONG SCAN_SDT_DATAMATRIX  = 203;  // Data Matrix
const LONG SCAN_SDT_QRCODE     = 204;  // QR Code
const LONG SCAN_SDT_UQRCODE    = 205;  // Micro QR Code
const LONG SCAN_SDT_AZTEC      = 206;  // Aztec
const LONG SCAN_SDT_UPDF417    = 207;  // Micro PDF 417

// - Two dimensional symbologies (Added in Release 1.14)
const LONG SCAN_SDT_GS1DATAMATRIX      = 208;  // GS1 DataMatrix
const LONG SCAN_SDT_GS1QRCODE          = 209;  // GS1 QR Code
const LONG SCAN_SDT_Code49             = 210;  // Code 49
const LONG SCAN_SDT_Code16k            = 211;  // Code 16K
const LONG SCAN_SDT_CodablockA         = 212;  // Codablock A
const LONG SCAN_SDT_CodablockF         = 213;  // Codablock F
const LONG SCAN_SDT_Codablock256       = 214;  // Codablock 256
const LONG SCAN_SDT_HANXIN             = 215;  // Han Xin Code

// - Postal Code Symbologies (Added in Release 1.14)
const LONG SCAN_SDT_AusPost            = 301;  // Australian Post
const LONG SCAN_SDT_CanPost            = 302;  // Canada Post
const LONG SCAN_SDT_ChinaPost          = 303;  // China Post
const LONG SCAN_SDT_DutchKix           = 304;  // Dutch Post
```

```
const LONG SCAN_SDT_InfoMail          = 305;  // InfoMail
const LONG SCAN_SDT_JapanPost         = 306;  // Japan Post
const LONG SCAN_SDT_KoreanPost        = 307;  // Korean Post
const LONG SCAN_SDT_SwedenPost        = 308;  // Sweden Post
const LONG SCAN_SDT_UkPost           = 309;  // UK Post BPO
const LONG SCAN_SDT_UsIntelligent      = 310;  // US Intelligent Mail
const LONG SCAN_SDT_UsPlanet          = 311;  // US Planet Code
const LONG SCAN_SDT_PostNet           = 312;  // US Postnet

// - Special cases
const LONG SCAN_SDT_OTHER       = 501;  // Start of Scanner-Specific
bar
                                  //   code symbologies
const LONG SCAN_SDT_UNKNOWN    =   0;  // Cannot determine the barcode
                                  //   symbology.


#endif               // !defined(OPOSSCAN_H)
```

OposScrw.h : スマートカードリーダライタヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposScrw.H
//
//  Smart Card Reader Writer header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------
// 2004-03-22 OPOS Release 1.8                                CRM
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSSCRW_H)
#define     OPOSSCRW_H


#include "opos.h"


////////////////////////////////////////////////////////////////////
// "CapInterfaceMode" Property Constants
////////////////////////////////////////////////////////////////////

const LONG SC_CMODE_TRANS       =   1;  // Simple Transaction Command
and Data Mode.
const LONG SC_CMODE_BLOCK      =   2;  // Block Data Mode.
const LONG SC_CMODE_APDU         =   4;  // Same as Block Data Mode
except APDU Standard Commands are used.
const LONG SC_CMODE_XML        =   8;  // XML Block Data Mode.


////////////////////////////////////////////////////////////////////
// "CapIsoEmvMode" Property Constants
////////////////////////////////////////////////////////////////////

const LONG SC_CMODE_ISO            =    1;   // APDU messaging format
conforms to the ISO standard.
const LONG SC_CMODE_EMV            =    2;   // APDU messaging format
conforms to the EMV standard.


////////////////////////////////////////////////////////////////////
// "CapTransmissionProtocol" Property Constants
////////////////////////////////////////////////////////////////////

const LONG SC_CTRANS_PROTOCOL_T0=   1;  // Asynchronous, Half Duplex,
Character, Transmission Protocol Mode.
const LONG SC_CTRANS_PROTOCOL_T1=   2;  // Asynchronous, Half Duplex,
Block Transmission Protocol Mode.


////////////////////////////////////////////////////////////////////
// "InterfaceMode" Property Constants
////////////////////////////////////////////////////////////////////

const LONG SC_MODE_TRANS        =   1;  // Simple Transaction Command
and Data Mode.
const LONG SC_MODE_BLOCK       =   2;  // Block Data Mode.
const LONG SC_MODE_APDU          =   4;  // Same as Block Data Mode
except APDU Standard Defines the Commands and data.
const LONG SC_MODE_XML         =   8;  // XML Block Data Mode.


////////////////////////////////////////////////////////////////////
// "IsoEmvMode" Property Constants
////////////////////////////////////////////////////////////////////

const LONG SC_MODE_ISO             =    1;   // APDU messaging format
currently in use conforms to the ISO standard.
const LONG SC_MODE_EMV             =    2;   // APDU messaging format
currently in use conforms to the EMV standard.
```

```
////////////////////////////////////////////////////////////////////
// "TransmissionProtocol" Property Constants
////////////////////////////////////////////////////////////////////

const LONG SC_TRANS_PROTOCOL_T0 =   1;  // Asynchronous, Half Duplex,
Character, Transmission Protocol Mode.
const LONG SC_TRANS_PROTOCOL_T1 =   2;  // Asynchronous, Half Duplex,
Block Transmission Protocol Mode.


////////////////////////////////////////////////////////////////////
// "ReadData" Method: "Action" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG SC_READ_DATA           =  11;
const LONG SC_READ_PROGRAM        =  12;
const LONG SC_EXECUTE_AND_READ_DATA =  13;
const LONG SC_XML_READ_BLOCK_DATA   =  14;


////////////////////////////////////////////////////////////////////
// "WriteData" Method: "Action" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG SC_STORE_DATA       =  21;
const LONG SC_STORE_PROGRAM    =  22;
const LONG SC_EXECUTE_DATA     =  23;
const LONG SC_XML_BLOCK_DATA   =  24;
const LONG SC_SECURITY_FUSE    =  25;
const LONG SC_RESET           =  26;


////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constant
////////////////////////////////////////////////////////////////////

const LONG SC_SUE_NO_CARD      =   1;  // No card detected in the SCR/W
Device.
const LONG  SC_SUE_CARD_PRESENT   =    2;   // There is a card in the
device.


////////////////////////////////////////////////////////////////////
// "ErrorEvent" Event: "ResultCodeExtended" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_ESC_READ       = 201;  // There was a read error.
const LONG OPOS_ESC_WRITE      = 202;  // There was a write error.
const  LONG  OPOS_ESC_TORN            = 203;   // The smart card was
prematurely    removed    from    the    SCR/W    unexpectedly.    Note:
CapCardErrorDetection capability must be true before this can be set.
const LONG OPOS_ESC_NO_CARD     = 204;  // There is no card detected
in the SCR/W but a card was expected.


#endif               // !defined(OPOSSCRW_H)
```

OposSig.h :　シグニチャ・キャプチャヘッダーファイル

```
///////////////////////////////////////////////////////////////////////
//
// OposSig.h
//
//   Signature Capture header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 1995-12-08 OPOS Release 1.0                                  CRM
//
///////////////////////////////////////////////////////////////////////

#if !defined(OPOSSIG_H)
#define     OPOSSIG_H


#include "Opos.h"


// No definitions required for this version.


#endif               // !defined(OPOSSIG_H)
```

OposSpch.h :　音声合成ヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposSpch.h
//
//   Speech Synthesis header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------
// 2021-01-26 OPOS Release 1.16                              KF
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSSPCH_H)
#define     OPOSSPCH_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG SPCH_SUE_START_SPEAK             =11;
const LONG SPCH_SUE_STOP_SPEAK              =12;


#endif               // !defined(OPOSSPCH_H)
```

OposSply.h :　サウンドプレーヤヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposSply.h
//
//  Sound Player header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------
// 2021-01-26 OPOS Release 1.16                              KF
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSSPLY_H)
#define     OPOSSPLY_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "CapStorage", "Storage" Property Constants
////////////////////////////////////////////////////////////////////

const LONG SPLY_CST_HOST_ONLY         = 1;
const LONG SPLY_CST_HARDTOTALS_ONLY    = 2;
const LONG SPLY_CST_ALL               = 3;

const LONG SPLY_ST_HOST               = 1;
const LONG SPLY_ST_HARDTOTALS          = 2;
const LONG SPLY_ST_HOST_HARDTOTALS     = 3;


////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG SPLY_SUE_START_PLAY_SOUND        =11;
const LONG SPLY_SUE_STOP_PLAY_SOUND         =12;


#endif               // !defined(OPOSSPLY_H)
```

OposSrec.h：　サウンドレコーダヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposSrec.h
//
//  Sound Recorder header file for OPOS Applications.
//
// Modification history
// ------------------------------------------------------------------
// 2021-01-26 OPOS Release 1.16                              KF
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSSREC_H)
#define     OPOSSREC_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "CapStorage", "Storage" Property Constants
////////////////////////////////////////////////////////////////////

const LONG SREC_CST_HOST_ONLY      = 1;
const LONG SREC_CST_HARDTOTALS_ONLY = 2;
const LONG SREC_CST_ALL            = 3;

const LONG SREC_ST_HOST            = 1;
const LONG SREC_ST_HARDTOTALS      = 2;
const LONG SREC_ST_HOST_HARDTOTALS  = 3;


////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG SREC_SUE_START_SOUND_RECORDING  = 11;
const LONG SREC_SUE_STOP_SOUND_RECORDING   = 12;


////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
////////////////////////////////////////////////////////////////////

const LONG OPOS_ESREC_NOROOM       = 201;


#endif               // !defined(OPOSSREC_H)
```

OposStat.h : スタティスティック名ヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposStat.h
//
//  Statistic names header file for OPOS Applications.
//
// Modification history
// ------------------------------------------------------------------
// 2004-07-14 OPOS Release 1.8.1                                CRM
//  New file to define constants for Device Statistic names.
// 2007-01-30 OPOS Release 1.11 (Add 1.10 and 1.11 constants)    CRM
// 2008-08-30 OPOS Release 1.12 (Add 1.12 constants)            CRM
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSSTAT_H)
#define     OPOSSTAT_H

// Common Statistics for all Device Categories.
#define                                    OPOS_STAT_HoursPoweredCount
 T("HoursPoweredCount")
#define                                 OPOS_STAT_CommunicationErrorCount
_T("CommunicationErrorCount")

// Statistics for one or more Device Categories.
#define OPOS_STAT_AverageFAR                      _T("AverageFAR")
#define OPOS_STAT_AverageFRR                     _T("AverageFRR")
#define                                  OPOS_STAT_BarcodePrintedCount
 T("BarcodePrintedCount")
#define OPOS_STAT_BumpCount                       _T("BumpCount")
#define                                OPOS_STAT_ChallengeRequestCount
 T("ChallengeRequestCount")
#define                                   OPOS_STAT_ChecksScannedCount
 T("ChecksScannedCount")
#define                                OPOS_STAT_DrawerFailedOpenCount
 T("DrawerFailedOpenCount")
#define                                  OPOS_STAT_DrawerGoodOpenCount
 T("DrawerGoodOpenCount")
#define OPOS_STAT_EraseCount                      _T("EraseCount")
#define                     OPOS_STAT_FailedCardAuthenticationDataCount
 T("FailedCardAuthenticationDataCount")
#define                                   OPOS_STAT_FailedDataParseCount
 T("FailedDataParseCount")
#define                          OPOS_STAT_FailedDeviceAuthenticationCount
 T("FailedDeviceAuthenticationCount")
#define                                  OPOS_STAT_FailedPaperCutCount
 T("FailedPaperCutCount")
#define                               OPOS_STAT_FailedPrintSideChangeCount
 T("FailedPrintSideChangeCount")
#define                                      OPOS_STAT_FailedReadCount
 T("FailedReadCount")
#define                                 OPOS_STAT_FailedSignatureReadCount
 T("FailedSignatureReadCount")
#define                                  OPOS_STAT_FailedTagDisableCount
 T("FailedTagDisableCount")
#define                                     OPOS_STAT_FailedTagLockCount
 T("FailedTagLockCount")
#define                                   OPOS_STAT_FailedTagWriteCount
 T("FailedTagWriteCount")
#define                                      OPOS_STAT_FailedWriteCount
 T("FailedWriteCount")
#define                                     OPOS_STAT_FormInsertionCount
 T("FormInsertionCount")
#define                           OPOS_STAT_GoodCardAuthenticationDataCount
 T("GoodCardAuthenticationDataCount")
#define                             OPOS_STAT_GoodDeviceAuthenticationCount
_T("GoodDeviceAuthenticationCount")
#define OPOS_STAT_GoodReadCount                    _T("GoodReadCount")
#define OPOS_STAT_GoodScanCount                   _T("GoodScanCount")
#define                                  OPOS_STAT_GoodSignatureReadCount
_T("GoodSignatureReadCount")
```

```
#define                                    OPOS_STAT_GoodTagDisableCount
 _T("GoodTagDisableCount")
#define                                       OPOS_STAT_GoodTagLockCount
 _T("GoodTagLockCount")
#define                                     OPOS_STAT_GoodTagWriteCount
 _T("GoodTagWriteCount")
#define                                    OPOS_STAT_GoodWeightReadCount
 _T("GoodWeightReadCount")
#define OPOS_STAT_GoodWriteCount                    _T("GoodWriteCount")
#define OPOS_STAT_HomeErrorCount                    _T("HomeErrorCount")
#define                                  OPOS_STAT_InvalidPINEntryCount
 _T("InvalidPINEntryCount")
#define                           OPOS_STAT_JournalCharacterPrintedCount
 _T("JournalCharacterPrintedCount")
#define                                OPOS_STAT_JournalLinePrintedCount
 _T("JournalLinePrintedCount")
#define                                        OPOS_STAT_KeyPressedCount
 _T("KeyPressedCount")
#define                                OPOS_STAT_LockPositionChangeCount
 _T("LockPositionChangeCount")
#define                                OPOS_STAT_MaximumTempReachedCount
 _T("MaximumTempReachedCount")
#define                                        OPOS_STAT_MediumFreeSpace
 _T("MediumFreeSpace")
#define                                     OPOS_STAT_MediumRemovedCount
 _T("MediumRemovedCount")
#define OPOS_STAT_MediumSize                        _T("MediumSize")
#define                        OPOS_STAT_MissingStartSentinelTrack1Count
 _T("MissingStartSentinelTrack1Count")
#define                        OPOS_STAT_MissingStartSentinelTrack2Count
 _T("MissingStartSentinelTrack2Count")
#define                        OPOS_STAT_MissingStartSentinelTrack3Count
 _T("MissingStartSentinelTrack3Count")
#define                        OPOS_STAT_MissingStartSentinelTrack4Count
 _T("MissingStartSentinelTrack4Count")
#define                                       OPOS_STAT_MotionEventCount
 _T("MotionEventCount")
#define OPOS_STAT_NoReadCount                       _T("NoReadCount")
#define                                      OPOS_STAT_NVRAMWriteCount
 _T("NVRAMWriteCount")
#define                                  OPOS_STAT_OnlineTransitionCount
 _T("OnlineTransitionCount")
#define OPOS_STAT_PaperCutCount                     _T("PaperCutCount")
#define                                 OPOS_STAT_ParityLRCErrorTrack1Count
 _T("ParityLRCErrorTrack1Count")
#define                                 OPOS_STAT_ParityLRCErrorTrack2Count
 _T("ParityLRCErrorTrack2Count")
#define                                 OPOS_STAT_ParityLRCErrorTrack3Count
 _T("ParityLRCErrorTrack3Count")
#define                                 OPOS_STAT_ParityLRCErrorTrack4Count
 _T("ParityLRCErrorTrack4Count")
#define                                        OPOS_STAT_PrinterFaultCount
 _T("PrinterFaultCount")
#define                                   OPOS_STAT_PrintSideChangeCount
 _T("PrintSideChangeCount")
#define                           OPOS_STAT_ReceiptCharacterPrintedCount
 _T("ReceiptCharacterPrintedCount")
#define                                  OPOS_STAT_ReceiptCoverOpenCount
 _T("ReceiptCoverOpenCount")
#define                                   OPOS_STAT_ReceiptLineFeedCount
 _T("ReceiptLineFeedCount")
#define                                OPOS_STAT_ReceiptLinePrintedCount
 _T("ReceiptLinePrintedCount")
#define OPOS_STAT_SessionCount                      _T("SessionCount")
#define                                 OPOS_STAT_SlipCharacterPrintedCount
 _T("SlipCharacterPrintedCount")
#define                                    OPOS_STAT_SlipCoverOpenCount
 _T("SlipCoverOpenCount")
#define                                      OPOS_STAT_SlipLineFeedCount
 _T("SlipLineFeedCount")
#define                                   OPOS_STAT_SlipLinePrintedCount
 _T("SlipLinePrintedCount")
#define                                        OPOS_STAT_StampFiredCount
_T("StampFiredCount")
```

```
#define                                    OPOS_STAT_SuccessfulMatchCount
 _T("SuccessfulMatchCount")
#define OPOS_STAT_TagReadCount                        _T("TagReadCount")
#define                                       OPOS_STAT_ToneSoundedCount
 _T("ToneSoundedCount")
#define                                     OPOS_STAT_UnreadableCardCount
 _T("UnreadableCardCount")
#define                                  OPOS_STAT_UnsuccessfulMatchCount
 _T("UnsuccessfulMatchCount")
#define                                     OPOS_STAT_ValidPINEntryCount
 _T("ValidPINEntryCount")
#define OPOS_STAT_WriteCount                          _T("WriteCount")

#endif               // !defined(OPOSSTAT_H)
```

OposTone.h :　トーンインジケータヘッダーファイル

```
////////////////////////////////////////////////////////////////////////
//
// OposTone.h
//
//  Tone Indicator header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 1997-06-04 OPOS Release 1.2                                  CRM
// 2009-10-02 OPOS Release 1.13                                 CRM
//   Add MelodyType constants.
//
////////////////////////////////////////////////////////////////////////

#if !defined(OPOSTONE_H)
#define     OPOSTONE_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////////
// "MelodyType" Property Constants (added in 1.13
////////////////////////////////////////////////////////////////////////

const LONG TONE_MT_NONE  = 0;
const LONG TONE_MT_TYPE1 = 1;
const LONG TONE_MT_TYPE2 = 2;
const LONG TONE_MT_TYPE3 = 3;
const LONG TONE_MT_TYPE4 = 4;
const LONG TONE_MT_TYPE5 = 5;


#endif               // !defined(OPOSTONE_H)
```

OposTot.h :　ハードトータルヘッダーファイル

```
///////////////////////////////////////////////////////////////////////
//
// OposTot.h
//
//  Hard Totals header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------------
// 1995-12-08 OPOS Release 1.0                                  CRM
//
///////////////////////////////////////////////////////////////////////

#if !defined(OPOSTOT_H)
#define     OPOSTOT_H


#include "Opos.h"


///////////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
///////////////////////////////////////////////////////////////////////

const LONG OPOS_ETOT_NOROOM    = 201; // Create, Write
const LONG OPOS_ETOT_VALIDATION = 202; // Read, Write


#endif               // !defined(OPOSTOT_H)
```

OposVcap.h :　ビデオキャプチャヘッダーファイル

```
///////////////////////////////////////////////////////////////////
//
// OposVcap.h
//
//  Video Capture header file for OPOS Applications.
//
// Modification history
// ------------------------------------------------------------------
// 2021-01-26 OPOS Release 1.16                              KF
//
///////////////////////////////////////////////////////////////////

#if !defined(OPOSVCAP_H)
#define     OPOSVCAP_H


#include "Opos.h"


///////////////////////////////////////////////////////////////////
// "CapStorage", "Storage" Property Constants
///////////////////////////////////////////////////////////////////

const LONG VCAP_CST_HOST_ONLY      = 1;
const LONG VCAP_CST_HARDTOTALS_ONLY = 2;
const LONG VCAP_CST_ALL            = 3;

const LONG VCAP_ST_HOST            = 1;
const LONG VCAP_ST_HARDTOTALS      = 2;
const LONG VCAP_ST_HOST_HARDTOTALS  = 3;


///////////////////////////////////////////////////////////////////
// "VideoCaptureMode" Property Constants
///////////////////////////////////////////////////////////////////

const LONG VCAP_VCMODE_PHOTO       = 1;
const LONG VCAP_VCMODE_VIDEO       = 2;


///////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constants
///////////////////////////////////////////////////////////////////

const LONG VCAP_SUE_START_PHOTO          = 11;
const LONG VCAP_SUE_END_PHOTO            = 12;

const LONG VCAP_SUE_START_VIDEO_RECORDING  = 21;
const LONG VCAP_SUE_STOP_VIDEO_RECORDING   = 22;


///////////////////////////////////////////////////////////////////
// "ResultCodeExtended" Property Constants
///////////////////////////////////////////////////////////////////

const LONG OPOS_EVCAP_NOROOM       = 201; // (Several)


#endif               // !defined(OPOSVCAP_H)
```

OposVrcg.h：　音声認識ヘッダーファイル

```
////////////////////////////////////////////////////////////////////
//
// OposVrcg.h
//
//  Voice Recognition header file for OPOS Applications.
//
// Modification history
// ---------------------------------------------------------------
// 2021-01-26 OPOS Release 1.16                              KF
//
////////////////////////////////////////////////////////////////////

#if !defined(OPOSVRCG_H)
#define     OPOSVRCG_H


#include "Opos.h"


////////////////////////////////////////////////////////////////////
// "HearingResult" Property Constants
////////////////////////////////////////////////////////////////////

const LONG VRCG_HRESULT_FREE                = 11;
const LONG VRCG_HRESULT_SENTENCE             = 21;
const LONG VRCG_HRESULT_WORD                = 31;
const LONG VRCG_HRESULT_YESNO_YES            = 41;
const LONG VRCG_HRESULT_YESNO_NO             = 42;
const LONG VRCG_HRESULT_YESNO_CANCEL         = 43;


////////////////////////////////////////////////////////////////////
// "HearingStatus" Property Constants
////////////////////////////////////////////////////////////////////

const LONG VRCG_HSTATUS_NONE                =  0;
const LONG VRCG_HSTATUS_FREE                = 10;
const LONG VRCG_HSTATUS_SENTENCE             = 20;
const LONG VRCG_HSTATUS_WORD                = 30;
const LONG VRCG_HSTATUS_YESNO               = 40;


////////////////////////////////////////////////////////////////////
// "StatusUpdateEvent" Event: "Data" Parameter Constants
////////////////////////////////////////////////////////////////////

const LONG VRCG_SUE_STOP_HEARING             =  0;
const LONG VRCG_SUE_START_HEARING_FREE        = 10;
const LONG VRCG_SUE_START_HEARING_SENTENCE    = 20;
const LONG VRCG_SUE_START_HEARING_WORD        = 30;
const LONG VRCG_SUE_START_HEARING_YESNO       = 40;


#endif               // !defined(OPOSVRCG_H)
```

付　録　C
技術的詳細説明

システム文字列(BSTR)

システム文字列の特徴

OPOSは、OLEシステム文字列を使用して、可変長のデータを渡したり、戻したりします。システム文字列はしばしばBStringとして参照され、Microsoft Visual C++がBSTRタイプを割り当てます。

システム文字列は一連のUnicode文字からなり、各Unicode文字は16ビット幅です。そのため、「ワイド」文字とも言います。この文字列の後にはヌル文字つまりゼロが1つ続きます。また、文字列のロングバイトカウントが符号なしで先行しますが、ヌル文字は含みません。このカウントを2で割って、文字列の文字数を割り出します。

ほとんどの場合、OPOSはシステム文字列を使用して、アプリケーション、コントロールオブジェクト、およびシステムオブジェクトの間で文字データを行き来させます。システム文字列(BSTR)は、メソッドによる文字列パラメータの受け渡しや、文字列をプロパティに返すために使用されます。メソッドが文字列データを戻さなければならない場合、システム文字列のポインタ(BSTR*)をメソッドパラメータとして使用します。

システム文字列の使用法

Visual Basic では、難なくシステム文字列の受け渡しが行えます。VBにおける文字列の内部表示は、長さを要素として持つワイド文字です。変数、文字列式、またはリテラルを使用して、BSTRを渡すことができます。データをメソッドで変更できるよう、BSTR*では変数として使用する必要があります。

同様に、ATLを使用するVisual C++も容易に行えます。BSTRとBSTR*データは、これらの型を利用して受け渡しができます。Unicodeとの相互変換は開発者の責任です。

しかし、MFCを伴うVisual C++ ではそれ以上の考慮が必要です。

BSTRは以下のように取り扱います。

BSTRメソッドパラメータ

- 呼び出し機能 : BSTRパラメータを持つオートメーションメソッドの呼び出しは、文字列ポインタLPCTSTRとしてMFCが処理します。VC++ ANSIオプションを使用すると、MFCがANSIからUnicodeへ自動的に変換を行います。
- 被呼び出し機能：オートメーションメソッドをインプリメントする機能は、文字列ポインタLPCTSTRとしてBSTRパラメータを受けとります。VC++ ANSIオプションを使用すると、制御をこの機能に渡す前に、MFCがUnicodeからANSIへの自動変換を実行します。文字列長が文字列ポインタの直前に存在します。

BSTR戻りタイプ（プロパティの入手に使用）

- 呼び出し機能：BSTR結果を戻すオートメーションメソッドは、MFCによって自動的にCStringに変換されます。
- 被呼び出し機能：オートメーションメソッドは、MFC CStringオブジェクトにデータを入れ、CStringの"AllocSysString"メンバ関数の結果を戻すことによって、BSTR結果を戻します。VC++ ANSIオプションを使用すると、この機能は文字列をANSIからUnicodeに自動的に変換します。

BSTR*はMFCによりBSTR*として受け渡され、そのために開発者の取り扱いはATLと同様になります。あるMFCマクロとクラスは有用です。

VC++のANSIオプションを利用すれば、UnicodeとMBCSの間の変換が可能です。T2OLE、OLE2Tなど、この変換が容易になるマクロを利用することもできます。（ただし、これらは文字列に挿入されたヌル文字を処理しません。）

文字列を設定するには、データをMFC CStringオブジェクトに入れ、CStringの"SetSysSting"メンバ関数を使用します。

システム文字列とバイナリデータ

OPOSは、BSTRとBSTR*を使用してバイナリデータを渡すことがあります。

これらのケースでは、16進の00からFFまでの範囲のバイトデータを戻します。システム文字列の各16ビット文字は、下位8ビットに1バイトのバイナリデータを含みます。上位8ビットはゼロです。これにより、プログラム的に2つのケースが発生します。

ヌル文字つまりゼロの場合。システム文字列に長さの要素があっても、ヌル文字によって文字列の終りを決めてしまうソフトウェアがあるからです。

0x80-0xFFの範囲の文字の場合。ANSIとUnicodeフォーマット間での変換で、特に極東の言語では、不正なデータが生じる場合があります。

変換と転送の問題を回避するためには、0x01から0x7Fの範囲外のデータをメソッドパラメータかプロパティで受け渡しをするなら、アプリケーションは**BinaryConversion**を利用すべきです。第1.2版で追加された**BinaryConversion**は、バイナリとASCIIフォーマット間で2通りの変換手段を提供します。

キャラクターセットのマッピング　　第1.10版で変更

ここでは、LineDisplay、POSPinter、PointCardReaderWriter、RemortOrderDisplayのようないデバイスで提供されているMapCharacterSetプロパティの適切な使い方の詳細を提供します。はじめに、アプリケーションは、サービスオブジェクトのCharacterSetプロパティに該当するデバイスの文字セットを選択しなければなりません。次に、使用するUnicode文字セットをサービスオブジェクトに文字列で渡します。その時に、サービスオブジェクトは、必要に応じて、デバイスのコードページにUnicodeキャラクタを割り当てたことを返します。

以下のコードによりサービスオブジェクトプロバイダーが簡単にそのサービスに対しマッピングメカニズムを加える事ができる。（デバイスに出力するためにサービスに転送されるデータは既にBSTRからLPCTSTRへ変換済みであると仮定される。）

```
BOOL AnsiTo0EMCCodePage(
    UNIT CodePage,  //the desired destination code page like 858
    LPCTSTR srg,    // source string assumed to be ACP (default
                    // system code page)
    INT srcLength,  // the length of the source string
    LPCTR dest,     // destination String; when called 'dest'
                    // shows to a reserved area of 'destLength'-
    INT *destLength  // bytes length of the destination string
{
    LPWSTR lpWideCharStr = NULL;
    INT WideCharStrLen = (srcLength+1)* sizeof (lpWideCharStr[0];
    lpWideCharStr = (LPWSTR) malloc (WideCharStrLen)

    if (lpWideCharStr == NULL)
        return FALSE;

    // convert to UNICODE
    WideCharStrLen = MultiByteToWideChar (CP_ACP, 0 ,src, srcLength,
        lpWideCharStr, WideCharStrLen);
    if (WideCharStrLen<= 0)
    {
        free (lpWideCharStr);
        return FALSE;
    }
    // convert UNICODE back to desired codepage ;
    // non mappable characters are mapped to space character
    const char defaultChar = 0✕20;
    *destLength = WideCharToMultiByte (CodePage, 0 , lpWideCharStr,
                    WideCharStrLen, dest, *destLength, &defaultChar, NULL);
    free (lpWideCharStr) ;
    if (*destLength == 0 && WideCharStrLen != 0) // cp does not exist
         return FALSE;
    return TRUE;
}
```

注：

現在システムが選択しているコードページは以下のレジストリに掲載：HKLM＼System＼CurrentControlSet＼Control＼NIs＼Codepage ACP.

マッピングのためにシステムAPIコールを使用の際にはデスティネーションコードページをインストールしなければならない。

付 録 D

第1.5版でのAPI変更：ClaimDeviceとReleaseDevice

Claimと**Release**の共通メソッドは最初のOPOSリリースで定義されました。その時以降、OPOS **Release**メソッドと、全てのCOMオブジェクトのメソッドとして使用されるCOM **Release**メソッドとの間で、競合が増加してきました。この競合により、開発に当たり幾つかの制限が発生しています。

コントロールオブジェクトとサービスオブジェクトは、純粋なディスパッチインタフェースで互いのインタフェースを定義する必要があります。ATLはデュアルインタフェースの実装を使用してのみIDispatchをサポートしているので、Microsoft Visual C++のActive Template Libraryを使用できなくなります。

ある開発環境では、COMと競合するディスパッチメソッドをActiveXコントロールは定義できないと仮定しています。例えば、Delphiのユーザーは**Release**の競合を回避して作業する必要がありました。将来のツールでは、この競合に対する許容範囲はより小さくなるでしょう。

このため、第1.5版ではこれらのメソッドは**ClaimDevice**と**ReleaseDevice**に名称が変更されました。

アプリケーションとサービスオブジェクトの大部分を移植するために、いくつか処理が必要です。これらは、”コモンコントロールオブジェクト”として知られるコントロールオブジェクトのレファレンスセットに含まれています。

アプリケーション

ClaimDeviceと**Claim**メソッド、**ReleaseDevice**と**Release** メソッドの両者はコントロールオブジェクトのIDispatchインタフェースでサポートされます。IDispatchインタフェースはレイトバインディングをアプリケーションで実装する時に使用されます。この動作により、現在使用されている従来アプリケーションに完全な互換性を提供します。

（Microsoft Visual C++やVisual Basicを含む）アーリーバインディングを実行する開発環境を利用するアプリケーションが、第1.4版以前のコントロールオブジェクトから第1.5版以降のコントロールオブジェクトに移行する際、**Claim**呼び出しから **ClaimDevice**へ、**Release** 呼び出しから **ReleaseDevice**へと修正しなければなりません。

サービスオブジェクト

サービスオブジェクトは、**Claim**か **ClaimDevice**かいずれかのソッド、**Release**か**ReleaseDevice**かいずれかのメソッドをIDispatch インタフェースで公開します。サービスオブジェクトがATLで実装されているなら、**Release**はCOMのIUnknwonの参照カウンタ更新に使用されているので、**ReleaseDevic**eを使用しなくてはなりません。

アプリケーションが**ClaimDevice**か**Claim**を呼び出した時、コントロールオブジェクトはサービスオブジェクトに**ClaimDevice**が実装されているなら**ClaimDevice**を呼び出し、実装されていなければ**Claim**を呼び出します。アプリケーションが**ReleaseDevice**か**Release**を呼び出した時、コントロールオブジェクトはサービスオブジェクトに**ReleaseDevice**が実装されているなら**ReleaseDevice**を呼び出し、実装されていなければ**Release**を呼び出します。この動作により、新たなサービスオブジェクトに対してATLを利用した実装を許容しながら、従来のサービスオブジェクトに対しても完全な互換性を提供します。

付　録　E
修正履歴
第1.1.1版

第1.1.1版は、米国OLE for Retail POS Committeeにおいて標準化された

“Application 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。

第1.1.1版では、第1.0版からのプロパティやメソッドの追加、削除はありませんが、ヘッダーファイルのシンボル定義値の修正、削除やプロパティの属性の変更、メソッドの制御方式の変更等が行われいます。アプリケーションの作成にあたっては、OPOSコントロールの対応リリースバージョンに十分、注意が必要です。

以下に第1.1.1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、はじめに	米国Application　Programmer’s　Guideの対応リリース番号を、1.01から1.1に変更
はしがき	著作権の記述の修正
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 メソッドとプロパティの使い方	プロパティ初期化後の再初期化についての記述を追加と一覧参照の説明の削除
はじめに デバイス共有モデルについて	備考欄に排他使用デバイスと規定している説明の追加
はじめに　　OPOS コントロール解説	コインディスペンサ，MICR，秤，シグニチャ・キャプチャを追加．付録Ｃの説明文を追加
共通プロパティ、メソッド、イベント	**ResultCodeExtended**プロパティに設定される内容についての明確化
各デバイスの概説	コントロール名の呼び方を、“OLEプログラマティクID”に統一して記述
各デバイスの概説	“機能”、”モデル”、”デバイスの共有”のサブタイトルの埋め込み。

ドロワーの排他制御権獲得時の動作	排他制御権を獲得したアプリケーションのみが、呼び出せるメソッドの明確化
ラインディスプレィの**CapCharacterSet**プロパティ	設定される内容についての説明の明確化
ラインディスプレィの**CursorColumn**プロパティ	有効値の上限値の誤記の修正。**(Columns)**から**(Columns-1)**に修正
ラインディスプレィの**CursorUpdate**プロパティ	本プロパティが、各ウィンドウ毎に管理されることを追加。
ラインディスプレィの**InterCharacterWait**プロパティ	キュー管理されている表示要求と本プロパティが0に設定されたときの表示処理についての説明を追加。
ラインディスプレィの**MarqueeFormat**プロパティ	本プロパティの参照される契機についての説明を追加。
ラインディスプレィの**MarqueeFormat**プロパティ	本プロパティの値が、DISP_MF_WALKの場合の詳細な処理内容についての説明を追加。
ラインディスプレィの**MarqueeUnitWait**プロパティ	本プロパティについて参照すべき項目に、**MarqueeFormat**プロパティを追加。
ラインディスプレィの**ClearText**メソッド	**ResultCode**プロパティに格納される値に、OPOS_E_ILLEGALを追加。
ラインディスプレィの**CreatWindow**メソッド	形式とパラメータ説明の文字列の誤記の修正。
ラインディスプレィの**CreatWindow**メソッド	有効なウィンドウサイズの説明の追加と初期化されるウィンドウ毎のプロパティに**MarqueeFormat**プロパティを追加。
ラインディスプレィの**DisplayText**メソッド	**ResultCode**プロパティに格納される値に、OPOS_E_ILLEGALを追加。
ラインディスプレィの**DisplayTextAt**メソッド	**ResultCode**プロパティに格納される値の、OPOS_E_ILLEGALの誤記を修正。
ラインディスプレィの**RefreshWindow**メソッド	**ResultCode**プロパティに格納される値の、OPOS_E_ILLEGALの誤記を修正。
ラインディスプレィの**ScrollText**メソッド	**ResultCode**プロパティに格納される値の、OPOS_E_ILLEGALの誤記を修正。
ハードトータルの概説	名称の誤記の修正のため、**ClearAll**メソッド名の記述を**SetAll**と修正。
ハードトータルの**Write**メソッド	誤記修正のため、**FreeSpace**を**FreeData**に変更。

キーロックの概説	デバイスの共有についての記述における**Claim**メソッドの実行結果の戻り値を、OPOS_SUCCESSからOPOS_E_ILLEGALへ修正。
磁気ストライプリーダの概説のモデル	誤記修正のため、**Data**パラメータを**Status**パラメータに修正
磁気ストライプリーダの**ParseDecodeData**プロパティ	名称の誤記の修正のため、**ParseDecodedData**を、**ParseDecodeData**に変更。
POSプリンタのプロパティ一覧	**CapCharacterSet**プロパティの型の誤記の修正のため、BooleanをLongに変更。
POSプリンタのプロパティ一覧	**RotateSpecial**プロパティの初期化条件を、”Open”のみに修正。
POSプリンタのプロパティ一覧	**JrnLineWidth**、 **RecLineWidth**、 **SlpLineWidth**の各プロパティのアクセスタイプの誤記を修正。R/WをＲに修正。
POSプリンタの概説	モデルの説明における、**Asyncmode**プロパティに依存するメソッドの誤記の修正のため、**TransactionPrint**メソッドを削除。
POSプリンタの概説	デバイスの共有の説明におけるメソッド呼び出しの前の条件の明確化のため記述を追加。
POSプリンタの印字データとエスケープシーケンス	印刷データの有効性チェックに**ValidateData**メソッドを使用する事の説明を追加。
POSプリンタの印字データとエスケープシーケンス	縦倍率、横倍率指定のエスケープシーケンスの名称と内容の誤記を修正。
POSプリンタの**AsyncMode**プロパティ	本プロパティがTRUEの時に、非同期に実行されるメソッドの誤記の修正のため、**TransactionPrint**メソッドを削除。また戻り値の説明を修正。
POSプリンタの**CapCharacterSet**プロパティ	設定される内容についての説明の明確化
POSプリンタの**JrnLineWidt**hプロパティ	**JrnLineChars**プロパティの設定により、本プロパティが更新されることの明確化。
POSプリンタの**JrnNearEnd**プロパティ	説明の脱字の追加。
POSプリンタの**RecLineWidth**プロパティ	**RecLineChars**プロパティの設定により、本プロパティが更新されることの明確化。

POSプリンタの**RotateSpecial**プロパティ	形式の型の誤記を修正のため、型の記述をBSTRからLONGに変更。
POSプリンタの**SlpLineWidth**プロパティ	**SlpLineChars**プロパティ設定により、本プロパティが更新されることの明確化。
POSプリンタの**SlpNearEnd**プロパティ	初期化および保持される値の説明の脱字の修正。
POSプリンタの**PrintImmediate**メソッド	「説明」の中の**ValidateEscape**メソッドの記述の誤記を修正のため、**ValidateData**メソッドに変更。
POSプリンタの**PrintNormal**メソッド	「説明」の中の**ValidateEscape**メソッドの記述の誤記を修正のため、**ValidateData**メソッドに変更。
POSプリンタの**PrintTwoNormal**メソッド	「形式」の中のData1の説明を、明確化。
POSプリンタの**TransactionPrint**メソッド	「説明」の中の一括処理対象となるメソッドに、**CutPaper**、**RotatePrint**、**PrintBarCode**、**PrintBitmap**メソッドを追加。
POSプリンタの**ValidateData**メソッド	「戻り値」の中の記述における誤字の修正。
POSプリンタの**ValidateData**メソッド	「戻り値」の中のOPOS_E_ILLEGALが返るケースとして「フォントタイプ指定」の記述の誤記の修正のため、該記述を削除。
スキャナの概説	デバイスの共有の説明における、プロパティがオープン後に読みとり可能となる記述を、誤記の修正のため、削除。
スキャナの**ScanData**プロパティ	設定されるデータについての明確化のため、削除または含まれるキャラクタについて詳細な説明の追加。
POSキーボードの概説	“エンコードデータの読み込み”の記述を、単に”キー”に変更。
POSキーボードの概説	デバイスの共有の説明における、プロパティがオープン後に読みとり可能となる記述を、誤記の修正のため、削除。
第11章～第14章	第11章 コインディスペンサ、第12章 MICR（磁気インクリーダ）、第13章 秤、第14章 シグニチャ・キャプチャを新規追加

OPOSDISP.H ヘッダーファイル	**CapCharacterSet**プロパティ用の定数定義値の追加、変更。
	MarqueeFormatプロパティ用の定数定義値の追加。
OPOSPTR.H ヘッダーファイル	**CapCharacterSet**プロパティ用の定数定義値の追加、変更。
	ErrorLevelプロパティ用の定数定義の追加、変更。
OPOSPTR.H ヘッダーファイル	バーコードシンボルの定数定義の次の２つを削除。 PTR_BCS_UPCD3、PTR_BCS_ISBNOCR
	RotatePrintメソッドの定数定義に**RotateSpecial**プロパティにも使用する旨のコメントを追加。
	TransactionPrintメソッド用のの定数定義値の変更。
	RotateSpecialプロパティ専用の定数定義の削除
OPOSCOIN.H、OPOSMICR.H OPOSSCAL.H、OPOSSIG.H ヘッダーファイル	コインディスペンサ、MICR（磁気インクリーダ）、秤、シグニチャ・キャプチャ ヘッダーファイルを新規追加

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1.1.1版のスーパセットに位置付けられます。

章または節	修正内容
自動釣り銭機	新デバイス：数箇所への情報追加と自動釣り銭機の特徴を記述。
トーンインジケータ	新デバイス：数箇所への情報追加と自動釣り銭機の特徴を記述。
対応する数箇所	*Timeout*パラメータを有するメソッドに、定数OPOS_FOREVER値を追加。また、OPOS_E_ILLEGALが戻り値となる事を追加。
はしがき	会社名と著作権表示の変更。
はじめにの メソッドとプロパティの使い方と デバイス共有モデルについて	コントロールが同時に複数のアプリケーションにオープンされる状態を明白にするとともに、**Claim**メソッドにより制限される機能についての記述を追加。
はじめにの **イベント**	本節の追加。
はじめにの **入力モデルについて**	エラー状態のハンドリングの明確化。 **AutoDisable**と**DataCount**の使用方法の追加。 エラー状態から抜け出す状況の明確化。 **ClearInput**の正しい実行契機の明確化。
はじめにの **出力モデルについて**	エラー状態についての状況の明確化。
共通プロパティ、メソッド、イベントの **BinaryConversion**プロパティ、 **AutoDisable**プロパティ、 **DataCount**プロパティ	左記の新プロパティの追加。 仕様書を通し、各デバイスの概説への追加。 仕様書を通し、**BinaryConversion**が影響するBstringプロパティとメソッドパラメータへの記述の追加。

共通プロパティ、メソッド、イベントの **ControlObjectVersion**プロパティ、 **ServiceObjectVersion**プロパティ	バージョン番号の不一致時に従うべき事項の追加。
共通プロパティ、メソッド、イベントの **FreezeEvents**プロパティ	イベント通知の遅延についての**FreezeEvents**規則の明確化。
共通プロパティ、メソッド、イベントの **ClearInput**メソッド **ClearOutput**メソッド	戻り値の修正：3つの値の一つを返却するように変更。
共通プロパティ、メソッド、イベントの **Open**メソッド	戻り値の修正：**ResultCode**の値がメソッドの戻り値と一致しない場合について追加。
共通プロパティ、メソッド、イベントの **StatusUpdateEvent**イベント	デバイスがイネーブルされた時の最初のイベント通知の明確化。
MICR（磁気インクリーダ）の **BankNumber**プロパティ	**TransitNumber**プロパティの4-8桁の定義を修正。
磁気ストライプリーダの **ErrorReportingType**プロパティ	左記の新プロパティの追加。
磁気ストライプリーダの **ParseDecodeData**プロパティ	不一致の明確化： **ParseDecodeData**と**ParseDecodedData**の２つが、同一プロパティとして使用されるべきことを追加。
磁気ストライプリーダの **ErrorEvent**	トラックレベルについてのエラー情報の変更。
POSキーボードの概説	POSキーボードの種別の明確化。
POSキーボードの **PosKeyData**プロパティ	本プロパティの定義の変更：論理キーコード他。
POSキーボードの **CapKeyUp**プロパティ **EventTypes**プロパティ **PosKeyEventType**プロパティ	左記の新プロパティの追加。
POSプリンタの エスケープシーケンス	OPOSで未定義のエスケープシーケンスは、プリンタへ出力されることを明確化。
POSプリンタの **CapConcurrentXxxYyy**プロパティ	FALSE値の場合を明確化。

POSプリンタの **XxxLineSpacing**プロパティ	通常文字の印刷行の高さを示しもの、すなわち、印字行の高さと行間スペースの高さの両方を加えたものと解釈されるように記述を修正。
POSプリンタの **PrintBarCode**メソッド	バーコード種別のシンボル追加。
POSプリンタの **MapMode**プロパティ、 **XxxLetterQuality**プロパティ	ハーフドットをサポートしているプリンタにおける**MapMode**の正しい取り扱いを明確化。 **XxxLetterQuality**が変更され**MapMode**がドット値の場合の影響についての明確化。
POSプリンタの **SetBitmap**メソッド	ビットマップ番号の使用において、同一のビットマップをレシートおよびスリッププリンタで使用できるように拡張。
POSプリンタの **TransactionPrint**メソッド	「戻り値」としてBUSYまたはEXTENDEDが設定されるときの明確化。
秤の概説	重量の単位がデバイスに依存し、アプリケーションで設定できないことを明記。
秤の**CapDisplay**プロパティ	左記の新プロパティの追加。
秤の**WeightUnit**プロパティ	不一致の明確化： **WeightUnit**と**WeightUnits**の２つを、本プロパティとして実装するべきことを追加。
スキャナーの **DecodeData**プロパティ、 **ScanDataLabel**プロパティ、 **ScanDataType**プロパティ	左記の新プロパティの追加。
シグニチャキャプチャの リアルタイム属性	**CapRealTimeData**、**RealTimeDataEnabled**プロパティの新プロパティの追加とリアルタイム動作についての各節の変更。
Opos.hヘッダーファイル	OPOS_FOREVER値の追加 **BinaryConversion**のための値の追加。
OposMsr.hヘッダーファイル	**ErrorReportingType**のための値の追加。
OposKbd.hヘッダーファイル	**EventTypes**のための値の追加。
OposPtr.hヘッダーファイル	PTR_RP_NORMAL_ASYNCシンボルの削除。 スキャナに対応するシンボルの追加。
OposScan.hヘッダーファイル	**ScanDataType**プロパティのためのシンボルの追加。

付録C 技術的詳細説明	イベントハンドラ節の削除。再検討し、第一章の”イベント”にマージ。
全体として	各種の校正ミスの修正

第1.3版

第1.3版は、デバイスクラスの追加といくつかのAPIの追加と誤記修正を行いました。

第1.3版は、第1.2版のスーパセットに位置付けられます。

章または節	修正内容
最初の2ページ	著作権表示の更新および参照Webサイト更新。
全般	イベント通知の使用用語の変更。”enqueue”(キューイング）と”deliver”(通知）を使用し、”firing”(発行)としていた記述を適切な表現に修正。（ただし、英文のみで使い分け。日本版は従来と同じ）
フィスカルプリンタ	新デバイス:数箇所への情報追加とフィスカルプリンタの章の新設とヘッダーファイルの追加。
PINパッド	新デバイス:数箇所への情報追加とPINパッドの章の新設とヘッダーファイルの追加。
リモートオーダディスプレイ	新デバイス:数箇所への情報追加とリモートオーダディスプレイの章の新設とヘッダーファイルの追加。
バンプバー	新デバイス:数箇所への情報追加とバンプバーの章の新設とヘッダーファイルの追加。
対応する数箇所	いくつかの入力デバイスにおける**ErrorEvent**イベントでのリトライ応答許可の緩和。
はじめにの **イベント**	先頭イベントがブロックされる効果の明確化。
はじめにの **入力モデルについて**	**ErrorEvent**のキューイングと通知に関する詳細な記述を追加。 非同期入力についての説明の追加。
はじめにの **デバイス電源通知モデルについて**	本節の追加
はじめにの**OPOSコントールの解説**	CURRENCYデータ型を追加。
共通プロパティ、メソッド、イベントの CapPowerReportingプロパティ、 PowerNotifyプロパティ、 PowerStateプロパティ	左記のプロパティの本章への追加と、各デバイスの一覧節への追加。
共通プロパティ、メソッド、イベントの ResultCodeプロパティ	**OPOS_E_BUSY**の意味の記述を一般化。
共通プロパティ、メソッド、イベントの StatusUpdateEventイベント	電源状態通知機能の追加とパラメータ名を*Data*から*Status*に変更。

各々デバイスの章	一覧節に電源状態通知関連のプロパティを追加。 **StatusUpdateEvent**イベントのサポート追加（前版で未通知時） **StatusUpdateEvent**イベントの説明に電源状態通知の参照記述を追加（前版で既に通知時）
ドロワーの**DirectIOEvent**イベント	必要条件に**Enable**を追加。
ドロワーの**StatusUpdate**イベント	パラメータ名を *Data* から *Status* に変更。 定数CASH_SUE_DRAWERCLOSED値、およびCASH_SUE_DRAWEROPEN値を追加し、**StatusUpdateEvent**イベントが返す電源状態値を受け取る記述を追加。
ラインディスプレイの**DataCount**プロパティ	プロパティ一覧での初期化の項目を適用外に修正。
ラインディスプレイの**DirectIOEvent**イベント	必要条件に**Claim**を追加。
ラインディスプレイの**StatusUpdateEvent**イベント	必要条件に**Open, Claim & Enable**を追加
ハードトータルのイベント一覧	誤記修正のため**DataUpdateEvent**を**DataEvent**に変更。
磁気ストライプリーダの**DecodeData**プロパティ	**FALSE**の時の説明追加と**RawFormat**時のコード表追加
磁気ストライプリーダの**ExpirationDate**プロパティ	格納形式と格納例の説明追加
磁気ストライプリーダの**Track1Data**プロパティ	制御コードを含まない事、**DecodeData**プロパティとの関係の説明追加
磁気ストライプリーダの**Track2Data**プロパティ	制御コードを含まない事、**DecodeData**プロパティとの関係の説明追加
磁気ストライプリーダの**Track3Data**プロパティ	制御コードを含まない事、**DecodeData**プロパティとの関係の説明追加
磁気ストライプリーダの**DataEvent**イベント	誤記修正のため、形式“**void**”の綴りを小文字に修正
磁気ストライプリーダの**ErrorEvent**イベント	データが読めた場合にサイズを返さない事、**DataEvent**と異なり**DataCount**プロパティは増加しない事、**AutoDisable**プロパティと関係しない事、の説明追加

POSプリンタの一覧の **DirectIOEvent**イベント	必要条件の変更。 「Open」→「Open, Claim」
POSプリンタの概説	追加した機能の版数の明記。
POSプリンタの **CapConcurrentJrnRec**プロパティ	説明に新定義値の追加。 （PTR_TWO_RECEIPT_JOURNAL）
POSプリンタの **CapConcurrentJrnSlp**プロパティ	説明に新定義値の追加。 （PTR_TWO_SLP_JOURNAL）
POSプリンタの **CapConcurrentRecSlp**プロパティ	説明に新定義値の追加。 （PTR_TWO_SLP_RECEIPT）
POSプリンタの **CapCharacterSet**プロパティ、 **CapTransaction**プロパティ、 **ErrorLevel**プロパティ、 **ErrorString**プロパティ、 **FontTypefaceList**プロパティ、 **RotateSpecial**プロパティ、 **RecBarCodeRotationList**プロパティ、 **SlpBarCodeRotationList**プロパティ	追加の版数明記。
POSプリンタの **ErrorStation**プロパティ	説明に新定義値の追加。 （PTR_TWO_RECEIPT_JOURNAL、 PTR_TWO_SLP_JOURNAL、 PTR_TWO_SLP_RECEIPT）
POSプリンタの **JrnLineChars**プロパティ	説明に**JrnLineCharsList**にない値が設定された時の説明を追加。
POSプリンタの **RecLineChars**プロパティ	説明に**RecLineCharsList**にない値が設定された時の説明を追加。
POSプリンタの **SlpLineChars**プロパティ	説明に**SlpLineCharsList**にない値が設定された時の説明を追加。
POSプリンタの **PrintTwoNormal**メソッド	形式、説明に第1.2版までの定数値と第1.3版以降の定数値の説明を追加。
POSプリンタの **TransactionPrint**メソッド、 **ValidateData**メソッド	追加の版数明記。
POSプリンタの **StatusUpdateEvent**イベント	*Status*の値に電源報告の値を追加。
POSキーボードの **CapKeyUp**プロパティ	形式の型の誤記を修正のため、型の記述をLONGからBOOLに変更。

POSキーボードの **EventTypes**プロパティ	注釈にて版数による仕様の違いを追加。
POSキーボードの **POSKeyData**プロパティ	注釈にて版数による仕様の違いを追加。
MICRの **DirectIOEvent**イベント	必要条件に**Claim**を追加。
秤のプロパティ、メソッド、イベント一覧	非同期入力追加による以下のプロパティを適応外から適応に変更 AutoDisable DataCount DataEventEnable AsyncMode ClearInput DataEvent ErrorEvent 文字列表示機能追加 CapDisplayText MaxDisplayTextChars DisplayText 売上価格計算機能追加 CapPriceCalculating SalesPrice UnitPrice ゼロアジャスト／風袋追加機能t追加 CapTareWeight CapZeroScale TareWeight ZeroScale
秤の機能	文字列表示装置、売上価格計算機能、風袋、ゼロアジャスト機能の説明追加。
秤のモデル	モデルの説明追加。
秤の**AsyncMode**プロパティ	左記の新プロパティ追加
秤の**CapDisplayText**プロパティ	左記の新プロパティ追加

秤の**CapPriceCalculating**プロパティ	左記の新プロパティ追加
秤の**CapTareWeight**プロパティ	左記の新プロパティ追加
秤の**CapZeroScale**プロパティ	左記の新プロパティ追加
秤の**MaxDisplayTextChars**プロパティ	左記の新プロパティ追加
秤の**SalesPrice**プロパティ	左記の新プロパティ追加
秤の**TareWeight**プロパティ	左記の新プロパティ追加
秤の**UnitPrice**プロパティ	左記の新プロパティ追加
秤の**DisplayText**メソッド	左記の新メソッド追加。
秤の**ReadWeight**メソッド	非同期時の説明追加。
秤の**ZeroScale**メソッド	左記の新メソッド追加。
秤の**DataEvent**イベント	左記の新イベント追加。
秤の**ErrorEvent**イベント	左記の新イベント追加。
シグニチャキャプチャの一覧の**DirectIOEvent**イベント	必要条件の変更。 「Open」→「Open, Claim」
シグニチャキャプチャの一覧の**OutputCompleteEvent**イベント	必要条件の訂正。 「*Not Supported*」→「*適用外*」
シグニチャキャプチャの一覧の**StatusUpdateEvent**イベント	必要条件の変更。 「*Not Supported*」→「Open, Claim, & Enable」
トーンインジケータ	排他デバイスとして説明している箇所と共有デバイスとして説明している箇所が混在していたため、排他アクセス可能な共有デバイスとした説明に統一。
Opos.hヘッダーファイル	**CapPwoerReporting**、**PwoerNotify**、**PowerState**の各プロパティおよび **StatusUpdateEvent**イベントの電源状態通知の定義を追加。
OposPTR.Hヘッダーファイル	**PrintTwoNormal**に対応する定義の追加。 （PTR_TWO_RECEIPT_JOURNAL、 PTR_TWO_SLP_JOURNAL、 PTR_TWO_SLP_RECEIPT）
全体を通して	校正ミスの修正

第1.4版

第1.4版は、デバイスクラスの追加を行いました。

章または節	修正内容
CAT	新デバイス:数箇所への情報追加とCATの章の新設とCATヘッダーファイルの追加。

第1.5版

第1.5版は、第1.4版のスーパセットに位置付けられます。

第1.5版は、2個のデバイスクラスを追加しています。

章または節	修正内容
最初の2ページ	著作権表示の更新および参照Webサイト更新。
全般	**Claim**を**ClaimDevice**に、**Release**を**ReleaseDevice**に置換。
はじめに	OLEを適宜ActiveXに修正。
共通プロパティ、メソッド、イベントの**OpenResult**プロパティ	新規プロパティを追加し、各デバイスの一覧セクションに追加。
自動釣り銭機	入金機能の追加。
ドロワー	マルチドロワー制御の追加
CAT	**PaymentMedia**プロパティの追加。
フィスカルプリンタ	**CountryCode**、**ErrorOutID**、**PrinterState**、**QuantityDecimalPlaces**そして**QuantityLength**プロパティを、オープン、クレイムそしてイネーブルの替わりに、オープン後の初期化を反映するように更新。 **DuplicateReceipt**：R/Wを示すように修正。戻り値を追加。
ラインディスプレイ	UNICODE文字セットサポートをデバイスに追加するため、DISP_CCS_UNICODE を**CapCharacterSet**に、DISP_CS_UNICODEを**CharacterSet**に追加。
磁気ストライプリーダ	**Track4Data**、**CapTransmitSentinels**と**TransmitSentinels**プロパティを追加。JIS-II トラックデータを明確にサポート。 **DataEvent** *Status*：ハイバイトに意味追加 **ResultCode**=OPOS_E_EXTENDEDの時の**ErrorEvent**の**ResultCodeExtended**：ハイバイトに意味追加
PINパッド	**Track4Data**プロパティを追加。 非同期入力についての説明の追加。
ポイントカード機	新規デバイス：ポイントカード機の章とヘッダファイルを追加し、必要な個所に適宜情報を追加。

パワーマネージメント	新規デバイス：パワーマネージメントの章とヘッダファイルを追加し、必要な個所に適宜情報を追加。
POSプリンタ	（インクジェット技術の）カラー印刷、スリップへの両面印字、及び検出位置マーク付用紙のサポートを追加。 UNICODE文字セットサポートをデバイスに追加するため、PTR_CCS_UNICODE を **CapCharacterSet**に、PTR_CS_UNICODEを **CharacterSet**に追加。
リモートオーダーディスプレイ	UNICODE文字セットサポートをデバイスに追加するため、ROD_CCS_UNICODE を **CapCharacterSet**に、ROD_CS_UNICODEを **CharacterSet**に追加。
秤	**SalesPrice**、**TareWeight**と**UnitPrice**プロパティを、オープン、クレイムそしてイネーブルの替わりに、オープン後の初期化を反映するように更新。 **ErrorEvent**：レスポンスの値にOPOS_ER_RETRYを追加
シグニチャ・キャプチャ	モデルの更新して**AutoDisable**の実装の議論を追加。 **RealTimeDataEnabled**：いつ有効に働くかを明確化。 **DataEvent**：リアルタイムデータを含むイベントが発生する条件を修正

トーンインジケータ	**AsyncMode**、**Tone1Pitch**、**Tone1Volume**、**Tone1Duration**、**Tone2Pitch**、**Tone2Volume**、**Tone2Duration**と**InterToneWait**プロパティを、オープン、クレイムそしてイネーブルの替わりに、オープン後の初期化を反映するように更新。
	他のアプリケーションがデバイスをクレームして**Sound**メソッドを呼び出したときの**Sound**メソッドの動作を明確化。
Opos.hヘッダファイル	**OpenResult**プロパティ値を追加。
付録B	ポイントカード機とパワーマネージメントのヘッダファイル追加。本仕様書で修正されたデバイスのヘッダファイルを更新。
付録C	ATL利用を含めるため、システムストリングの情報を更新。
付録D	**ClaimDevice**と**ReleaseDevice**の詳細を付録に追加。

第1.6版

第1.6版は、第1.5版のスパーセットに位置づけられます。

章または節	修正内容

FiscalPrinter	プロパティを追加 **CapAdditionalHeader,CapAdditionalTrailer,** **CapChangeDue,CapEmptyRecieptIsVoidable,** **CapFiscalRecieptStation,CapFiscalRecieptType,** **CaoMultiContractor,CapOnlyVoidLastItem,** **CapPackegeAdjustment,CapPostPreLine,** **CapSetCurrency,CapTotalizerType,** **ActualCurrency,AdditionHeader,AdditionalTrailer,ChangeDue,ContractorID,DateType,** **FiscalRecieptStation,FiscalRecieptType,** **MessageType,PostLine,PreLine,TotalizerType** **Open**メソッドでの初期化の変更 **CountryCode,ErrorOutID,PrinterState,** **QuantityDecimalPlaces,QuantityLength** メソッドの追加 **SetCurrency,PrintRecCash,PrintRecItemFuel,** **PrintRecItemFuelVoid,** **PrintRecPackageAdjustment,** **PrintRecPackageAdjustmentVoid,** **PrintRecRefundVoid,PrintRecSubtotalAdjustVoid,** **PrintRecTaxID** ベルギーとルーマニアの国コードをサポート 概要の説明を変更 **CapPositiveAdjustment**の説明を変更 プロパティ(機能追加の対応) **CountryCode,DayOpened,DescriptionLength** メソッド(機能追加の対応) **EndFiscalReciept,GetData,GetDate,PrintRecItem,** **PrintRecMessage,PrintRecNotPaid,** **PrintRecRefound,PrintRecSubtotal,** PrintrecSubtotalAdjustment,PrintRecTotal,

	PrintRecVoid,PrintRecVoidItem,PrintZReport, SetHeaderLine イベント(機能追加の対応) ErrorEvent
LineDisplay	プロパティの追加 **CapBlinkRate,CapCursorType,CapCustomGlyph,** **CapReadBack,CapReverse,BlinkRate,CursorType,** **CustomGlyphList,GlyphHeight,GlyphWidth** メソッドの追加 **DefineGlyph,ReadCharacterAtCursor** 追加した定義(リバース表示対応のための変更) DISP_DT_REVERSE, DISP_DT_BLINK_REVERSE リバース表示対応のため変更したメソッド **DisplayText,DisplayTextAt**
秤	**Open**メソッドでの初期化タイミングの変更 **SalesPrice,TareWeight,UnitPrice**
ToneIndicator	**Open**メソッドでの初期化タイミングの変更 **AsyncMode,InterToneWait,Tone1Duration,** **Tone1Pitch,Tone1Volume,Tone2Duration,** **Tone2Pitch,Tone2Volume**

第1.7版

第1.7版は、第1.6版のスパーセットに位置づけられます。

章または節	修正内容
ＰＯＳプリンタ	一覧に下記プロパティの追加 **CapMapCharacterSet、MapCharacterSet、** **RecBitmapRotationList、SlpBitmapRotationList** **“機能”** 一般的なグラフィカル印刷で、**MapMode**プロパティにPTR_MM_DOTSを指定した場合の記述を追加。 **“モデル”** **AsyncMode**プロパティがTRUEの時のメソッドの動作の記述を追加。 エラー通知モデルの記述を追加。 **“印字データとエスケープシーケンス”** ”ESC|#E”の機能について記述を追加。 **新プロパティを追加。** **CapMapCharacterSet 、MapCharacterSet 、MapCharacterSet、RecBarCodeRotationList、RecBitmapRotationList、SlpBarCodeRotationList、SlpBitmapRotationList、SlpPrintSide、** **PrintBarCode** メソッド OPOS_E_ILLEGALの意味を記述を追加。 参照欄に**MapMode**プロパティ、**RotateSpecial**プロパティを追加。

PrintBitmap メソッド

説明の欄に文字データとビットマップの印刷位置の関係の記述を追加。

参照欄に**MapMode**プロパティを追加。

RotatePrint メソッド

PTR_RP_BARCODE, PTR_RP_BITMAPを追加。

SetBitmap メソッド

ビットマップ番号の有効な値を“１から２０”までに変更。

説明の欄に文字データとビットマップの印刷位置の関係の記述を追加。

TransactionPrint メソッド

参照欄に**CapTransaction** プロパティ、**CutPaper** メソッド、**PrintBarCode** メソッド、**PrintBitmap** メソッド、**PrintNormal** メソッド、**RotatePrint**メソッドを追加。

ErrorEvent イベント

OPOS_ER_CLEARの記述を追加。

LineDisplay	**DeviceColumns**, **DeviceRows** プロパティが**ScreenMode**の変更で更新
	コードページマッピング機能
	CapMapCharacterSet, **MapCharacterSet** プロパティの追加
	スクリーンモード機能
	CapScreenMode, ScreenMode, ScreenModeList プロパティの追加
	ビットマップ表示機能
	CapBitmap, MaximumX, MaximumY プロパティ、および
	DisplayBitmap, SetBitmap メソッドの追加
	ClearText, ScrollText メソッドで表示されているビットマップの処理を追加
	DisplayText メソッドにエスケープシーケンスのサポートによる機能説明を追加
PointCardReaderWriter	一覧に下記プロパティの追加
	CapMapCharacterSet、MapCharacterSet、
	“印字データとエスケープシーケンス”
	”ESC|#E”の機能について記述を追加。
共通項目	以下の機能を日本語仕様から削除
	BumpBar、RemoteOrderDisplay、FiscalPrinter、MICR、SignatureCapture
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
付録A	デバイスクラスキーにCheckScannerとMotionSenserを追加
付録B	ヘッダーファイルにCheckScannerとMotionSenserを追加

第1.8版

第1.8版は、第1.7版のスパーセットに位置づけられます。

第1.8版は、１個のデバイスクラスを追加しています。

章または節	修正内容
表紙ほか	「OLE for Retail POS」を「OpenPOS for OLE」に変更 「OPOS」を「OLE-POS」の略称から「OpenPOS」の略称に変更
はじめに	「デバイスインフォメーションレポーティングモデルについて」の説明を追加
共通プロパティ、メソッド、イベントおよび全デバイスの「一覧」	新たに追加したプロパティ **CapStatisticsReporting** プロパティ **CapUpdateStatistics** プロパティ 新たに追加したメソッド **ResetStatistics** メソッド **RetrieveStatistics** メソッド **UpdateStatistics** メソッド
LineDisplay	更新したプロパティ **CapCursorType** プロパティ **CursorType** プロパティ

ＰＯＳプリンタ	**“機能”** ピクセルレベル印刷の説明を変更。 “モデル” **StatusUpdateEvent**を通知する条件の説明を追加 動機印刷の機能を明確にした説明を追加 **CapJrnPresent**プロパティ 「**CapJrnPresent**プロパティがFALSEならば、このプロパティはゼロに初期化されます。」を解説に追加 **CapRecPresent**プロパティ **CapRecPresent**プロパティがFALSEならば、このプロパティはゼロに初期化されます。 **CapSlpPresent**プロパティ **CapSlpPresent**プロパティがFALSEならば、このプロパティはゼロに初期化されます。 **PrintBarCode** メソッド 印字可能なバーコードの種類を追加 PTR_BCS_Code128_Parsed Code 128 Parsed PTR_BCS_RSS14　　RSS (Reduced Space Symbology) PTR_BCS_RSS_EXPANDED　RSS Expanded **Code128のための特別な考慮**を追加 **RotatePrint** メソッド “説明” 「*Rotation*にPTR_RP_BARCODEが含まれている場合には、**RotateSpecial**の内容は無視されます。」を追加 **StatusUpdateEvent**イベント PTR_SUE_SLP_EMPTYの説明を変更 カバー状態の説明を追加

Scanner	**ScanDataType**プロパティ
	読み取り可能なバーコードの種類を追加
	SCAN_SDT_RSS14
	SCAN_SDT_RSS_EXPANDED
	SCAN_SDT_CCA
	SCAN_SDT_CCB
	SCAN_SDT_CCC
SmartCardReaderWriter	新規デバイス：ポイントカード機の章とヘッダファイルを追加し、必要な個所に適宜情報を追加。
付録A	デバイスクラスキーにSmartCardRWを追加
付録B	ヘッダーファイルにCheckScannerとMotionSenserを追加

第1.9版

第1.9版は、全てのデバイス共通項目に、ファームウェア更新の対応を追加しています。また、第1.8版の各項目に機能の追加、仕様の明記、修正などのさまざまな修正が含まれます。

章または節	修正内容
表紙ほか	発行日とＶｅｒｓｉｏｎの変更
	目次の追加（アップデートファームウェアデバイスモデル）
はじめに	「アップデートファームウェアデバイスモデルについて」の説明を追加
共通プロパティ、メソッド、イベントおよび全デバイスの「一覧」	新たに追加したプロパティ
	CapCompareFirmwareVersion プロパティ
	CapUpdateFirmware プロパティ
	新たに追加したメソッド
	CompareFirmwareVersion メソッド
	UpdateFarmware メソッド
	StatusUpdateEventに6項目のステータスを追加
CAT	電子決済機能の追加
	新たに追加したプロパティ
	CapCashDeposit プロパティ
	CapLockTerminal プロパティ
	CapLogStatus プロパティ
	CapUnlockTerminal プロパティ
	LogStatus プロパティ
	SettledAmount プロパティ
	新たに追加したメソッド
	CashDeposit メソッド
	LockTerminal メソッド
	UnlockTerminal　メソッド

CheckScanner	コントラストの機能を追加
	新たに追加したプロパティ
	CapAutoContrast プロパティ
	CapContrast プロパティ
	Contrast プロパティ
Keylock	**ResetStatistics**、**RetreavwStatistics**、**UpdateStatistics**メソッドの条件を「**Open**、**Claim**」に変更
PINPad	**CapLanguage**プロパティの引数PPADDISP_RESTRICTED_ORDERをPPAD_LANG_UNRESTRICTEDに変更
	MerchantIDプロパティの説明で**EnamledPINEntry**を**BeginEFTTransaction**に変更
	VerifyMACメソッドの戻り値にOPOS_E_FAILUREを追加
POSPower	電池駆動の機能の追加
	新たに追加したプロパティ
	BatteryCapacityRemainingプロパティ
	BatteryCriticallyLowThresholdプロパティ
	BatteryLowThresholdプロパティ
	CapBatteryCapacityRemainingプロパティ
	CapRestartPOSプロパティ
	CapStandbyPOSプロパティ
	CapSuspendPOSプロパティ
	CapValiableBatteryCriticallyLowThresholdプロパティ
	CapVariableBatteryLowThresholdプロパティ
	PowerSourceプロパティ
	新たに追加したメソッド
	RestartPOSメソッド
	StandbyPOSメソッド
	SuspendPOSメソッド

ＰＯＳプリンタ	ページモード印刷機能の追加 新たに追加したプロパティ **CapConcurrentPageMode**プロパティ **CapRecPageMode**プロパティ **CapSlpPageMode**プロパティ **PageModeArea**プロパティ **PageModeDescriptior**プロパティ **PageModeHorizontalPosotion**プロパティ **PageModePrintArea**プロパティ **PageModePrintDirection**プロパティ **PageModeStation**プロパティ **PageModeVerticalPosition**プロパティ 新たに追加したメソッド **ClearPrintArea**メソッド **PageModePrint**メソッド 初期化値の変更 **JrnCurrentCartridge**プロパティ **RecCurrentCartridge**プロパティ **SlpCurrentCartridge**プロパティ **ChangePrintSide**メソッドのOPOS_E_EXTEDEDの値で2項目を削除 **ErrorEvent**イベントの説明で**DataEventEnabled**に関する誤った記述を修正

Scale	LiveWeightの機能の追加 新たに追加したプロパティ **CapStatusUpdate**プロパティ **ScaleLiveWeight**プロパティ **StatusNotify**プロパティ 機能を追加したメソッド **ReadWeight**メソッド 値を追加したイベント **StatusUpdateEvent**イベント
SmartCardReaderWriter	新規デバイス：ポイントカード機の章とヘッダファイルを追加し、必要な個所に適宜情報を追加。

第1.10版

第1.10版は、「はじめに」と「共通項目」にPOS for .NETに関する内容が追加されています。新デバイスとしてバイオメトリクス、電子ジャーナルが追加されています。全デバイスに入力プロパティをクリアする機能が追加されています。また、第1.9版の各項目に機能の追加、仕様の明記、修正などのさまざまな修正が含まれます。

章または節	修正内容
表紙ほか	発行日とＶｅｒｓｉｏｎの変更 目次の追加（電子ジャーナル、バイオメトリクス、POS for .NET）
はじめに	「入力モデルについて」の**ClearInputProperties**メソッドの説明を追加
共通プロパティ、メソッド、イベントおよび全デバイスの「一覧」	**ClearInputProperties**メソッドの追加 **ResetStatictics** メソッドの戻り値にOPOS_ESTATS_DEPENDENCYを追加 **UpdateStatictics** メソッドの戻り値にOPOS_ESTATS_DEPENDENCYを追加
ラインディスプレイ	**CharacterSet** プロパティに実装に依存する情報を追加
磁気ストライプリーダ	書き込み機能を追加 新たに追加したプロパティ CapWritableTracks　プロパティ TracksToWrite　プロパティ EncodingMaxLength　プロパティ 新たに追加したメソッド WriteTracks メソッド
ＰＯＳプリンタ	「機能」の同期印刷の説明を変更 動作するエスケープシーケンスに「ESC|#R」を追加 エスケープシーケンスにバーコード印刷の説明を追加 横倍率と縦倍率の説明が逆転していたものを修正 新たに追加したメソッド **PrintMemoryBitmap**メソッド 変更のあったプロパティ **CaracterSet**　プロパティ 変更のあったプロパティ **RotatePrint**　メソッド **SetLogo**　メソッド

スキャナ	**ScanDataLavel**プロパティ
	説明分の差し替え。
秤	**ScaleLiveWeight**プロパティ
	StatusNotifyプロパティ
	StatusUpdateEventイベント
	説明の*Reason*で定数定義名称を変更
	SCL_XXXX　→　SCAL_XXXX
	ReadWriteメソッド、**ZeroScale**メソッド
	戻り値に値と説明を追加
	OPOS_E_BUSY　非同期のReadWeightが実行中です。
CAT	**TransactionType**プロパティ
	説明に「CAT_TRANSACTION_CHECKCARD　カードチェック」を追加
パワーマネージメント	**SuspendPOS**の記述修正
	説明にパラーメータの*Reason*を追加
	StanbyPOSメソッド、**SuspendPOS**メソッド
	説明の*Reason*で定数定義名称を変更
	PWR_Request　→　PWR_Reason_Request
	PWR_Allow　→　PWR_Reason_Allow
	PWR_Deny　→　PWR_Reason_Deny
バイオメトリクス	新規デバイス：バイオメトリクス機の章とヘッダファイルを追加し、必要な個所に適宜情報を追加。
電子ジャーナル	新規デバイス：電子ジャーナル機の章とヘッダファイルを追加し、必要な個所に適宜情報を追加。
付録C	「キャラクタセットのマッピング」の説明を変更。

第1.11版

第1.1１版は、「共通項目」にPOS for .NET開発リファレンス完全版が追加されています。新デバイスとして紙幣入金機、紙幣出金機、硬貨入金機、イメージスキャナの４つのデバイスカテゴリーの章が追加されています。

以下が変更の詳細一覧です。各詳細は、変更された項・章が記載されています。スペルミス等の軽微な変更は記載されていませんので御了承ください。

章または節	修正内容
表紙ほか	発行日とVersionの変更
	目次の追加（紙幣釣銭挿入機、紙幣排出機、硬貨投入機、イメージスキャナ）
はじめに	「入力モデルについて」の**ClearInputProperties**メソッドの説明を追加
	「データタイプ」に「バイナリ配列」、「int32配列」、「int32配列ポインタ参照」のデータタイプ定義を追加。バイオメトリクスおよび磁気ストライプリーダデバイスカテゴリーで使用されるBIR構造体およびその他のパラメータをサポートするために「バイナリポインタ参照」の定義を見直し。
	「エラーコード」にOPOS_E_DEPRECATEDを追加
	「サポート削除への対応」にサポート削除に関する説明を追加 ハイドラ接続機器の考慮に関する記載を追加
共通プロパティ、メソッド、イベントおよび全デバイスの「一覧」	**CapPowerReport**プロパティ**およびPowerState**プロパティ**のエラーの説明に、エラーで例外が発生する記述を追加**
	ResultCodeの値一覧にOPOS_E_DEPRECATEDを追加
キーロック	電子錠のサポートを追加。
磁気ストライプリーダ	ExpirationDateプロパティの形式省略を修正
	WriteTracksメソッドの引数をプログラミングしやすいように文字列型からバイナリ配列型に変更
ＰＯＳプリンタ	PageModePrintメソッドがサポートするメソッド一覧にPrintMemoryBitmapメソッドを追加
	RotatePrintメソッドの引数rotationの説明を明記

フィスカルプリンタ	PrintRecNotPaidメソッドでサポートから削除されたCapAmountNotPaidプロパティのかわりにCapReceiptNotPaidを参照するように更新
	新しく定義（FPTR_RT_EOD_ORDINAL）を追加し、PrintReceiptメソッドの既存の定義（FPTR_RT_ORDINAL）を明記
	ActualCurrencyプロパティ（新貨幣）、CapCheckTotalプロパティ（CheckTotalに対する制限事項）、CapPositiveSubtotalAdjustmentプロパティ（新機能）、DateTypeプロパティ（新規の値）、FiscalReceiptTypeプロパティ（新規レシートタイプ）、BeginFiscalDocumentメソッド（制限解除およびエラーコード追加）、BeginFiscalReceiptメソッド（エラーコード追加）、GetVatEntryメソッド（Capabilityの参照修正）、PrintRecItemAdjustmentメソッド(クーポン追加)、PrintRecItemAdjustmentVoidメソッド（クーポン追加）、PrintRecMessageメソッド（制限緩和）、PrintRecSubtotalAdjustmentメソッド（割増許可およびクーポン追加）、SetVatTableメソッド（機能チェック追加）、SetVatValueメソッド（機能チェック追加）、ErrorEventイベント（ErrorCodeExtendedに新しい値を追加）
	PrintRecItemVoidメソッドおよびPrintRecItemAdjustmentVoidメソッドを追加
	CapAmountNotPaidプロパティおよびPrintRecVoidItemメソッドをサポートから削除
自動釣銭機	新しい現金管理装置のサポートを追加
スキャナ	ScanDataTypeプロパティに2Dバーコード体系を追加
チェックスキャナ	ImageTagDataプロパティの使用方法および詳細と関連プロパティおよびメソッドを明記
コインディスペンサ	新しい現金管理装置のサポートを追加

バイオメトリクス	ProcessPrematchDataメソッドとCapPrematchDataプロパティにクロスリファレンスを追加
	CapSensorTypeおよびSensorTypeプロパティの定数*_KEYSTROKE_DYNAMICSのスペルを修正
	SensorColor、SensorOrientation、SensorTypeプロパティのOPOS_E_ILLEGALの説明を追加および修正
	メソッドのエラーコードでOPOS_E_FAILUREを変更（旧バージョンではパラメータエラー時にOPOS_E_ILLEGALを設定）
	EndCaptureメソッド以外のすべてのメソッドのErrorCodeに、キャプチャ処理中にメソッドが呼ばれた場合に設定されるOPOS_E_ILLEGALを追加
	IdentifyおよびIdentifyMatchメソッドの引数ReferenceBIRPopulationおよびCandidateRankingを、それぞれ「バイナリ」型から「バイナリ配列」型、「バイナリ」型から「int32配列」型に修正
	ErrorEventに削除された備考欄を追加
	StatusUpdateEventに新しくBIO_SUE_MOVE_SLOWER、BIO_SUE_MOVE_FASTER、BIO_SUE_SENSOR_DIRTYのステータスを3つ追加
電子ジャーナル	CapMediumIsAvailable、CapPrintContent、CapPrintContentFileプロパティに対応する各プロパティおよびメソッドのクロスリファレンスを追加。ＰＯＳプリンタへのすべての参照にフィスカルプリンタへの参照も追加。モデルの項およびQueryContentメソッドの説明をわかりやすく修正。
	PrintContentおよびQueryContentメソッドの引数ToMarkerの説明をわかりやすく修正
紙幣入金機	新規デバイス：紙幣入金機の章とヘッダファイルを追加し、必要な個所に適宜情報を追加。
硬貨入金機	新規デバイス：硬貨入金機の章とヘッダファイルを追加し、必要な個所に適宜情報を追加。
紙幣出金機	新規デバイス：紙幣出金機の章とヘッダファイルを追加し、必要な個所に適宜情報を追加。
イメージスキャナ	新規デバイス：イメージスキャナ機の章とヘッダファイルを追加し、必要な個所に適宜情報を追加。

第1.12版

第1.12版は、新デバイスとして電子バリューR/W、RFIDスキャナの２つのデバイスカテゴリーの章が追加されています。

以下が変更の詳細一覧です。各詳細は、変更された項・章が記載されています。スペルミス等の軽微な変更は記載されていませんので御了承ください。

章または節	修正内容
表紙ほか	発行日とVersionの変更
	目次の追加（電子バリューR／W、ＲＦＩＤスキャナ）
はじめに	ハイドロデバイスの説明を変更
スキャナ	RSSのコンスタントをGS1の定義に変更
磁気ストライプリーダ	AAMVAカードサポートによる機能を追加
	次のプロパティを追加
	CapCardAuthentication, CapDataEncription, CapDeviceAuthentication, CapTrackDataMasking, AdditionalSecurityInformation, CardAuthenticationData, CardAuthenticationDataLength, CardPropatyList, CardType, CardTypeList, DataEncryptionAlgorithm, DeviceAuthenticated, DeviceAuthenticationProtocal, Track1EncryptedData, Track1EncryptedDataLength, Track2EncryptedData, Track2EncryptedDataLength, Track3EncryptedData, Track3EncryptedDataLength, Track4EncryptedData, Track4EncryptedDataLength, WriteCardType
	次のメソッドを追加
	AuthenticateDevice, DeauthenticateDevice, RetrieveCardProperty, RetrieveDeviceAuthenticationData, UpdateKey
ＰＯＳプリンタ	ESCシーケンス表でStrike-Throughをサポート
	ESCシーケンス表にアンダーラインを追加
	PrintBarcodeメソッドのRSSのコンスタントをGS1の定義に変更
バイオメトリクス	メソッドの引数と記述の修正と各プロパティの連携した説明を修正。
	プロパティの修正
	CapRawSensorData,CapRealTimedata,RawSensorData,RealTimeDataEnabled,
	メソッドの修正
	Identify, IdentifyMatch, Verify, VerifyMatch
	StatusUpdateEventに値を追加
電子ジャーナル	StatusUpdateEventに新しく値を追加
電子バリューリーダライタ	新規デバイス：電子バリューリーダライタの章とヘッダファイルを追加し、必要な個所に適宜情報を追加。

RFIDスキャナ	新規デバイス：RFIDスキャナの章とヘッダファイルを追加し、必要な個所に適宜情報を追加。
XMLPOS	付録Ｆ：ＸＭＬＰＯＳ仕様を追加。

第1.13版

第1.13版は、Version1.12を使用されているデバイスベンダーとアプリケーションベンダーからのフィードバックによる更新が含まれています。

以下が変更の詳細一覧です。各詳細は、変更された項・章が記載されています。スペルミス等の軽微な変更は記載されていませんので御了承ください。

章または節	修正内容
表紙ほか	発行日とVersionの変更
はじめに	アプリケーションがDisabledだと判断しているデバイスからデータを受信した場合のドライバの動作を入力モデルに明記しました。
	OutputCompleteEventのイベントIDがシーケンシャルなデータではない事を明記しました。
共通プロパティ、メソッド、イベントおよび全デバイスの「一覧」	**ErrorEvent** イベントでER_RETRY,ER_CLEAR,ER_CONTINUEINPUTの値を明確にしました。
磁気ストライプリーダ	MSRに暗号と認証の機能を追加しました。
	AcountNumberプロパティをマスクする方法を追加しています。
	MSRマスクする機能を持つ事を示す**CapTrackDataMasking**プロパティを追加しました。
	CardAuthenticationDataLengh プロパティの修正とDecordeDataプロパティを修正しました。
	ErrorReportingプロパティに例題を追加しました。
	以下のプロパティを暗号化される前のデータ長を示す為に修正しました。 Track1EncriptionDataLength Track2EncriptionDataLength Track3EncriptionDataLength Track4EncriptionDataLength
	ReciveCardPropertyメソッドの引数inputとoutの型と値を訂正しました。
ＰＯＳプリンタ	CapRecResultLineプロパティ、CapSlpRuleedLineプロパティ、DrawRuledLineメソッドを追加しました。
	データ文字とESCシーケンスに可変長データを含むデータの扱い方を記述しました。
	エスケープシーケンスの表に罫線についての表記を追加しました。
	POSプリンタで表現できる罫線の描画機能を追加しました。
	PrintBarCodeメソッドに二次元コードを追加しましました。

秤	ZeroValidプロパティを追加しました。 秤デバイスに有効な重さとして、重さ“0”を通知する機能の表記を追加しました。 **ReadWeight**プロパティに、**ZeroValid**プロパティがTrueの時に0の値を受け取る事を追加しました。
トーンインジケータ	CapMelodyプロパティ、MelodyTypeプロパティ、MelodyVolumeプロパティを追加しました。 メロディの機能が追加されました。 **Sound**メソッドの引数にsirenとmelodyが追加され機能が修正されました。。
バイオメトリクス	**StatusUpdateEvent**イベントに以下の値が追加されました。 BIO_SUE_STATUS_SENSOR_READY BIO_SUE_STATUS_SENSOR_COMPLETE
POS for .NET関連	Enumerationテーブルを変換できない項目の追加した適切な表記に更新しました。 MSDNのWebアドレスが間違っていましたので修正しました。 DevicePathプロパティの使用について注意書きを追加しました。
XMLPOS	Version1.13に更新する事で全ての図を更新しました。

以下が2010年11月11日版の修正の詳細一覧です。各詳細は、変更された項・章が記載されています。スペルミス等の軽微な変更は記載されていませんので御了承ください。

章または節	修正内容
キーロック	**ElectronicKeyValue**プロパティのデータ型をBSTRに修正
ラインディスプレイ	**CharacterSet**プロパティの値にDISP_CS_ANSIを追加
ＰＯＳプリンタ	印字データとエスケープシーケンスの説明と表にESC “|*”で始まる可変長データを含むデータの扱い方と、ESC “｜!″でを始まる個々の機能の無効化の記述が不足していたのを修正 赤色のエスケープシーケンスはカスタムカラーのエスケープシーケンスに統合されているため説明を削除 **AsyncMode**プロパティでTRUEの時に非同期に実行されるメソッドを追加 **CharacterSet**プロパティの値にPTR_CS_ANSIを追加 **RotatePrint**メソッドのPTR_RP_ROTATE180_BUFFEREDに関連する記述を削除
秤	**StatusNotify**プロパティを書き込み可能(R/W)に修正
CAT	StatusUpdateEventイベントの説明で”(新規)”を”第1.9版で更新”に修正
ポイントカード機	**CharacterSet**プロパティの値にPCRW_CS_ANSIを追加
RFIDスキャナ	**CapDisableTag**プロパティのデータ型をBOOLに修正 **ProtocolMask**プロパティの値にRFID_PR_ALLを追加
MICR(磁気インクリーダ)	参照先のUPOS仕様書Versioを1.13に修正
シグニチャ・キャプチャ	
フィスカルプリンタ	
リモートオーダーディスプレイ	
バンプバー	
モーションセンサ	
チェックスキャナ	目次、タイトル、一覧、概説などが不足していたものを追加
ベルト	
ゲート	
アイテムディスペンサ	
ライト	

付録B OPOSで定義するヘッダーファイル	デバイスクラス名とヘッダーファイルの対応で不足していたデバイスを追加 OPOS基本や各デバイスのヘッダーファイルでのAPG1.11～1.13の変更・追加が反映されていなかったものを修正 紙幣入金機・紙幣出金機・硬貨入金機・電子バリューリーダライタ・イメージスキャナ・RFIDスキャナの各デバイスのヘッダーファイル内容を追加
付録H システムマネジメントインフォメーション	新規機能：記述が不足していたのを追加

第1.14版

第1.14版は、スキャナ、秤、PINPad、電子バリューが更新されています。

以下が変更の詳細一覧です。各詳細は、変更された項・章が記載されています。スペルミス等の軽微な変更は記載されていませんので御了承ください。

章または節	修正内容
表紙ほか	発行日とVersionの変更
磁気ストライプリーダ	以下、UPOS 1.13 版で追加・変更された内容を反映しました。 「ＭＳＲの暗号化と、認証」の項目説明を新規に追加しました。 以下のプロパティ・メソッドについて、記述が明確になるよう変更しました。 **AccountNumber**プロパティ CapTrackDataMaskingプロパティ CardAuthenticationDataLengthプロパティ DecodeDataプロパティ ErrorReportingTypeプロパティ Track1EncryptedDataLengthプロパティ Track2EncryptedDataLengthプロパティ Track3EncryptedDataLengthプロパティ Track4EncryptedDataLengthプロパティ RetrieveCardPropertyメソッド
ＰＯＳプリンタ	スキャナー（バーコードスキャナー）への新規スキャナーコード追加を受け、POSプリンターの章に記載されなかったことの注釈を追加。（**PrintBarCode**メソッド）
スキャナ（バーコードリーダ）	**ScanDataType**プロパティに1D,2D,組み合わせ,郵便の各種バーコードを追加
秤	UPOS1.14版参照の記載を追加
PINPad	BeginEFTTransactionメソッドのPINPadSystemパラメータに、記述されている値以外を許し、それについての注意を追加しました

電子バリューリーダライタ	トレーニングモードをサポートするため、**CapTrainingMode**プロパティ、**TrainingModeState**プロパティを追加しました。 メソッドを呼び出して処理を要求するときや処理が完了して結果を通知するときに情報の受け渡しを行うためのタグ値の設定と取得を行うため、**CapPINDevice**プロパティ、**PINEntry**プロパティ、**ClearParameterInformation**メソッド、**RetrieveResultInformation**メソッド、**SetParameterInformation**メソッドを追加しました。 非同期処理中にデバイスからの双方向コミュニケーションを実現するため、**TransitionEvent**イベントを追加しました。 直前に実施した取引情報を取得するため、QueryLastSuccessfulTransactionResultメソッドを追加しました。
MICR(磁気インクリーダ)	参照先のUPOS仕様書Versionを1.14に修正
シグニチャ・キャプチャ フィスカルプリンタ リモートオーダーディスプレイ バンプバー モーションセンサ チェックスキャナ ベルト ゲート アイテムディスペンサ ライト	
付録B OPOSで定義するヘッダーファイル	OPOS1.13版までを掲載しています。OPOS1.14版の変更は、コモンコントロールオブジェクトと共に公開されるヘッダーファイル公開後に掲載します。
付録F XMLPOS Mapping Reference	注釈が開発者向けに追加されました。具体的には、適正な名前空間の参照方法が提供されない限り、実装事例を使ってはなりません。
付録H システムマネジメントインフォメーション	UPOS1.14版参照の記載を追加

以下が2013年11月27日版の変更の詳細一覧です。

付録B OPOSで定義するヘッダーファイル	OPOS1.14版のヘッダーファイルに対応 スキャナ（バーコードリーダ）、電子バリューリーダライタ各デバイスのヘッダーファイル内容を追加

第1.14.1版

第1.14.1版は、電子バリューが更新されています。

以下が変更の詳細一覧です。各詳細は、変更された項・章が記載されています。スペルミス等の軽微な変更は記載されていませんので御了承ください。

章または節	修正内容
表紙ほか	発行日とVersionの変更
電子バリューリーダライタ	1.14.1版において、保守・運用（設置や定時の業務）機能、非決済サービス（ポイントなど）機能、デバイスのエラーコードの通知内容が標準化されています。 追加・変更されたプロパティ、メソッド、イベントは以下の通りです。 以下のプロパティを追加しました。 **ServiceType** 以下のプロパティを変更しました。 **CapMembershipCertificate、CardServiceList、CurrentService、ReaderWriterServiceList、ResultCodeExtended** 以下のメソッドを追加しました。 **AccessData、ActivateEVService、CheckServiceRegistrationToMedium、CloseDailyEVService、DeactivateEVService、OpenDailyEVService、RegisterServiceToMedium、UnregisterServiceToMedium、UpdateData、UpdateKey** 以下のメソッドを変更しました。 **AccessLog、LockTerminal、RetrieveResultInformation、UnlockTerminal** 以下のイベントを変更しました。 **TransitionEvent** モデルの説明に、以下を追加しました。 サブサービスのライフサイクル サービスがバリエーションを持つ場合 電子バリューリーダライタデバイスと決済センタの接続形態

磁気ストライプリーダ	以下のプロパティ・メソッドについて、**BinaryConversion**プロパティに依存する記述を追加しました。 プロパティ **AdditionalSecurityInformation**、**CardAuthenticationData**、**Track1Data**、**Track1DiscretionaryData**、**Track1EncryptedData**、**Track2Data**、**Track2DiscretionaryData**、**Track2EncryptedData**、**Track3Data**、**Track3EncryptedData**、**Track4Data**、**Track4EncryptedData** メソッド **AuthenticateDevice**、**DeauthenticateDevice**、**RetrieveDeviceAuthenticationData**、**WriteTracks**
PINPad	以下のプロパティについて、**BinaryConversion**プロパティに依存する記述を追加しました。 Track1Data、Track2Data、Track3Data、Track4Data
付録B OPOSで定義するヘッダーファイル	OPOS1.14版までを掲載しています。OPOS1.14.1版の変更は、コモンコントロールオブジェクトと共に公開されるヘッダーファイル公開後に掲載します。
付録F XMLPOS Mapping Reference	バイナリーデータを XML データ構造に変換することに関する注釈を追加しました。 バイナリーデータを正しく送受信するための BinaryConversion を活用するデバイスの XML 記述例を追加しました。(例：″BIR プロパティの返答例″, BIR の GetProperty など) EVRW デバイスの新しく追加されたプロパティやメソッドが記載された図を更新しました。 各デバイスでのドメイン、プロパティ、メソッドの図を更新しました サンプルコードを更新しました。 以下のエラーコードを更新しました。 OPOS_E_CLOSE -> OPOS_E_CLOSED OPOS_E_NOTCALIM -> OPOS_E_NOTCLAIMED OPOS_E_EXIST -> OPOS_E_EXISTS
付録H システムマネジメントインフォメーション	UPOS1.14.1版参照の記載を追加

以下が2015年3月25日版の変更の詳細一覧です。

付録B OPOSで定義するヘッダーファイル	OPOS1.14.1版のヘッダーファイルに対応 電子バリューリーダライタのヘッダーファイル内容を追加

第1.15版

第1.15版は電子バリューリーダライタを更新しました。

以下が変更の詳細一覧です。各詳細は、変更された項・章が記載されています。スペルミス等の軽微な変更は記載されていませんので御了承ください。

章または節	修正内容
表紙ほか	発行日とVersionの変更
電子バリューリーダライタ	以下のCapプロパティを追加しました。 CapAdditionalSecurityInformation, CapAuthorizeCompletion, CapAuthorizePreSales, CapAuthorizeRefund, CapAuthorizeVoid, CapAuthorizeVoidPreSales, CapCashDeposit, CapCenterResultCode, CapCheckCard, CapDailyLog, CapInstallments, CapPaymentDetail, CapTaxOthers, CapTransactionNumber 以下の専用プロパティを追加しました。 CardCompanyID, CenterResultCode, DailyLog, LogStatus, PaymentCondition, PaymentDetail, PaymentMedia, SlipNumber, TransactionNumber, TransactionType 以下の専用メソッドを追加しました。 AccessDailyLog, AuthorizeCompletion, AuthorizePreSales, AuthorizeRefund, AuthorizeSales, AuthorizeVoid, AuthorizeVoidPreSales, CashDeposit, CheckCard TransitionEventイベントに以下の定数を追加しました。 EVRW_TE_CONFIRM_SEARCH_TABLE EVRW_TE_CONFIRM_PAYMENT_CONDITION EVRW_TE_CONFIRM_AUTHORIZE EVRW_TE_NOTIFY_CHECK_CARD EVRW_TE_NOTIFY_ELECT_PAYMENT_CONDITION
付録B OPOSで定義するヘッダーファイル	OPOS1.15版のヘッダーファイルに対応 電子バリューリーダライタのヘッダーファイル内容を追加

第1.16版

第1.16版は、新デバイスとして「リテールコミュニケーションサービスデバイス」である下記の9つのデバイスカテゴリーの章が追加されています。

第39章 ビデオキャプチャ
第40章 個体認識
第41章 サウンドレコーダ
第42章 音声認識
第43章 サウンドフレーヤ
第44章 音声合成
第45章 ジェスチャーコントロール
第46章 デバイスモニタ
第47章 グラフィックディスプレイ

パワーマネジメント、ライトが更新されています。

以下が変更の詳細一覧です。各詳細は、変更された項・章が記載されています。スペルミス等の軽微な変更は記載されていませんので御了承ください。

章または節	修正内容
表紙ほか	発行日とVersionの変更
ビデオキャプチャ	新規追加
個体認識	新規追加
サウンドレコーダ	新規追加
音声認識	新規追加
サウンドフレーヤ	新規追加
音声合成	新規追加
ジェスチャーコントロール	新規追加
デバイスモニタ	新規追加
グラフィックディスプレイ	新規追加
パワーマネジメント	以下のCapプロパティを追加しました。 CapBatteryCapacityRemainingInSeconds, CapVariableBatteryCriticallyLowThresholdInSeconds, CapVariableBatteryLowThresholdInSeconds, CapChargeTime 以下の専用プロパティを追加しました。 BatteryCapacityRemainingInSeconds, BatteryCriticallyLowThresholdInSeconds, BatteryLowThresholdInSeconds, ChargeTime StatusUpdateEventに以下の定数を追加しました。 PWR_SUE_BAT_CAPACITY_REMAINING_IN_SECONDS
ライト	以下のCapプロパティを追加しました。 CapPattern 以下の専用メソッドを追加しました。 switchOnMultiple, switchOnPattern, switchOffPattern
付録A　OPOSレジストリの利用法	デバイスの追加

付録B OPOSで定義するヘッダーファイル

OPOS1.16版のヘッダーファイルに対応

- ヘッダーファイル内容を追加
  ビデオキャプチャ
  個体認識
  サウンドレコーダ
  音声認識
  サウンドフレーヤ
  音声合成
  ジェスチャーコントロール
  デバイスモニタ
  グラフィックディスプレイ
- ヘッダーファイル内容を更新
  パワーマネジメント
  ライト

第1.16.1版

第1.16.1版は、POSプリンタ、スキャナ、CAT、電子バリューリーダライタが更新されています。

以下が変更の詳細一覧です。各詳細は、変更された項・章が記載されています。スペルミス等の軽微な変更は記載されていませんので御了承ください。

章または節	修正内容
はじめに（.NET版）	列挙体を更新しました。
POSプリンタ	UPOS 1.15.1 版で更新された内容を反映しました。 以下の専用メソッドを更新しました。 PrintBarCode
スキャナ	以下の専用プロパティを更新しました。 ScanDataType
CAT	以下の専用プロパティを更新しました。 DailyLog, PaymentDetail
電子バリューリーダライタ	以下の専用プロパティを更新しました。 DailyLog, PaymentDetail, ServiceType 以下の専用メソッドを更新しました。 ReadValue, RetrieveResultInformation, SetParameterInformation, TransactionAccess
付録Ｂ　OPOSで定義するヘッダーファイル	OPOS1.16.1版のヘッダーファイルに対応 POSプリンタ、スキャナ、秤、フィスカルプリンタ、PINパッド、電子バリューリーダライタのヘッダーファイル内容を追加

付　録　F

XMLPOS　XML POS Mapping Reference

この付録は、第1.12版で追加され、第1.13版で大きく更新されました。

概要第1．14版で更新

OPOS技術部会は、XMLをベースにしたWeb技術を追加するために構成されたコンポーネントを提供します。このＸＭＬの構成がＸＭＬＰＯＳです。

XMLPOSは基本的に、XMLドキュメント内や、XMLの要素の中に直接OPOSのプロパティ、メソッド、イベントの各パラメータをUMLでマッピングする事で、エンタープライズアプリケーションがリモートデバイスにアクセスする事を可能にし、OPOSを拡張しています。

XMLPOSには二つの面があります。

まずは、同じようにXMLタグ名称にOPOSのプロパティ、メソッド、イベントをマッピングすることです。同時に、適切なＷ３Ｃ XML Schemaの中のタグ名称のグループ化が出来るArts－XML Best Practicesを提供します。

XMLPOS要件項目

アプリケーションがリモート入力デバイスをサポートする。（スキャナなど）

アプリケーションがリモート出力デバイスをサポートする。（プリンタなど）

複数のアプリケーション間で出力デバイスを共有する。

現行のOPOSに準拠したアプリケーションの変更を最小限にする。

<オプション>OPOSのServiceObjectの変更を最小限にする。

異なるプラットフォームとの接続性

エンタープライズアプリケーションとデバイス間の相互接続性

現行のアプリケーション、サービスオブジェクトを（継続的に）継承されなければならない。

リモートデバイスにもかかわらず十分なパフォーマンス提供しなければならない。

デバイスタイプ毎の転送要求ではない。

Webサービス環境での効率的な処理

ブラウザでの効率的な処理

シングルコマンド、複数の組合せたコマンド環境で動作する必要があります。ひとつのメッセージにおいてひとつのコマンド出すことも出来るし、コマンドセットを発行することも出来ます。

仕様の範囲外

特殊な拡張

参照資料

ARTS-XML Data Dictionary

UnifiedPOS Retail Peripheral Architecture V1.13 もしくはそれ以降

ARTS,ARTS-XML Best Practices V2.1.0

[ISO2382]ISO/IEC2382-14:1997 Information Technology – Vocabulary Part 14 Reliability, Maintainability and Availability.

OPOSからXMLに移行に関して　　第1.14.1版で更新

このセクションでは、プロパティ、メソッド、イベントをXMLのTag名に移行するルールを記述します。

すべてのプロパティ、メソッド、イベントは、ATRSの「ARTS-XML Best Practices」に基づいてUpper Camel Case（単語の先頭キャラクタを大文字で表す）へ移行されます。

一貫性を保つため、一覧表には、OPOS表記で記述された現行の文字形式で表現します。

WAMPOSの様式に準拠して、バッファやOPOSオブジェクトはXMLPOSに適したXML要素として置き換えられます。

バイナリーデータは、「ARTS-XML Best Practices」で定義されたARTSBinaryを用いて、エンコード・デコードされなければなりません。

XMLPOSへの変更　　第1.13版で更新

XMLPOSが作成された時、Version1.12ではじめて標準仕様に組み込まれ、実装を保留され、UnifiedPOSが、ARTS XMLコミッティーのXMLメッセージ標準を基にして支えました。OFSC(Open Foodservice System Consortium)とOPOS-Jは、コンセプトの確証を実装し、Version1.13にてXMLPOSのアーキテクチャを変更することで、いくつかの問題を解決しました。

属性の使用は、1つのメッセージで属性を再利用する機能を限定します。例としては、メッセージは、開始時にDeviceEnabledをtrueにし、終了時にfalseにします。これは、全ての属性を変更する必要がある要素に適用されます。

アプリケーションは、特定の問題を解決する為に、要求と値を通知する為にプロパティとメソッドを発行する事ができる必要があります。これは、XML様式のTAGを繰り返す事で全ての要素を置き換えて要求を操作します。

UPOSの共通プロパティ、メソッド、イベントの再利用と、XMLも出るグループの使用に、まだ満足していない１か２の上位の要求される動作とリファレンスタイプコードでのXMLアクセス。

今日のプログラムの実践は、組み込まれたプロパティにアクセスする為にゲットとセットのプロパティを使用することを推奨します。これは、プロパティ名を<GetProperty>と<SetProperty>ノードでプロパティを囲んで操作するUnifiedPOSVersion1.13として、一貫性を保つ為に必要な組み合わせです。全てのデバイススキーマが、結果として変更されました。

XMLPOS構成の概要　　第1.14版で更新

OPOSのXML要件

ARTS標準に沿った一貫性の維持のために、OPOSは、XMLPOS開発でARTS-XML BEST Practicesに従うことを選択しました。ARTS-XML Best Practicesドキュメントは、ARTS-XMLスキーマを構築するために、Venetian Blined Design Methodologyで表記します。基本的に、これは、接点を基点にした方法論で、個々の接点は、スキーマの構築においてブロックを組上合わせる様に接続されます。

例

```
<DirectIO>
        <Command>0</Command>
        <Data>0</Data>
        <Object>String</Object>
</ DirectIO >
```

OPOSメソッドとイベントのXMLPOSへの変換

OPOSメソッドとイベントをXMLPOSへ変換する方法論は：

OPOSのメソッド、イベント名をXMLTag名に、大文字を先頭に続けて変換するのが最も効率の良い方法です。

ARTS-XMLデータ辞書には、前述のタグ名の定義をが掲載されています。

XMLの要素名は大文字を先頭につける（Pasca形式）変換を用います。ARTSXML委員会はARTSXMLSchemasを生成するのに最も適した方法を開発しました。"CR Best Practices V2.1.0 20070515.doc"中、章分類3.1項に最適解が記述されており、すべてのXML要素とアトリビュートに対して、先頭を大文字に記述する手法とのべられております。

XMLPOSスキーマのEvent中にPropertyが含まれます。従来からのOPOSのイベントモデルは、イベントを発行し、そのイベントの情報を持つプロパティをキューイングし、アプリケーションが受信するまで残ります。この動作は、典型的なローカル接続型POSアプリケーションでは何も課題なく動作しますが、遠隔操作型POSアプリケーションでは、動作完了までに少々時間が掛かります。従って、WS-POS標準の考え方の中では、イベントはイベント操作の過程で、適切なプロパティを送信する事が許容されます。従ってEventのXMLスキーマは、プロパティをキューイングするか、或いは、デバイスイベントスキーマの中で、Eventハンドリングの過程の一部としてプロパティを直接返す機能の両方をサポートします。

XMLPOSのスキーマは”xs:nil”を利用します。この仕組みでは、記述が空欄であっても、記述の型が許容できるものでなかったとしても、要素は正常として受け入れられます。ARTS標準は参照するプロパティの値を戻すサービスへのリクエストを明示する機能を使っています。

従って、このような要素の定義は、一見したところ問題が存在するかのように見えるヵ所が文書中数か所に散見されます。しかしながら、ARTS辞書委員会は、ARTS標準が包含する混合型からルート要素レベルまでのすべての要素への定義を提供する必要があると決定しております。その為、特別な定義が用いられても、辞書検索を可能にしています。

イベントは、XXXPropertyタイプを用い、XXXProteryグループといった記述を用いておりません。この必要性は、XXXPropertyグループの記述は、特別なデバイスへのそれに対応したPropertyを定義する事になるからにほかなりません。XXXPropertyグループの記述はXXXProperty型を形成する、OPOS共通プロパティとの組み合わせだからです。

注記：以下のXML例は“名前空間参照先”を含んでいます。参照先は、実際のファイルの置き場所ではありませんが、サポートファイルが見つけられる適正な名前空間が配置されている格納場所ではあります。

例えば、XMLPOSの参照先は、“http://www.nrf-arts.org/UnifiedPOS/namespace/” と見えますが、サポートファイルの

本当の置き場所ではありません。実装者は、これら参照先を実際の参照先に置き換えねばなりません。

つまり、アプリケーションがXMLPOSスキーマ例をアプリケーション構築の基礎情報として使う場合には、適正な名前空間の場所へ格納場所を置き換える必要があります。

OPOS同期XML通信

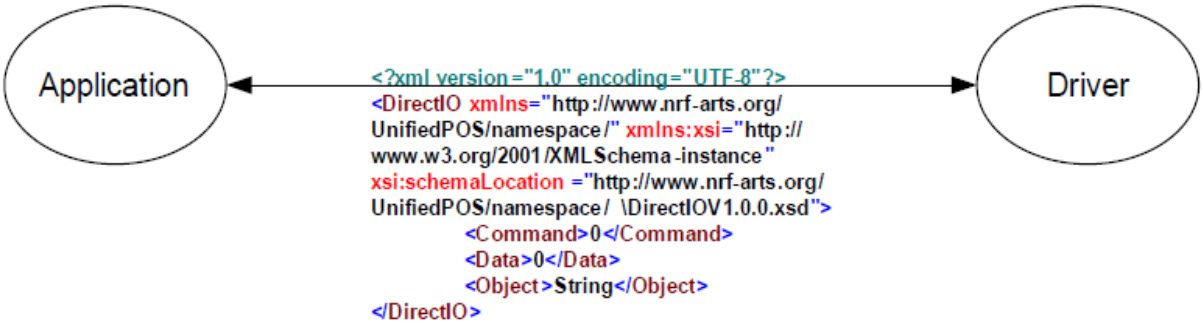


Figure 1: Synchronous Communications

同期環境は、おのおのの交換要求の情報を持つ結び付けられた両端によって表現されます。確立しているセッションで、唯一コマンド（ＯＰＯＳのプロパティとメソッドに代表される）とレスポンスは、結びついて横切る必要があります。ＸＭＬの世界では、各個々のコマンドとレスポンスが、メッセージになります。

それらＸＭＬメッセージを構築するために、ARTS XML Data Dictionaryで定義されたタグ、OPOS仕様から提供されるスキーマは、適度に形成されたXMLメッセージに移行する為に必要なツールとして連携しています。OPOSプロパティ、メソッド、イベントをXMLPOSメッセージに変換することは、程よく形成されたXMLヘッダでXMLPOS Embedded タグを包むことを意味します。

```
<?xml version=”1.0” encoding=”UTF-8”?>
<DirectIO xmlns=”http://www.nrf-arts.org/UnifiedPOS/namespace/”
xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
xsi:schemaLocation=”http://www.nrf-arts.org/UnifiedPOS/namespace/
¥DirectIOV1.14.1.xsd”>
                    <Command>0</Command>
                    <Data>0</Data>
                    <Object>String</Object>
</DirectIO>
```

OPOS非同期XML通信

非同期通信は、メッセージを要求したタイミングやソースの情報の重要性ではなく、アプリケーションから到着するメッセージによって現されます。図2、図3でＯＰＯＳプロパティ、メソッド、イベントがどのようにＸＭＬＰＯＳで使用されるＸＭＬメッセージに翻訳されるかを示します。

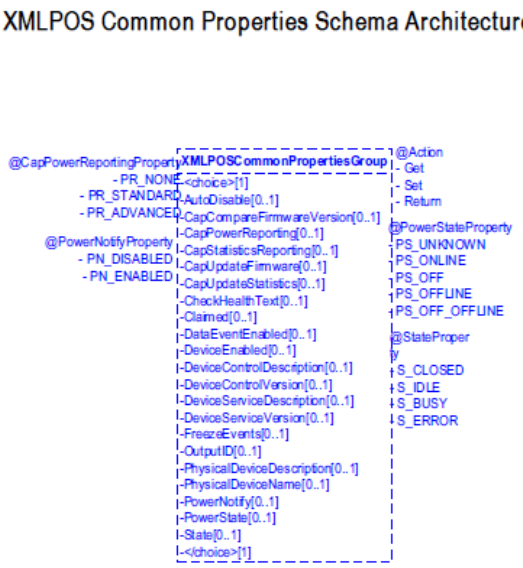


Figure 2: XML Common Properties Schema Architecture Sample

XMLPOS Common Methods Schema Architecture

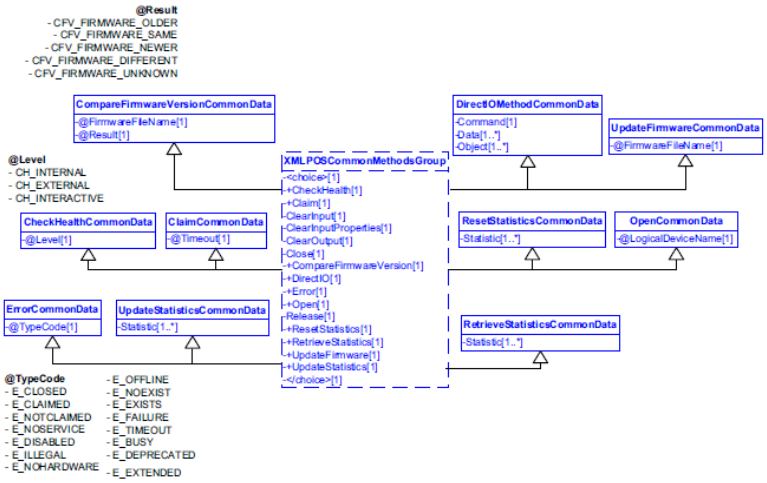


Figure 3: XML Common Methods Schema Architecture Sample

NOTE: @ - represents XPath nomenclature for an attribute

この分野では、ＯＰＯＳの共通メソッドを主に表現します。拡張によって派生するＸＭＬスキーマの方法論で各デバイス固有のＸＭＬスキーマを表します。概念として、これは、抽象的なベースクラスに似ています。

共通のデータの複雑なタイプのスキーマの構築は、共通プロパティ、メソッド、イベントのＸＭＬタグ名の要素を含んでいます。

各デバイスのOPOSAPIのためのノードの構築は、共通のデータの複雑なタイプのスキーマ(XMLPOSCommonData)からの拡大で引き出される共通データ複合スキーマの要素として提供されます。



Figure 4: Asynchronous Example

非同期でコマンドを送信する最初の例の中で、アプリケーションは、複数の個々のＸＭＬメッセージを転送します。これは、同期モデルの拡張ですが、各メッセージが転送されると共にメッセージのソースを見分ける追加的サポート情報を要求します。

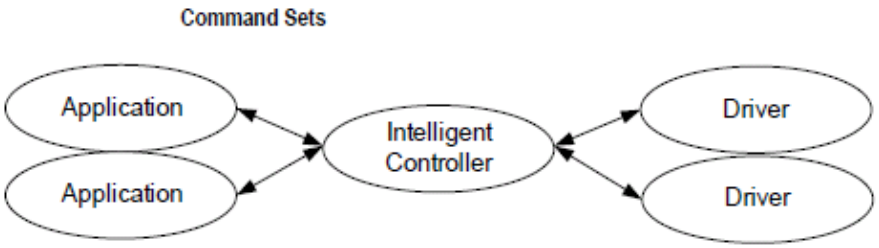


Figure 5: Asynchronous with Intelligent Controller Example

より効率的に利用可能な帯域幅を利用するために、一連の集められたメッセージの伝達を利用することができます。このより複雑な方法論は、Intelligent Controllerがアプリケーションと周辺機ドライバーの間に位置する事が必要です。 それはただ一つのXMLメッセージの中に、一つのより複雑なOPOSコマンドか、簡単なものを集めたより複雑なOPOSコマンドを使用する事を盛り込んでいます。 Intelligent Controllerは個々のOPOSコマンドの中のメッセージを分析して出力し、、適切なオーダーで適切なPeripheral Device Driverにそれらを適用します。

以下が、複数のOPOSコマンドを含んだひとつのXMLPOS Message Command Setを作成した例です。

```
<?xml version="1.0" encoding="UTF-8"?>
<CashDrawerDevice xmlns="http://www.nrf-arts.org/UnifiedPOS/namespace/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/namespace/
        \CashDrawerDeviceV1.14.1.xsd" MessageType="Request">
    <MessageID>12412341234</MessageID>
    <DateTime TypeCode="Message">2001-12-17T09:30:47.0Z</DateTime>
    <RequestID Name="String" Timestamp="2001-12-17T09:30:47.0Z">String
        </RequestID>
    <LogicalDeviceName>String</LogicalDeviceName>
    <CheckHealth Level="CH_INTERNAL"/>
    <ClaimType Timeout="0"/>
    <ClearInput/>
    <ClearOutput/>
    <Close/>
    <CompareFirmwareVersion Result="Text" FirmwareFileName="Text"/>
    <DirectIO Command="0" Object="String" Data="0"/>
    <Open LogicalDeviceName="String"/>
    <Release/>
    <ResetStatistics StatisticsBuffer="String"/>
    <RetrieveStatistics StatisticsBuffer="String"/>
    <UpdateFirmware FirmwareFileName="String"/>
    <UpdateStatistics StatisticsBuffer="String"/>
    <CashDrawerID>String</CashDrawerID>
    <OpenDrawer/>
    <WaitForDrawerClose BeepFrequency="0" BeepTimeout="0"
        BeepDuration="0" BeepDelay="0"/>
    <ClearInputProperties/>
</CashDrawerDevice>
```

OPOS XMLエラー

デバイスエラーコードとメッセージセベリティコード

デバイスエラーコードとメッセージセベリティコードの間で直接割り当てられる事が必要条件ではありません。デバイスエラーコードは、周辺デバイスや操作上の問題あるいはアクセスするための通信上のサービスから発生します。メッセージセベリティコードは、おのおののXMLメッセージに割り当てられ、そしてメッセージが、暗黙に操作順位を構築することによって如何に操作されるべきだったかを表現します。たとえば、プリンタカバーオープンはDevice Error Codeを生成します。XMLメッセージは、元来、このエラーが“information”のメッセージセベリティコードである事を通知しています。これは、まさにプリンター状態の受信部を通知することになります。ある時間の後に、Messageは、機器で障害が発生し、緊急の警戒が必要であることを述べるために、“Error”のメッセージセビリティであるデバイスエラーをあげることが可能です。

この論理に従って、デバイスエラーコードを伝えるほとんどのメッセージは、時間通りの操作がされていない場合に、時間が超えると注意が必要であることを示すためにひとつのメッセージセベリティコードを出力し始めます。Device Failure Codesは、いつもより高いメッセージセベリティコードレベルで開始するコードのクラスのひとつです。 いくつかの例が以下のテーブルに示されます。

メッセージセベリティコード

継続的に活動している定義で主要なISO規格は、Information Technology—Vocabulary –Reliability Maintainabilityのセクション14 and Availability ISO2382-14です。以下の定義はその仕様から引用しています。

セベリティコードはメッセージの優先権を特定します。 基本的に、セベリティコードには3つのタイプがあります。 それらは機器の1つの通常の操作の効果を特徴付けます。 次のいづれかの効果があります、それは、 機器の状態の変化する結果であり、あるいは、同じ状態にいますが、減少している機能性の結果を差し迫った問題として予測するかです。 最終的なカテゴリは、誤りでも警告でもなく、機器の状態に関する情報、すなわち簡単な情報です。

致命的なエラーは、機器の安定した状態に変化をさせます。これは、しばしば、機器そのものまたは機器の一部が機能しなくなる重要なまたは致命的なエラーです。

フォールトエラーは、デバイスの機能で重要でない、いずれかの機能部が停止したか差し迫ったエラーが発生する警告を示します。

そして最後は、冷凍室の温度。冷凍庫の温度なの簡単な通知です。

それら三つの対応は、セベリティ（エラー、警告、情報）として分類されます。

一般的に「セベリティコード」は、欠点を特定します。各一般的なセベリティコードは、自身の(任意)の記述があるゼロあるいは複数の機器固有の欠点コードを持ちます。とわいえ、実際に、エラーが統一して通知するべき一般的な名称は、「Error of Fault code」です。

セベリティエラー対応表

値	セベリティ	意味
OPOS_E_CLOSED	Warning	クローズされているデバイスにアクセスしようとしました
OPOS_E_CLAIMED	Information	別のプロセスが排他アクセスしているデバイスにアクセスしようとしました。このアクセスを行えるには、その前に別のプロセスが排他アクセス権を解放しなければなりません。排他使用デバイスの場合、アプリケーションがデバイスの排他アクセス権を獲得していないと、アクセスができません。
OPOS_E_NOTCLAIMED	Information	メソッドまたはプロパティ設定処理を使用する前に、排他アクセス権の獲得をしなければならない排他使用デバイスにアクセスしようとしました。デバイスを別のプロセスがすでに排他アクセスしている場合には、ステータス OPOS_E_CLAIMEDが戻されます。
OPOS_E_NOSERVICE	Warning	コントロールがサービスオブジェクトと通信できません。おそらく、セットアップエラーかコンフィギュレーションエラーを修正しなければなりません。
OPOS_E_DISABLED	Information	デバイスをディセーブルしているときには動作を実行できません。
OPOS_E_ILLEGAL	Error	デバイスに無効な動作か、サポートされていない動作を実行しようとしたか、無効なパラメータ値を使用しました。
OPOS_E_NOHARDWARE	Error	デバイスをシステムに接続していないか、電源が入っていません。
OPOS_E_OFFLINE	Warning	デバイスがオフラインです。
OPOS_E_NOEXIST	Error	ファイル名（またはほかの指定値）が存在していません。
OPOS_E_EXISTS	Error	ファイル名（またはほかの指定値）がすでに存在しています。

OPOS_E_FAILURE	Warning	デバイスがシデバイスがシステムに接続され、電源が入っていて、オンラインですが、リクエストされた処理をデバイスが実行できません
OPOS_E_TIMEOUT	Error	デバイスからの応答を待ち合わせていたサービスオブジェクトがタイムアウトしたか、サービスオブジェクトからの応答を待ち合わせていたコントロールがタイムアウトしました。
OPOS_E_BUSY	Warning	現在のSOの状態は、この要求を受け付けられません。例えば、非同期出力が実行中の場合、いくつかのメソッドは受け付けられません。
OPOS_E_DEPRECATED	Error	要求された処理が拒否されたので実行することが出来ません。
OPOS_E_CLOSED	Information	デバイスはOpen処理を実施されなければなりません。
OPOS_E_CLAIMED	Warning	Open処理は実行されていますが、ほかのアプリケーションが占有しているため、この時点でClaim処理を実行できません。
OPOS_E_NOTCLAIMED	Information	Open処理は実行されていますが、Claim処理が実行されていません。ほかにデバイスを占有しているアプリケーションが無いのでClaim処理を実行してください。
OPOS_E_DISABLED	Information	デバイスは、Open処理が実行され、排他制御デバイスの場合、Claim処理が実行されていますが、DeviceEnabledがfalseの状態です。

セベリティ状態対応表

値	セベリティ	意味
OPOS_SUE_POWER_ONLINE	Information	デバイスは通電され使用可能な状態です。
OPOS_SUE_POWER_OFF	Information	デバイスの電源がOFFか接続されていません。
OPOS_SUE_POWER_OFFLINE	Warning	デバイスは、通電されていますが、使用できない状態か要求に対して応答できない状態です。
OPOS_SUE_POWER_OFF_OFFLINE	Warning	デバイスが通電されていないかオフラインです。
OPOS_SUE_UF_PROGRESS+1 to 100	Information	ファームウェア更新処理が全体のXX%完了しました。
OPOS_SUE_UF_COMPLETE	Information	ファームウェアアップデートが正常に完了しました。この値は、SUE_UF_PROGRESS+100を意味します。
OPOS_SUE_UF_COMPLETE_DEV_NOT_RESTORED	Warning	ファームウェアの更新は成功しました。しかしながら、ＯＰＯＳサービスもしくはデバイスはファームウェア更新前である為、状態を返すことが出来ません。ＯＰＯＳサービスは総ての値を初期値に置き換えました。ＯＰＯＳサービスとデバイスの値の整合を保つために、アプリケーションは、ＯＰＯＳサービスを終了し、Open、Claim、DeviceEnabledの実行と全てのアプリケーション独自の設定を置き換える必要があります。
OPOS_SUE_UF_FAILED_DEV_OK	Error	ファームウェアの更新に失敗しましたが、デバイスは動作しています。
OPOS_SUE_UF_FAILED_DEV_UNRECOVERABLE	Error	ファームウエアの更新に失敗したので、使用することもソフトウェアで復旧することも出来ません。デバイスはOPOSサービスが操作可能状態に戻されることを要求します。

OPOS_SUE_UF_FAILED_DEV_NEEDS_FIRMWARE	Error	ファームウェアの更新が失敗しました。デバイスは更新が成功する他の対応がされるまで使用できません。
OPOS_SUE_UF_FAILED_DEV_UNKNOWN	Error	ファームウェア更新に失敗し、デバイスは不定の状態です。

OPOS同期XMLエラー

```
<?xml version="1.0" encoding="UTF-8"?>
<CashDrawerDevice xmlns="http://www.nrf-arts.org/UnifiedPOS/namespace/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/namespace/
        ..\CashDrawerDeviceV1.14.1.xsd">
    <Error TypeCode="E_CLOSED"/>
</CashDrawerDevice>
```

OPOS非同期XMLエラー

```
<?xml version="1.0" encoding="UTF-8"?>
<CashDrawerDevice xmlns="http://www.nrf-arts.org/UnifiedPOS/namespace/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/namespace/
        ..\CashDrawerDeviceV1.14.1.xsd" MessageType="Response">
    <MessageID>1242341234</MessageID>
    <DateTime TypeCode="Message">2001-12-17T09:30:47.0Z</DateTime>
    <RequestID Name="String" Timestamp="2001-12-17T09:30:47.0Z">String</
        RequestID>
    <Response ResponseCode="OK">
            <RequestID>String</RequestID>
            <ResponseTimestamp>2001-12-17T09:30:47.0Z</
                ResponseTimestamp>
            <ResponseDescription Language="eng">String</
                ResponseDescription>
            <BusinessError Severity="Information">
                    <ErrorID>String</ErrorID>
                    <Code>String</Code>
                    <Description Language="eng">String</Description>
                    <RelatedErrorID>String</RelatedErrorID>
            </BusinessError>
            <ResponderID/>
    </Response>
    <LogicalDeviceName>String</LogicalDeviceName>
    <Error TypeCode="E_CLOSED"/>
</CashDrawerDevice>
```

XMLPOS共通イベント

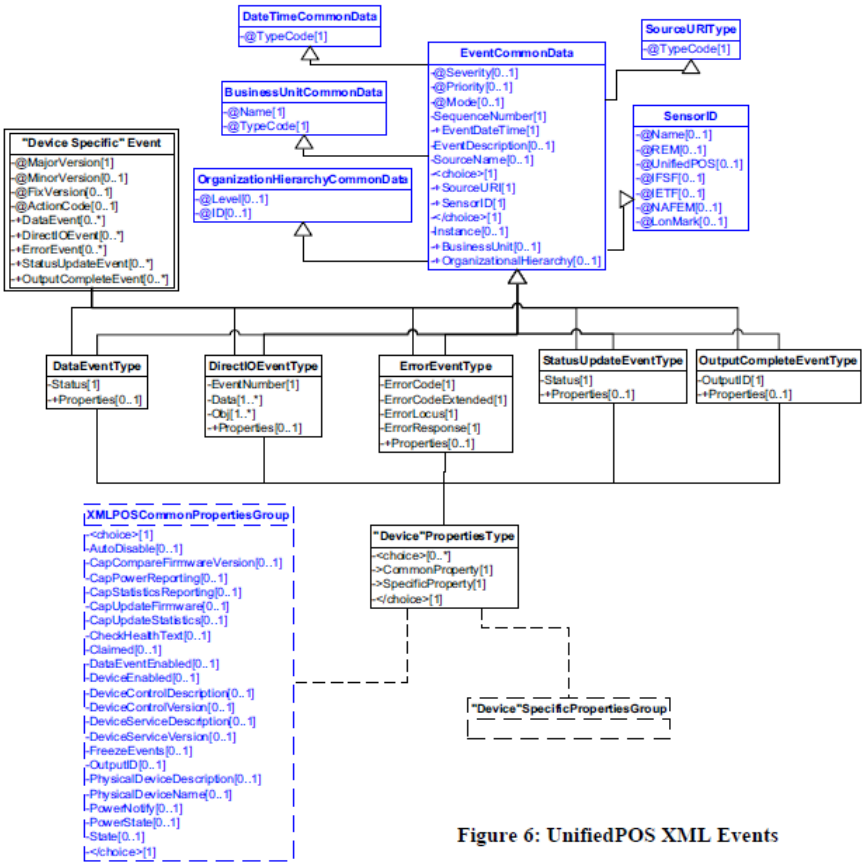


Figure 6: UnifiedPOS XML Events

OPOSのイベントを得るための手順は、アプリケーションが初めにイベントを受信しそしてその時に、発生したイベントをプロパティに関連付けます。イベントに応答するための時間を削減するために、WAMPOSはプロパティを得るために選択可能な二つの方法を紹介します。WAMPOSが紹介するプロパティの考え方は、イベントの結果として返す方法とイベントと共に送信される方法です。これは、イベントに応答し返信する回数を削減することになります。システムの必要性によって、XMLPOSは、どちらの代替手段もサポートするように設計されています。

OPOS同期XMLイベント

```
<?xml version="1.0" encoding="UTF-8"?>
<"DeviceSpecific" Event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="..\XMLPOSEventV1.14.1.xsd"
        Severity="Information" Mode="Production" Priority="-0">
    <SequenceNumber>4294967295</SequenceNumber>
    <EventDateTime TypeCode="Message">2001-12-17T09:30:47.0Z</
        EventDateTime>
    <SensorID UnifiedPOS="Scanner">0</SensorID>
    <DirectIOEvent EventNumber="0" Obj="String" Data="0"/>
</"DeviceSpecific"Event>
```

OPOS非同期XMLイベント

Single Events

```
<?xml version="1.0" encoding="UTF-8"?>
<"DeviceSpecific"Event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="..\XMLPOSEventV1.14.1.xsd"
        Severity="Information" Mode="Production" Priority="-0">
    <SequenceNumber>4294967295</SequenceNumber>
    <EventDateTime TypeCode="Message">2001-12-17T09:30:47.0Z</
        EventDateTime>
    <EventDescription>String</EventDescription>
    <SourceName>String</SourceName>
    <SourceURI>String</SourceURI>
    <Instance>String</Instance>
    <BusinessUnit TypeCode="RetailStore" Name="String">String</BusinessUnit>
    <OrganizationalHierarchy ID="String" Level="Corporation">String</
        OrganizationalHierarchy>
    <DirectIOEvent EventNumber="0" Obj="String" Data="0"/>
</"DeviceSpecific"Event>
```

Event Sets

```
<?xml version="1.0" encoding="UTF-8"?>
<"DeviceSpecific"Event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="..\XMLPOSEventV1.14.1.xsd"
        Severity="Information" Mode="Production" Priority="-0">
    <SequenceNumber>4294967295</SequenceNumber>
    <EventDateTime TypeCode="Message">2001-12-17T09:30:47.0Z</
        EventDateTime>
    <EventDescription>String</EventDescription>
    <SourceName>String</SourceName>
    <SourceURI>String</SourceURI>
    <Instance>String</Instance>
    <BusinessUnit TypeCode="RetailStore" Name="String">String</BusinessUnit>
    <OrganizationalHierarchy ID="String" Level="Corporation">String</
        OrganizationalHierarchy>
    <DataEvent Status="0"/>
    <DirectIOEvent EventNumber="0" Obj="String" Data="0"/>
    <ErrorEvent ErrorLocus="EL_INPUT" ErrorResponse="ER_RETRY"
        ErrorCode="0" ErrorCodeExtended="0"/>
    <StatusUpdateEvent Status="0"/>
    <OutputCompleteEvent OutputID="0"/>
</"DeviceSpecific"Event>
```

XMLPOS共通プロパティ



Figure 7: UnifiedPOS XMLPOS Common Properties

各型を含むXMLPOS 共通プロパティは、すべての装置カテゴリによって使用される、OPOSプロパティを一つに纏めたものです。そして、それぞれの個々の装置カテゴリで標準のXMLスキーマの拡張メカニズムを使用することで示されます。この各型は、各戸別のデバイスごとに描かれる箱によって表現されます。

XMLPOS共通データ

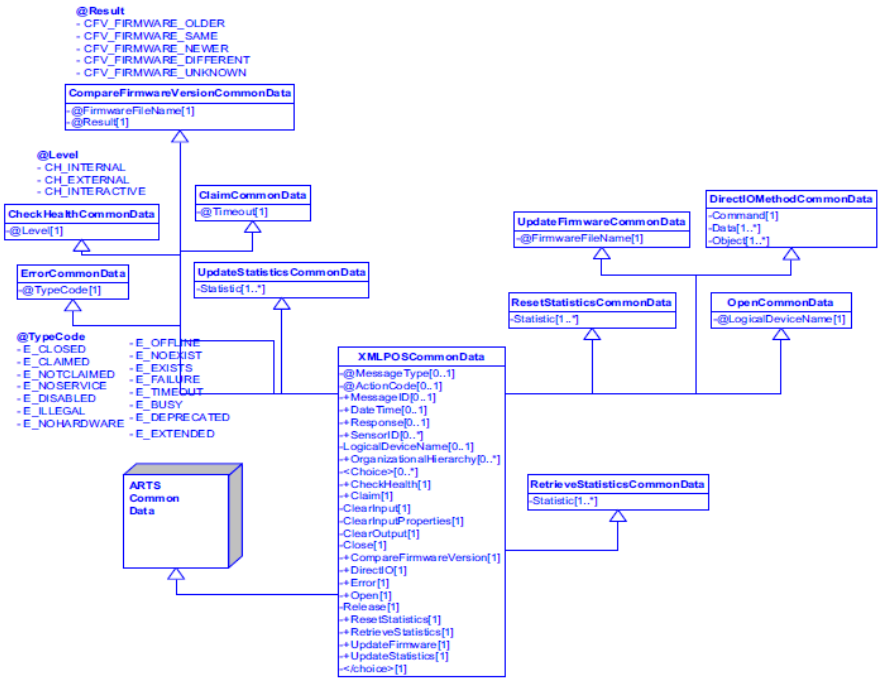


Figure 8: UnifiedPOS XMLPOS Common Data

XMLPOS共通データは、ＯＰＯＳ共通メソッドを加えて、ＸＭＬＰＯＳ共通プロパティとＡＲＴＳ共通データの両方を持ちます。共通の本質のために、この各タイプは、各戸別のデバイスごとに描かれる箱によって提供されます。

ARTS共通データ

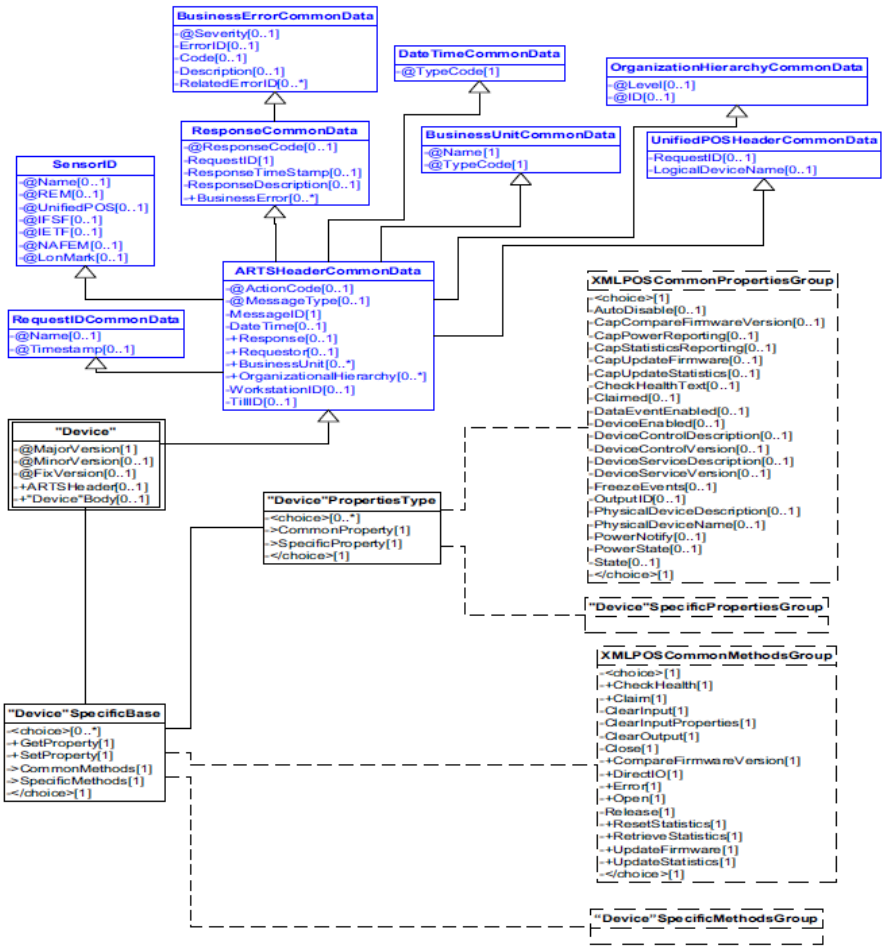


Figure 9: ARTS Common Data

ＡＲＴＳ　XMLは、全てのＡＲＴＳ　XMLスキーマを横断的に使うために共通の要素と構成をセットとして明らかにしています。この共通ヘッダーには、各型と標準の要求や応答のような状況とビジネスエラーを通知するメカニズムをハンドルするセットを含みます。共通の本質のために、この各タイプは、各戸別のデバイスごとに描かれる箱によって提供されます。

OPOSデバイス

各デバイスカテゴリのドメインビューは、以下の図で表されます。「デバイスカテゴリー」は、デバイスに特定のプロパティとメソッドを含む特定のデバイススキーマで表されます。以下に各デバイスカテゴリーのドメインビューの詳細を描写したセクションで、各デバイスに固有のプロパティ、メソッド、イベントイベントビューが描かれています。。

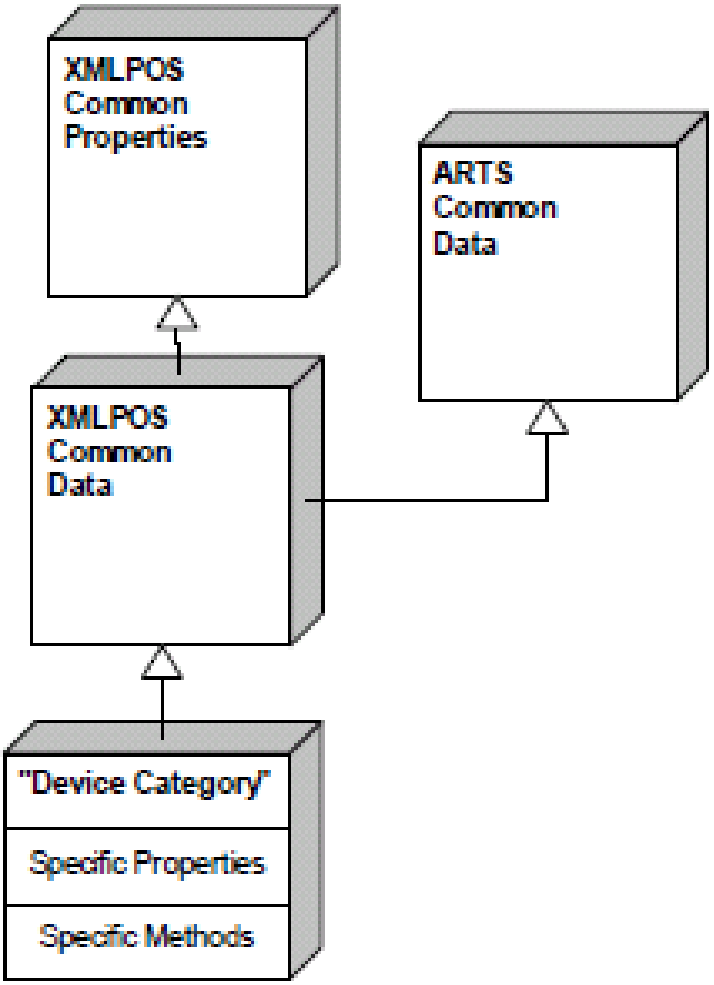


Figure 10: "Device Category" Domain View

Bill Acceptor

Bill Acceptor Example

→ SetRealTimeDataEvents to true

```
<?xml version="1.0" encoding="UTF-8"?>
<BillAcceptor xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/BillAcceptor/
        BillAcceptorV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        BillAcceptor/" MajorVersion="1" MinorVersion="14"FixVersion="1">
    <ARTSHeader MessageType="Request" ActionCode="Update">
            <MessageID>123412341234143</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="BillAcceptor">1</SensorID>
    </ARTSHeader>
    <BillAcceptorBody>
            <GetProperty>
                    RealTimeDataEnabled>true</RealTimeDataEnabled>
            </GetProperty>
    </BillAcceptorBody>
</BillAcceptor>
```

→:BeginDeposit()

```
<?xml version="1.0" encoding="UTF-8"?>
<BillAcceptor xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/BillAcceptor/
        BillAcceptorV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        BillAcceptor/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request">
            <MessageID>asdf2345sdfg</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="BillAcceptor">1</SensorID>
    </ARTSHeader>
    <BillAcceptorBody>
            <BeginDeposit/>
    </BillAcceptorBody>
</BillAcceptor>
```

Cash is accepted

← DataEvent is fired

```
<?xml version="1.0" encoding="UTF-8"?>
<BillAcceptorEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        BillAcceptorEvents/ ../BillAcceptorEventV1.14.1.xsd" xmlns="http://
        www.nrf-arts.org/UnifiedPOS/BillAcceptorEvents/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <DataEvent Severity="Information">
        <SequenceNumber>4294967295</SequenceNumber>
        <EventDateTime>2001-12-17T09:30:47.0Z</EventDateTime>
        <SensorID UnifiedPOS="BillAcceptor">1</SensorID>
        <Status>0</Status>
    </DataEvent>
</BillAcceptorEvent>
```

→ EndDeposit()

```
<?xml version="1.0" encoding="UTF-8"?>
<BillAcceptor xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/BillAcceptor/
        BillAcceptorV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        BillAcceptor/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request">
            <MessageID>asdf2345sdfg</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="BillAcceptor">1</SensorID>
    </ARTSHeader>
    <BillAcceptorBody>
            <EndDeposit>
                    <Success>BACC_DEPOSIT_COMPLETE</Success>
            </EndDeposit>
    </BillAcceptorBody>
</BillAcceptor>
```

Bill Acceptor Domain

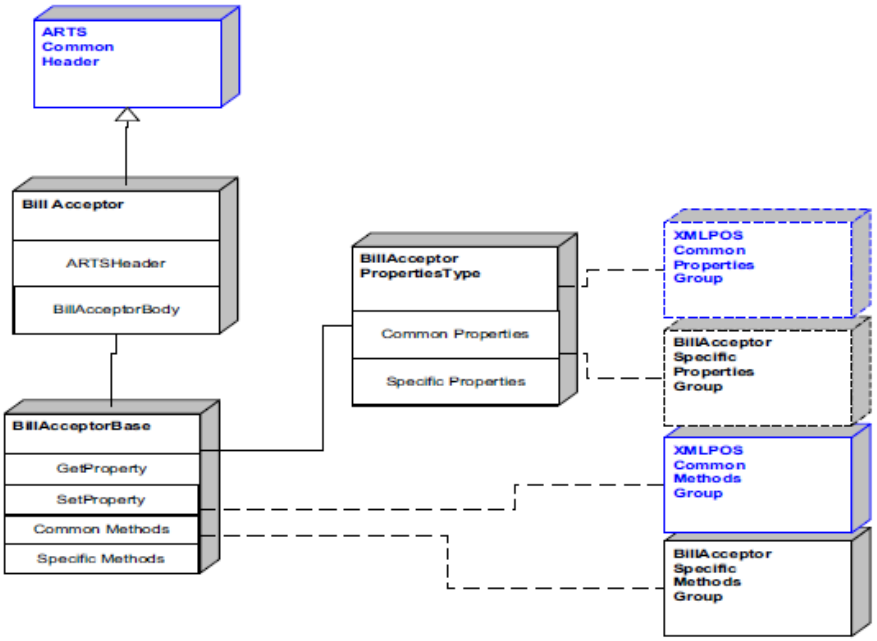


Figure 15: Bill Acceptor Domain View

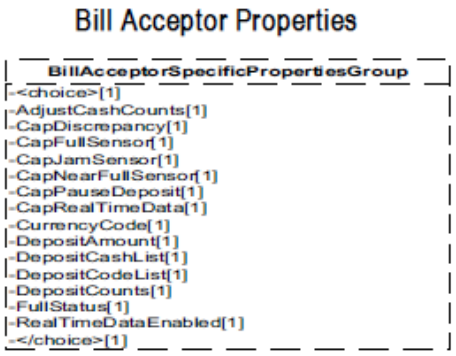


Figure 16: Bill Acceptor Properties Domain View

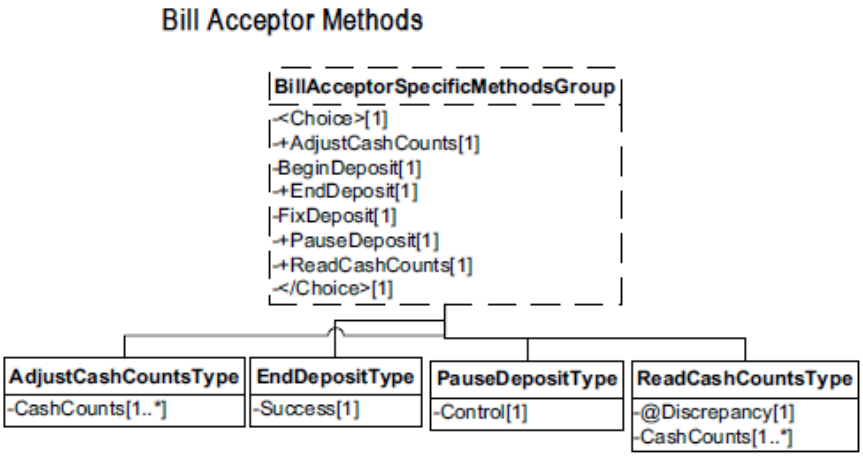


Figure 17: Bill Acceptor Methods Domain View

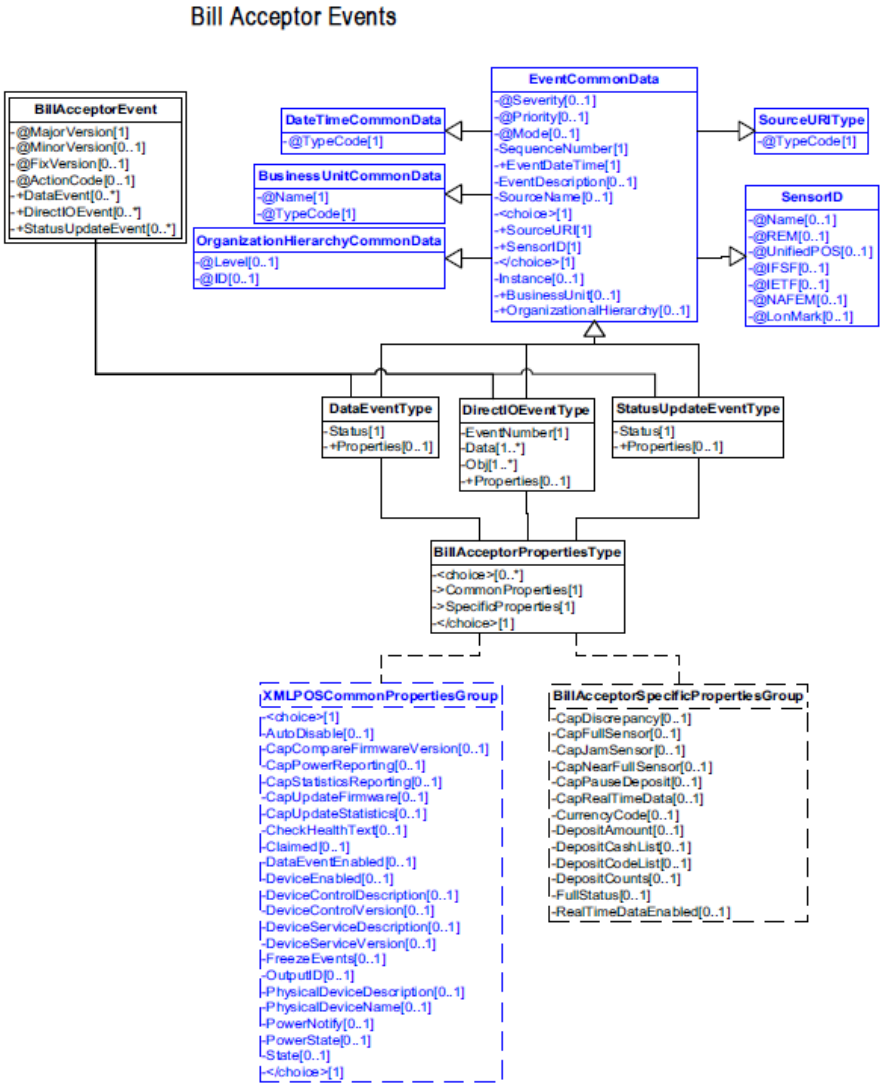


Figure 18: Bill Acceptor Events Domain View

Device Error Codes to Message Severity Codes

This device only has common errors as defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
N/A			

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
BACC_STATUS_FULL	Warning	Some cash slots are full
BACC_STATUS_NEARFULL	Warning	Some cash slots are nearly full
BACC_STATUS_FULLOK	Information	No cash slots are either full or nearly full.
BACC_STATUS_JAM	Error	A mechanical fault has occurred.
BACC_STATUS_JAMOK	Information	A mechanical fault has recovered.

Bill Dispenser

Bill Dispenser Example

→DispenseCash(“;100:4”)Dispense 4 $1.00 bills

```
<?xml version="1.0" encoding="UTF-8"?>
<BillDispenser xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/BillDispenser/
        BillDispenserV1.14.1.xsd"xmlns="http://www.nrf-arts.org/UnifiedPOS/
        BillDispenser/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>123421342134</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="BillDispenser">100</SensorID>
    </ARTSHeader>
    <BillDispenserBody>
            <DispenseCash>
            <CashCounts Denomination="100">4</CashCounts>
            </DispenseCash>
    </BillDispenserBody>
</BillDispenser>
```

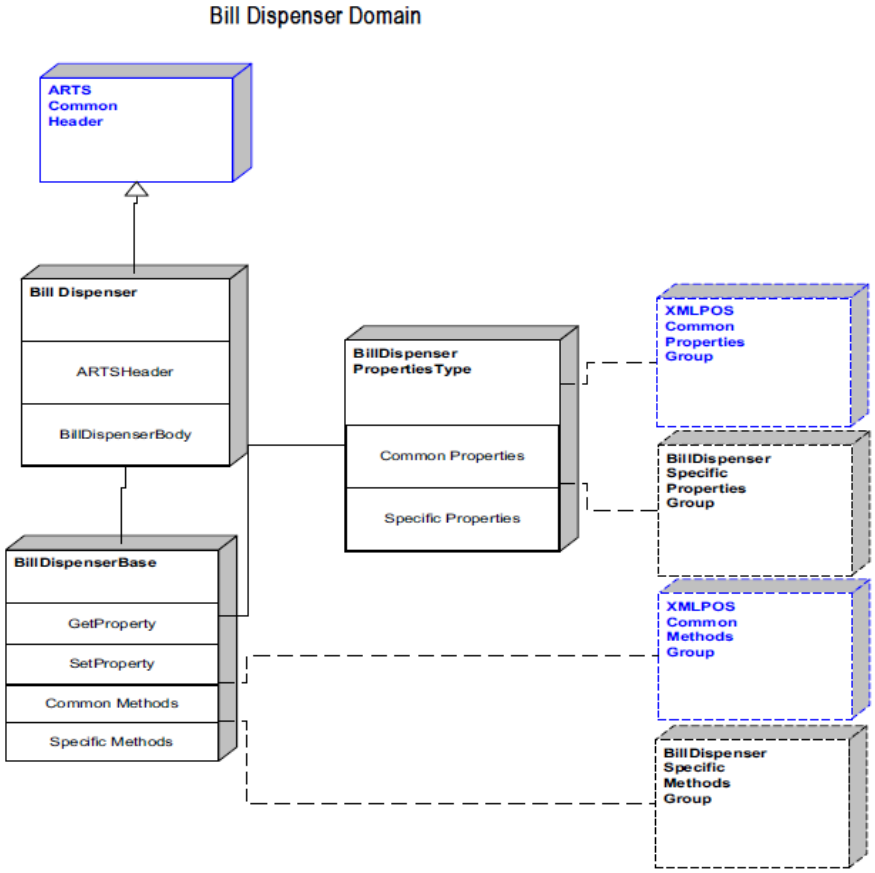


Figure 19: Bill Dispenser Domain View

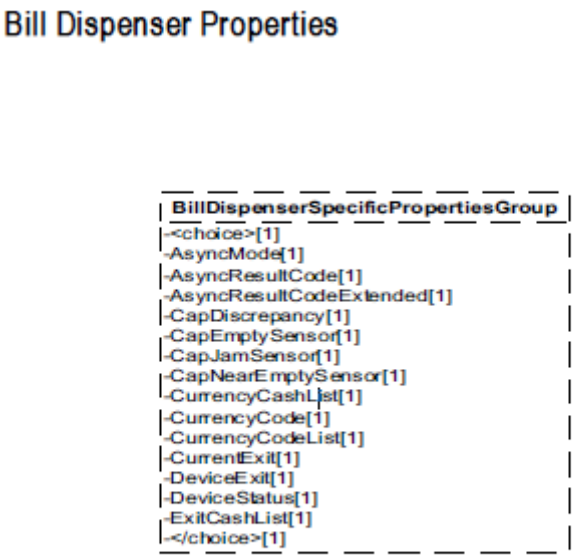


Figure 20: Bill Dispenser Properties Domain View

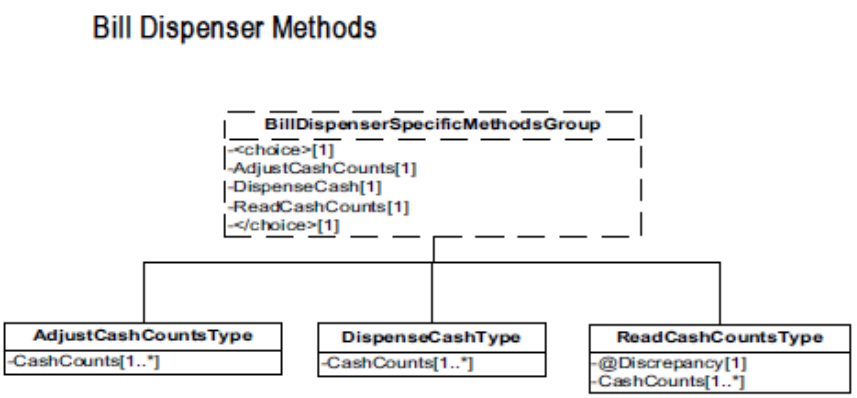


Figure 21: Bill Dispenser Methods Domain View

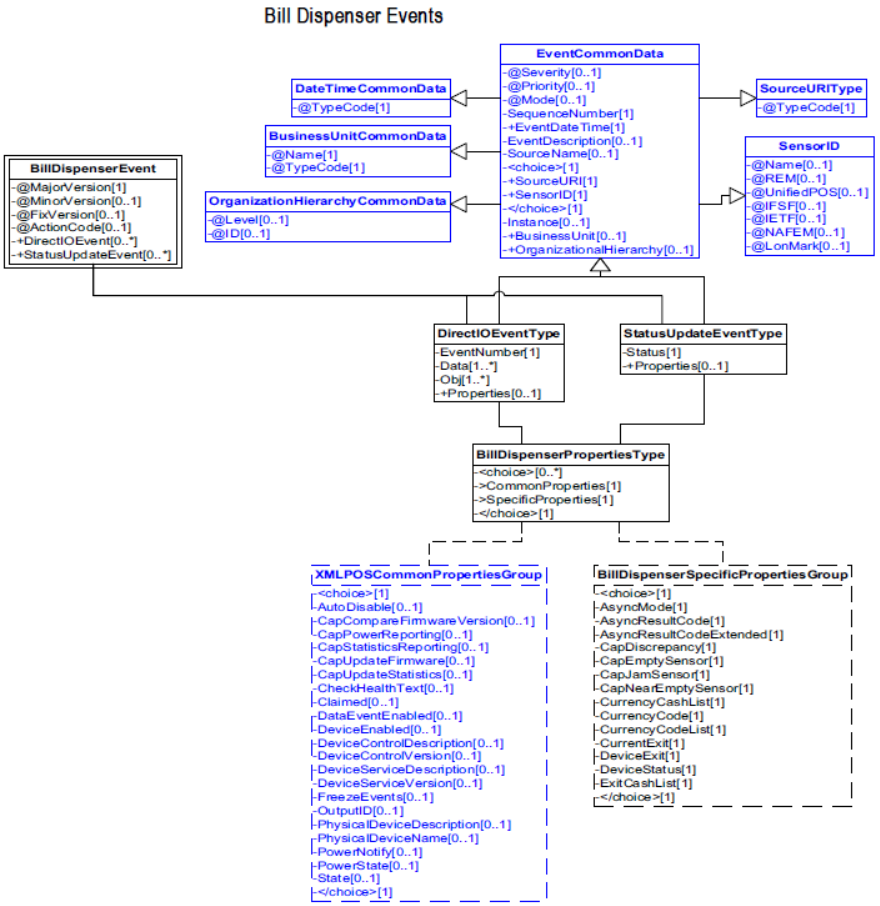


Figure 22: Bill Dispenser Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
dispenseCash	EBDSP_OVERDISPENSE	Warning	The specified cash cannot be dispensed because of a cash shortage.

Status Codes to Message Severity Codes

Device Specific Status Messages

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Value	Severity	Meaning
BDSP_STATUS_EMPTY	Warning	Some cash slots are empty.
BDSP_STATUS_NEAREMPTY	Warning	Some cash slots are nearly empty
BDSP_STATUS_EMPTYOK	Information	No cash slots are either empty or nearly empty
BDSP_STATUS_JAM	Error	A mechanical fault has occurred.
BDSP_STATUS_JAMOK	Information	A mechanical fault has recovered
BDSP_STATUS_ASYNC	Information	Asynchronously performed method has completed.

Biometrics

Biometrics Example

→ SetDataEventEnabled to true

```
<?xml version="1.0" encoding="UTF-8"?>
<Biometrics xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/Biometrics/
        BiometricsV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        Biometrics/" MajorVersion="1" MinorVersion="14" FixVersion="1">
  <ARTSHeader MessageType="Request" ActionCode="Update">
            <MessageID>123412341234</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="Biometrics">001</SensorID>
  </ARTSHeader>
  <BiometricsBody>
            <SetProperty>
                  <DataEventEnabled>true</DataEventEnabled>
            </SetProperty>
  </BiometricsBody>
</Biometrics>
```

→:BeginEnrollCapture()

```
<?xml version="1.0" encoding="UTF-8"?>
<Biometrics xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/Biometrics/
        BiometricsV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        Biometrics/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request">
            <MessageID>12341234</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="Biometrics">1</SensorID>
    </ARTSHeader>
    <BiometricsBody>
            <BeginEnrollCapture>
                  <ReferenceBIR/>
                  <Payload/>
            </BeginEnrollCapture>
    </BiometricsBody>
</Biometrics>
```

Capture Biometric Data

← Data Event is fired

```
<?xml version="1.0" encoding="UTF-8"?>
<Biometrics xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/Biometrics/
        BiometricsV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        Biometrics/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <DataEvent Severity="Information">
            <SequenceNumber>4294967295</SequenceNumber>
            <EventDateTime>2001-12-17T09:30:47.0Z</EventDateTime>
            <SensorID UnifiedPOS="Biometrics">1</SensorID>
            <Status>BIO_DATA_VERIFY</Status>
    </DataEvent>
</BiometricsEvent>
```

→ GetBIR()

```
<?xml version="1.0" encoding="UTF-8"?>
<Biometrics xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/Biometrics/
        BiometricsV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        Biometrics/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request" ActionCode="Read">
            <MessageID>45674567</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="Biometrics">1</SensorID>
    </ARTSHeader>
    <BiometricsBody>
            <GetProperty>
                    <BIR/>
            </GetProperty>
    </BiometricsBody>
</Biometrics>
```

← BIR property returned

```
<?xml version="1.0" encoding="UTF-8"?>
<Biometrics xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/Biometrics/
        BiometricsV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        Biometrics/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Response" ActionCode="Read">
            <MessageID>asdfasfString</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <Response>
                    <RequestID>45674567</RequestID>
                    <ResponderID>500</ResponderID>
            </Response>
            <SensorID UnifiedPOS="Biometrics">1</SensorID>
    </ARTSHeader>
    <BiometricsBody>
            <GetProperty>
        <BIR>\u005b\u0029\u003e\u001e\u0030\u0000\u001d\u0030\u0030\u0030
        \u0031\u001d\u0032\u001d\u0033\u001d\u0031\u0032\u0033</BIR>
            </GetProperty>
    </BiometricsBody>
</Biometrics>
```

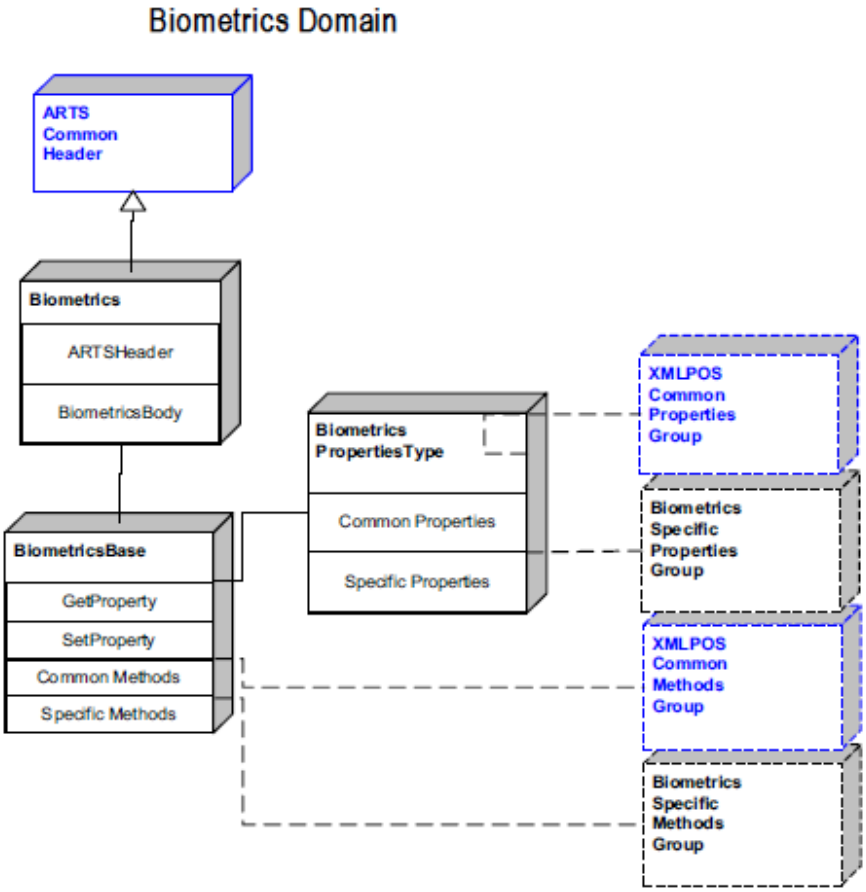


Figure 23: Biometrics Domain View

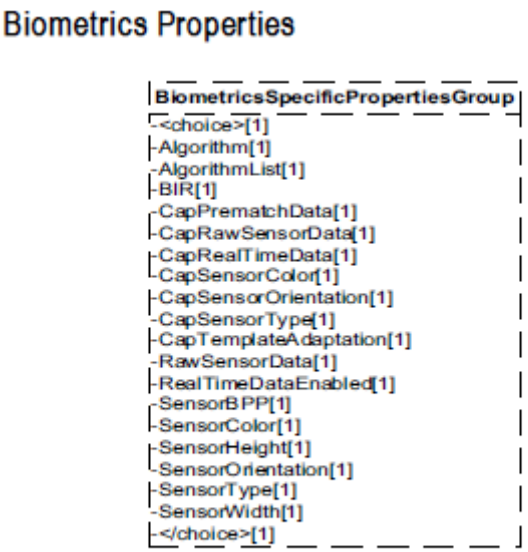


Figure 24: Biometric Properties Domain View

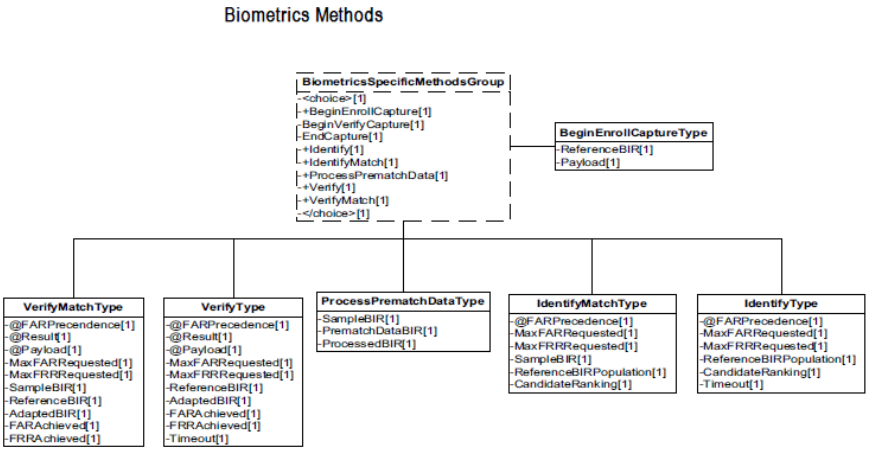


Figure 25: Biometric Methods Domain View

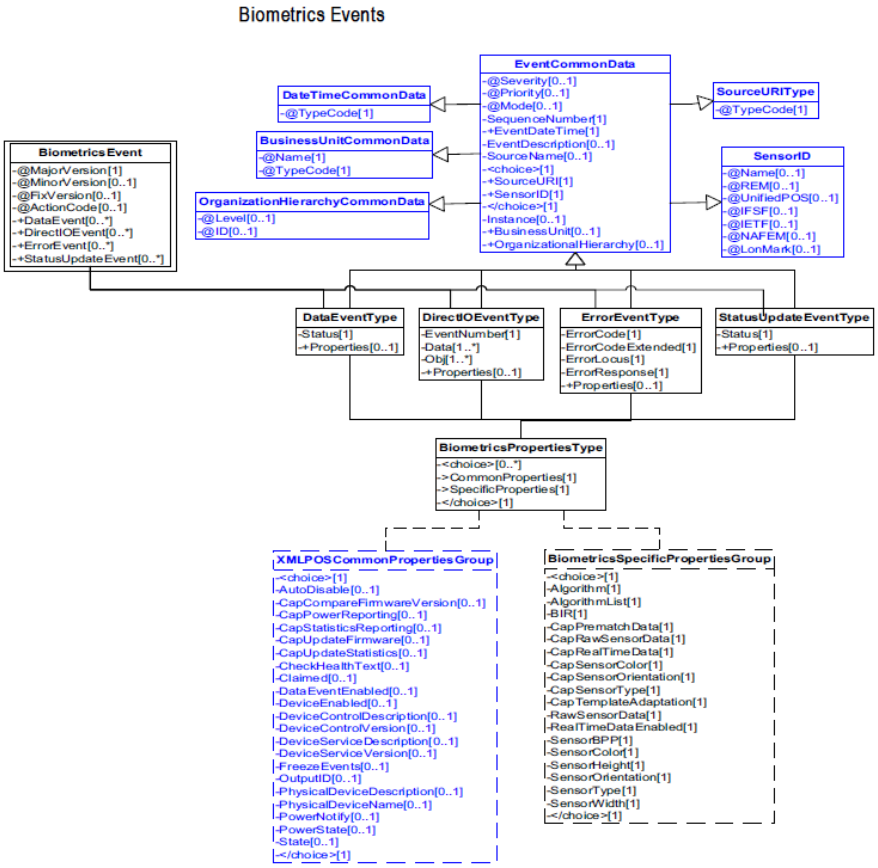


Figure 26: Biometrics Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
beginEnrollCapture	E_FAILURE	Warning	referenceBIR could not be adapted.
endCapture	E_ILLEGAL	Warning	Biometrics capture was not in progress.
identify	E_FAILURE	Error	referenceBIRPopulation was not valid.
identifyMatch	E_FAILURE	Error	referenceBIRPopulation was not valid.
processPrematchData	E_FAILURE	Error	sampleBIR was not valid.
verify	E_FAILURE	Error	referenceBIRPopulation was not valid.
verifyMatch	E_FAILURE	Error	referenceBIRPopulation was not valid.

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
BIO_SUE_RAW_DATA	Information	Raw image data is available
BIO_SUE_MOVE_LEFT	Warning	The position was too far to the right.
BIO_SUE_MOVE_RIGHT	Warning	The position was too far to the left
BIO_SUE_MOVE_DOWN	Warning	The position was too high
BIO_SUE_MOVE_UP	Warning	The position was too low
BIO_SUE_MOVE_CLOSER	Warning	The position was too far away
BIO_SUE_MOVE_AWAY	Warning	The position was too near (close)
BIO_SUE_MOVE_BACKWARD	Warning	The position was too far forward
BIO_SUE_MOVE_FORWARD	Warning	The position was too far backward
BIO_SUE_MOVE_SLOWER	Warning	The motion was too fast, move slower.
BIO_SUE_MOVE_FASTER	Warning	The motion was too slow, move faster.
BIO_SUE_SENSOR_DIRTY	Information	The sensor is dirty and requires cleaning
BIO_SUE_FAILED_READ	Warning	Unable to capture data from sensor
BIO_SUE_SENSOR_READY	Information	The sensor is ready to scan an object
BIO_SUE_SENSOR_COMPLETE	Information	The object scan has completed

Cash Changer

Cash Changer Example

→ SetDataEventEnabled to true

```
<?xml version="1.0" encoding="UTF-8"?>
<CashChanger xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CashChanger/
        CashChangerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        CashChanger/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request" ActionCode="Update">
            <MessageID Timestamp="2001-12-17T09:30:47.0Z">1234123
            </MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="CashChanger">1</SensorID>
    </ARTSHeader>
    <CashChangerBody>
            <SetProperty>
                    <DataEventEnabled>true</DataEventEnabled>
            </SetProperty>
    </CashChangerBody>
</CashChanger>
```

→:BeginDeposit()

```
<?xml version="1.0" encoding="UTF-8"?>
<CashChanger xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CashChanger/
        CashChangerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        CashChanger/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>1243124</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="CashChanger">1</SensorID>
    </ARTSHeader>
    <CashChangerBody>
            <BeginDeposit/>
    </CashChangerBody>
</CashChanger>
```

Cash is accepted

← DataEvent is fired

```
<?xml version="1.0" encoding="UTF-8"?>
<CashChangerEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        CashChangerEvents/ CashChangerEventV1.14.1.xsd" xmlns="http://
        www.nrf-arts.org/UnifiedPOS/CashChangerEvents/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <DataEvent Severity="Information" Mode="Production" Priority="-0">
            <SequenceNumber>4294967295</SequenceNumber>
            <EventDateTime>2001-12-17T09:30:47.0Z</EventDateTime>
            <SensorID UnifiedPOS="CashChanger">1</SensorID>
            <Status>0</Status>
    </DataEvent>
</CashChangerEvent>
```

→ EndDeposit(Change)

```
<?xml version="1.0" encoding="UTF-8"?>
<CashChanger xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CashChanger/
        CashChangerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        CashChanger/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>12341234</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="CashChanger">1</SensorID>
    </ARTSHeader>
    <CashChangerBody>
            <EndDeposit>
            <Success>CHAN_DEPOSIT_CHANGE</Success>
            </EndDeposit>
    </CashChangerBody>
</CashChanger>
```

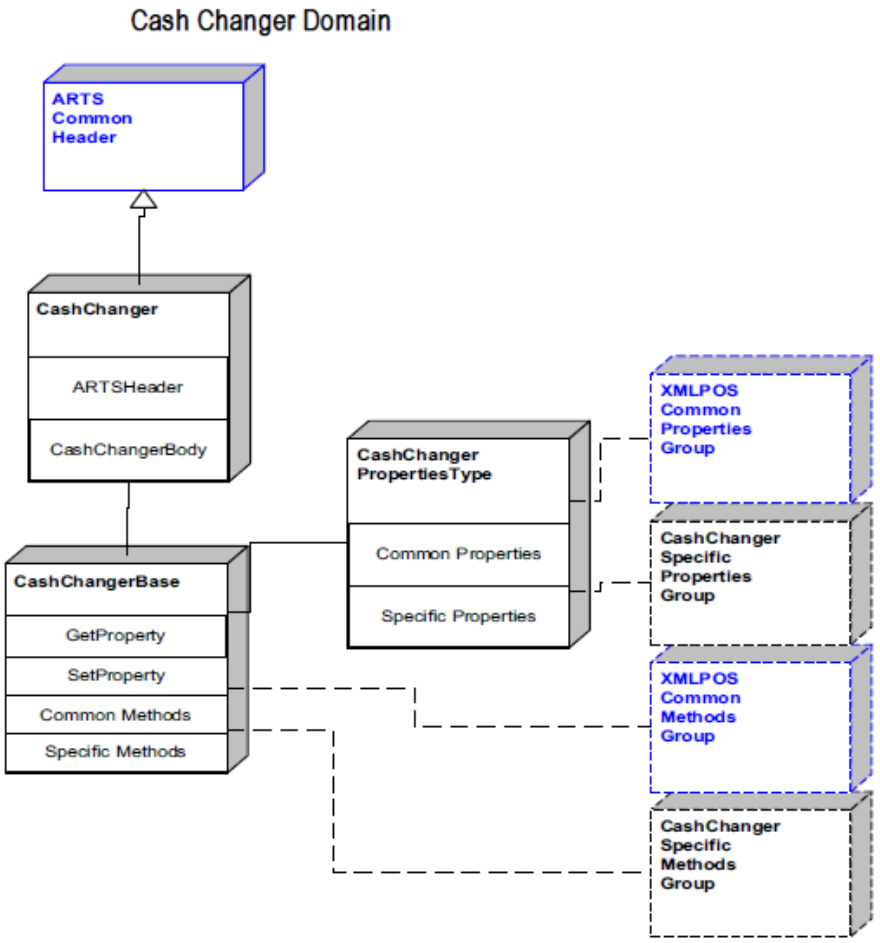


Figure 31: Cash Changer Domain View

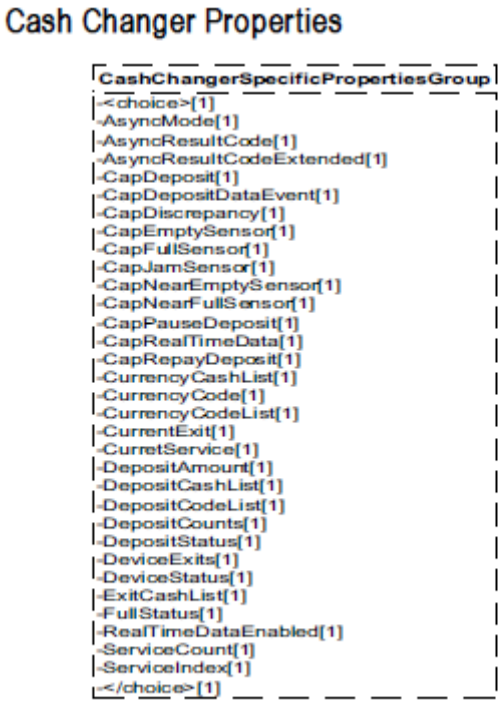


Figure 32: Cash Changer Properties Domain View

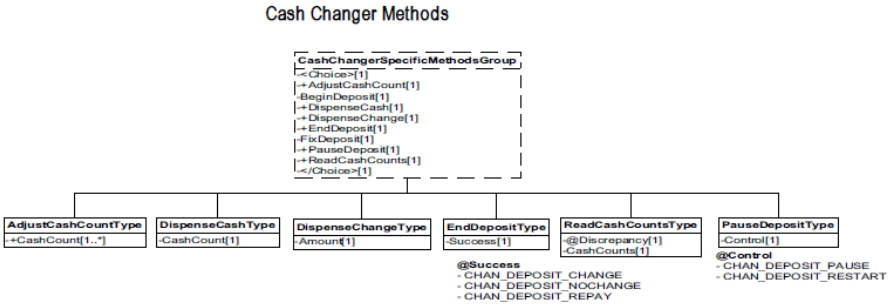


Figure 33: Cash Changer Methods Domain View

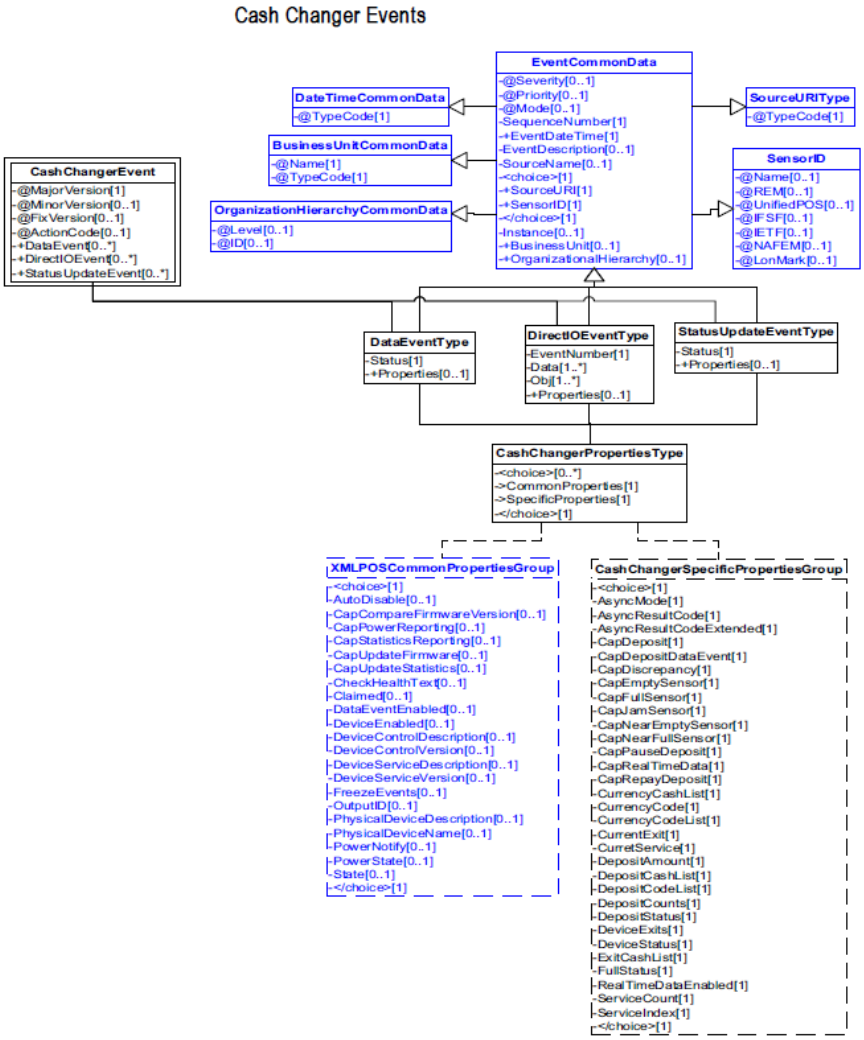


Figure 34: Cash Changer Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
beginDeposit			
	E_ILLEGAL	Error	Either the Cash Changer does not support cash acceptance, or the call sequence is not correct.
dispenseCash			
	E_BUSY	Warning	Cash cannot be dispensed because an asynchronous method is in progress.
	E_ILLEGAL	Error	One of the following errors occurred: • The cashCounts parameter value was illegal for the current exit. • Cash could not be dispensed because cash acceptance was in progress.
	E_EXTENDED	Error	ECHAN_OVERDISPENSE: The specified cash cannot be dispensed because of a cash shortage.
dispenseChange			
	E_BUSY	Warning	The specified change cannot be dispensed because an asynchronous method is in progress.
	E_ILLEGAL	Error	One of the following errors occurred: • A negative or zero amount was specified. • The amount could not be dispensed based on the values specified in ExitCashList for the current exit. • Change could not be dispensed because cash acceptance was in progress.
	E_EXTENDED	Error	ECHAN_OVERDISPENSE: The specified change cannot be dispensed because of a cash shortage.
endDeposit			
	E_ILLEGAL	Error	One of the following errors occurred: • Cash acceptance is not supported. • The call sequence is invalid. **beginDeposit** and **fixDeposit** must be called in sequence before calling this method.
fixDeposit			

	E_ILLEGAL	Error	One of the following errors occurred: • Cash acceptance is not supported. • The call sequence is invalid. beginDeposit must be called before calling this method.
pauseDeposit			
	E_ILLEGAL	Error	One of the following errors occurred: • Cash acceptance is not supported. • The call sequence is invalid. beginDeposit must be called before calling this method. • The deposit process is already paused and control is set to CHAN_DEPOSIT_PAUSE, or the deposit process is not paused and control is set to CHAN_DEPOSIT_RESTART.
readCashCounts			
	E_BUSY	Information	Cash units and counts cannot be read because an asynchronous method is in process.

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
CHAN_STATUS_EMPTY	Error	Some cash slots are empty
CHAN_STATUS_NEAREMPTY	Warning	Some cash slots are nearly empty.
CHAN_STATUS_EMPTYOK	Information	No cash slots are either empty or nearly empty.
CHAN_STATUS_FULL	Information	Some cash slots are full
CHAN_STATUS_NEARFULL	Information	Some cash slots are nearly full.
CHAN_STATUS_FULLOK	Warning	No cash slots are either full or nearly full.
CHAN_STATUS_JAM	Error	A mechanical fault has occurred
CHAN_STATUS_JAMOK	Information	A mechanical fault has recovered.
CHAN_STATUS_ASYNC	Information	Asynchronously performed method has completed.

Cash Drawer

Cash Drawer Example

→openDrawer()

```
<?xml version="1.0" encoding="UTF-8"?>
<CashDrawer xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CashDrawer/
        CashDrawerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        CashDrawer/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>1243124</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="CashDrawer">1</SensorID>
    </ARTSHeader>
    <CashDrawerBody>
            <OpenDrawer/>
            </CashDrawerBody>
    </CashDrawer>
```

← StatusUpdateEvent fired

```
<?xml version="1.0" encoding="UTF-8"?>
<CashDrawerEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        CashDrawerEvents/ CashDrawerEventV1.14.1.xsd" xmlns="http://
        www.nrf-arts.org/UnifiedPOS/CashDrawerEvents/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <StatusUpdateEvent Severity="Information">
            <SequenceNumber>4294967295</SequenceNumber>
            <EventDateTime>2001-12-17T09:30:47.0Z</EventDateTime>
            <SensorID UnifiedPOS="CashDrawer">1</SensorID>
            <Status>CASH_SUE_DRAWEROPEN</Status>
    </StatusUpdateEvent>
</CashDrawerEvent>
```

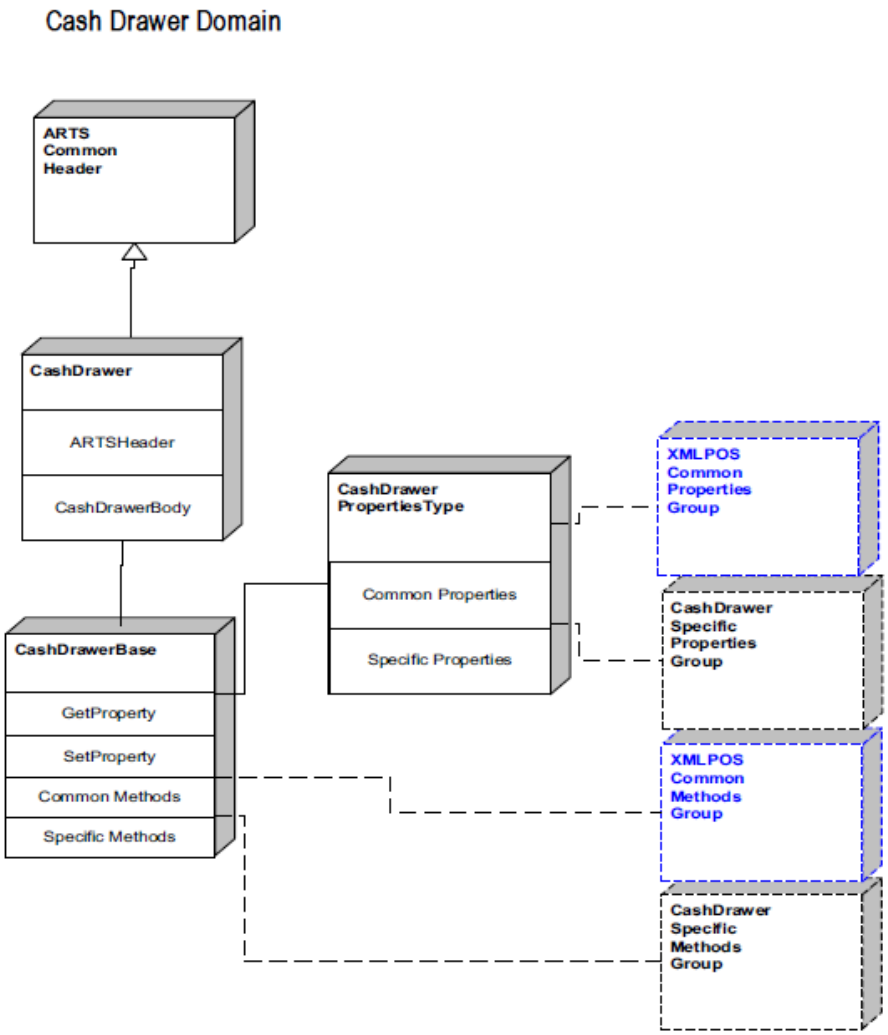


Figure 35: Cash Drawer Domain View

Cash Drawer Properties

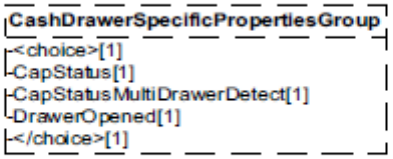


Figure 36: Cash Drawer Properties Domain View

Cash Drawer Methods

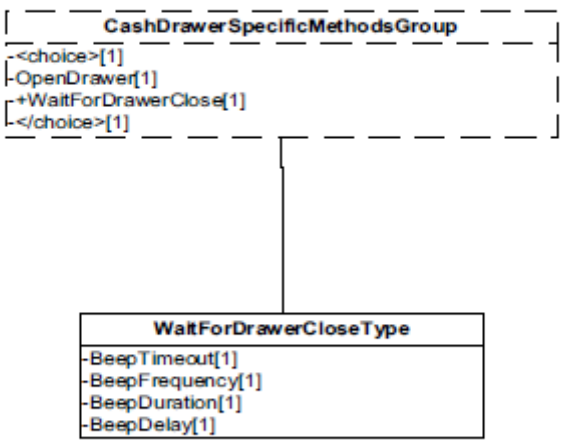


Figure 37: Cash Drawer Methods Domain View

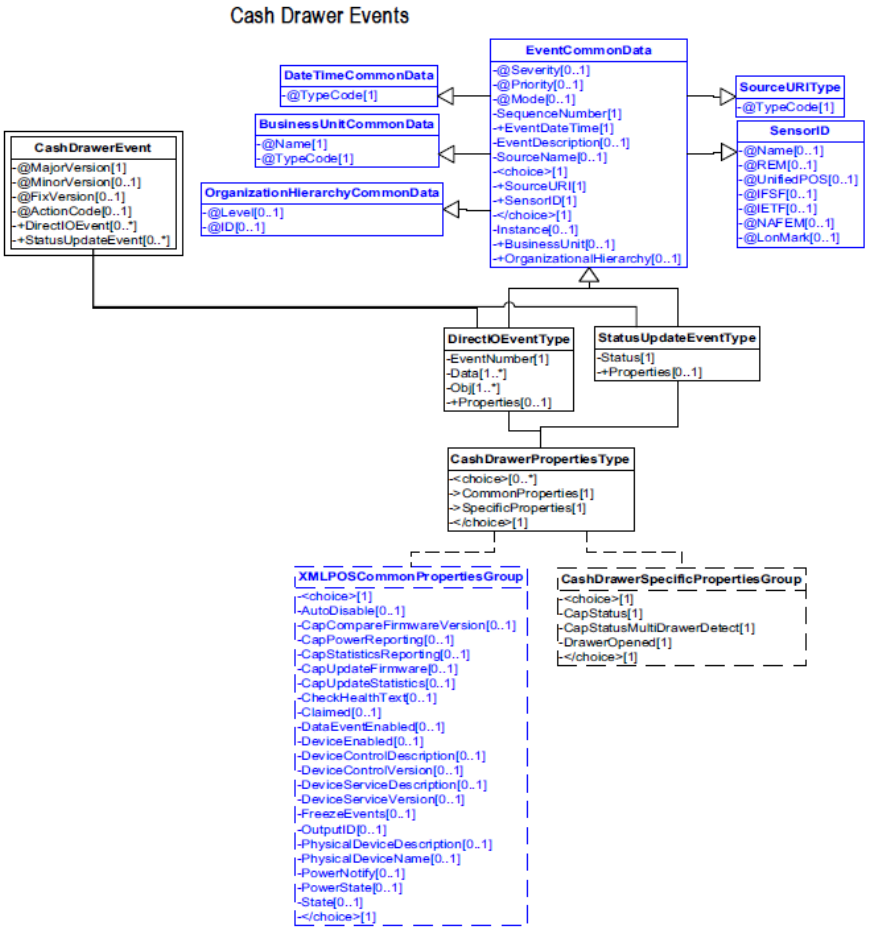


Figure 38: Cash Drawer Events Domain View

Device Error Codes to Message Severity Codes

This device only has common errors and they are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
N/A			

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
CASH_SUE_DRAWERCLOSED	Information	The drawer is closed.
CASH_SUE_DRAWEROPEN	Information	The drawer is open.

CAT

CAT Example

→set PaymentMedia

```
<?xml version="1.0" encoding="UTF-8"?>
<CAT xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CAT/
        CATV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/CAT/"
        MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request" ActionCode="Update">
            <MessageID>1234234</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="CAT">2</SensorID>
    </ARTSHeader>
    <CATBody>
            <SetProperty>
                    <PaymentMedia>CAT_MEDIA_CREDIT</PaymentMedia>
            </SetProperty>
    </CATBody>
</CAT>
```

→AuthorizeSales

```
<?xml version="1.0" encoding="UTF-8"?>
<CAT xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CAT/
        CATV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/CAT/"
        MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>2431243</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="CAT">100</SensorID>
    </ARTSHeader>
    <CATBody>
            <AuthorizeSales>
                    <SequenceNumber>1</SequenceNumber>
                    <Amount>100</Amount>
                    <TaxOthers>4</TaxOthers>
                    <Timeout>10</Timeout>
            </AuthorizeSales>
    </CATBody>
</CAT>
```

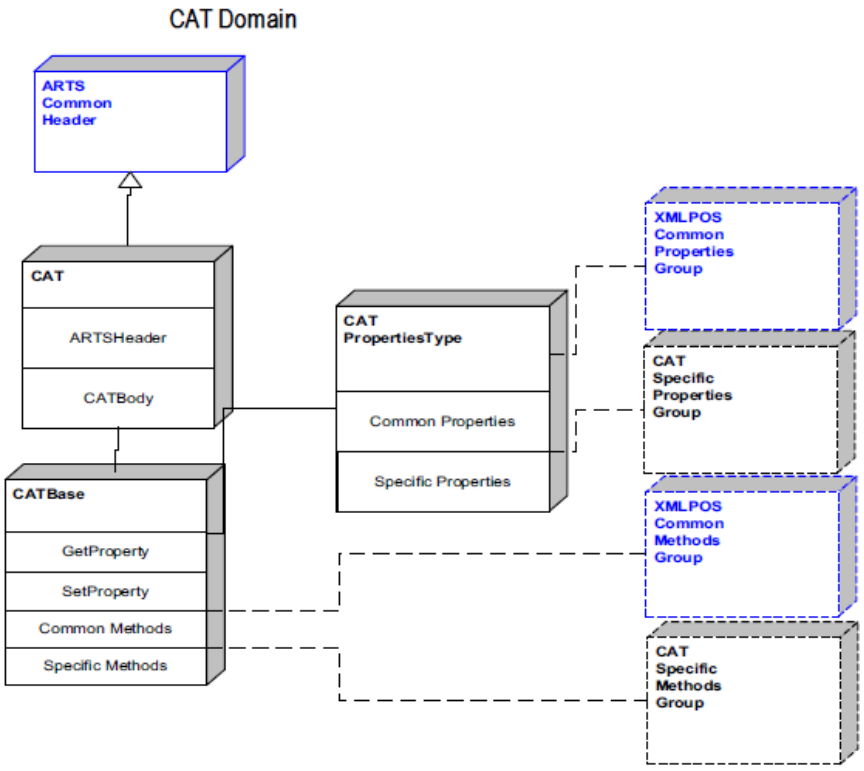


Figure 39: CAT Domain View

CAT Properties

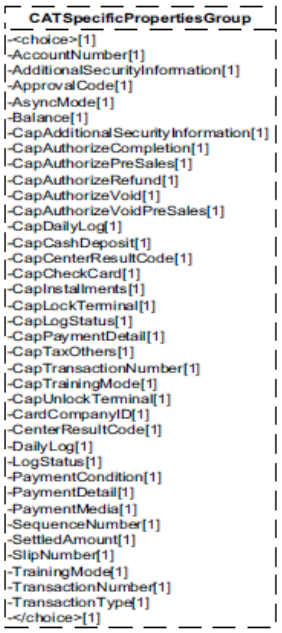


Figure 40: CAT Properties Domain View

CAT Methods

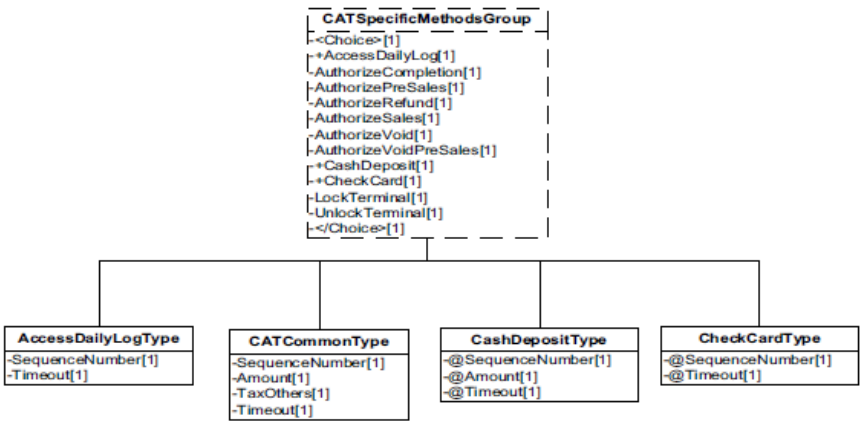


Figure 41: CAT Methods Domain View

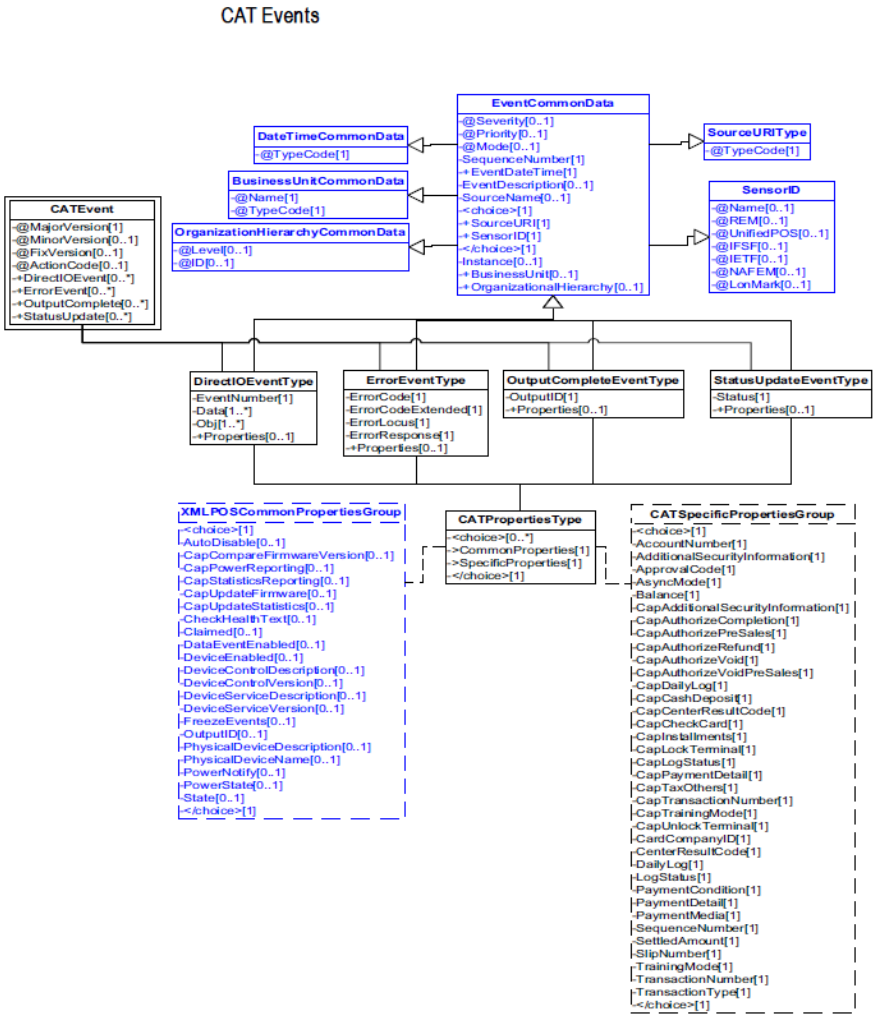


Figure 42: CAT Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
accessDailyLog			
	E_ILLEGAL	Error	Invalid or unsupported *type* or *timeout* parameter was specified, or **CapDailyLog** is false.
	E_TIMEOUT	Error	No response was received from CAT during the specified *timeout* time in milliseconds.
	E_BUSY	Warning	The CAT device cannot accept any commands now.
authorizeCompletion			
	E_ILLEGAL	Error	Invalid *timeout* parameter was specified, or **CapAuthorizeCompletion** is false.
	E_TIMEOUT	Warning	No response was received from CAT during the specified *timeout* time in milliseconds.
	E_BUSY	Information	The CAT device cannot accept any commands now.
authorizePreSales			
	E_ILLEGAL	Error	Invalid *timeout* parameter was specified, or **CapAuthorizePreSales** is false.
	E_TIMEOUT	Warning	No response was received from CAT during the specified *timeout* time in milliseconds.
	E_BUSY	Information	The CAT device cannot accept any commands now.
authorizeRefund			
	E_ILLEGAL	Error	Invalid timeout parameter was specified, or **CapAuthorizeRefund** is false.
	E_TIMEOUT	Warning	No response was received from CAT during the specified *timeout* time in milliseconds.
	E_BUSY	Information	The CAT device cannot accept any commands now.
authorizeSales			
	E_ILLEGAL	Error	Invalid *timeout* parameter was specified.
	E_TIMEOUT	Warning	No response was received from CAT during the specified *timeout* time in milliseconds.
	E_BUSY	Information	The CAT device cannot accept any commands now.

authorizeVoid			
	E_ILLEGAL	Error	Invalid *timeout* parameter was specified, or **CapAuthorizeVoid** is false.
	E_TIMEOUT	Warning	No response was received from CAT during the specified *timeout* time in milliseconds.
	E_BUSY	Information	The CAT device cannot accept any commands now.
authorizeVoidPreSales			
	E_ILLEGAL	Error	Invalid *timeout* parameter was specified, or **CapAuthorizeVoidPreSales** is false.
	E_TIMEOUT	Warning	No response was received from CAT during the specified *timeout* time in milliseconds.
	E_BUSY	Information	The CAT device cannot accept any commands now.
cashDeposit			
	E_ILLEGAL	Error	Invalid *timeout* parameter was specified, or **CapCashDeposit** is false.
	E_TIMEOUT	Warning	No response was received from CAT during the specified *timeout* time in milliseconds.
	E_BUSY	Information	The CAT device cannot accept any commands now.
cashCheck			
	E_ILLEGAL	Error	Invalid *timeout* parameter was specified, or **CapCheckCard** is false.
	E_TIMEOUT	Warning	No response was received from CAT during the specified *timeout* time in milliseconds.
	E_BUSY	Information	The CAT device cannot accept any commands now.
lockTerminal			
	E_ILLEGAL	Information	The Electronic Money Device does not have a security lock function.
	E_BUSY	Information	The CAT device cannot accept any commands now.
unlockTerminal			
	E_ILLEGAL	Information	The Electronic Money Device does not have a security lock function.
	E_BUSY	Information	The CAT device cannot accept any commands now.

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
CAT_LOGSTATUS_OK	Information	**DealingLog** is enough capacity
CAT_LOGSTATUS_NEARFULL	Warning	**DealingLog** is nearly full.
CAT_LOGSTATUS_FULL	Error	**DealingLog** is full

Check Scanner

Check Scanner Example

→beginInsertion

```
<?xml version="1.0" encoding="UTF-8"?>
<CheckScanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CheckScanner/
        CheckScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        CheckScanner/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>12341234</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="CheckScanner">1</SensorID>
    </ARTSHeader>
    <CheckScannerBody>
            <BeginInsertion>
                    <Timeout>10</Timeout>
            </BeginInsertion>
    </CheckScannerBody>
</CheckScanner>
```

→endInsertion

```
<?xml version="1.0" encoding="UTF-8"?>
<CheckScanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CheckScanner/
        CheckScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        CheckScanner/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>12341234</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="CheckScanner">Front Counter<
            /SensorID>
    </ARTSHeader>
    <CheckScannerBody>
            <EndInsertion/>
    </CheckScannerBody>
</CheckScanner>
```

← fire StatusUpdateEvent (check detected)

```
<?xml version="1.0" encoding="UTF-8"?>
<CheckScannerEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        CheckScannerEvents/ CheckScannerEventV1.14.1.xsd" xmlns="http://
        www.nrf-arts.org/UnifiedPOS/CheckScannerEvents/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <StatusUpdateEvent>
            <SequenceNumber>4</SequenceNumber>
            <EventDateTime>2001-12-17T09:30:47.0Z</EventDateTime>
            <SensorID UnifiedPOS="CheckScanner">1</SensorID>
            <Status>CHK_SUE_SCANCOMPLETE</Status>
    </StatusUpdateEvent>
</CheckScannerEvent>
```

→retrieveImage

```
<?xml version="1.0" encoding="UTF-8"?>
<CheckScanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CheckScanner/
        CheckScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        CheckScanner/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request">
            <MessageID>34563456</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="CheckScanner">1</SensorID>
    </ARTSHeader>
    <CheckScannerBody>
            <RetrieveImage>
                    <CropAreaID>2</CropAreaID>
            </RetrieveImage>
    </CheckScannerBody>
</CheckScanner>
```

←fire DataEvent

```
<?xml version="1.0" encoding="UTF-8"?>
<CheckScannerEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        CheckScannerEvents/ CheckScannerEventV1.14.1.xsd" xmlns="http://
        www.nrf-arts.org/UnifiedPOS/CheckScannerEvents/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <DataEvent Severity="Information" Mode="Production" Priority="-0">
            <SequenceNumber>4294967295</SequenceNumber>
            <EventDateTime TypeCode="Message">2001-12-17T09:30:47.0Z<
            /EventDateTime>
            <EventDescription>String</EventDescription>
            <SensorID UnifiedPOS="CheckScanner">1</SensorID>
            <Status>0</Status>
    </DataEvent>
</CheckScannerEvent>
```

→Retrieve Image

```
<?xml version="1.0" encoding="UTF-8"?>
<CheckScanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CheckScanner/
```

```
        CheckScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        CheckScanner/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Response" ActionCode="Read">
            <MessageID>12431234</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <Response>
                <RequestID>34563456</RequestID>
                <ResponderID>1</ResponderID>
            </Response>
            <SensorID UnifiedPOS="CheckScanner">1</SensorID>
    </ARTSHeader>
    <CheckScannerBody>
            <GetProperty><ImageData>\u005b\u0029\u003e\u001e\u0030
        \u0000\u001d\u0030\u0030\u0030\u0031\u001d\u0032\u001d
        \u0033\u001d\u0031\u0032\u0033</ImageData>
            </GetProperty>
    </CheckScannerBody>
</CheckScanner>
```

→beginRemoval

```
<?xml version="1.0" encoding="UTF-8"?>
<CheckScanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CheckScanner/
        CheckScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        CheckScanner/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>124379</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="CheckScanner">1</SensorID>
    </ARTSHeader>
    <CheckScannerBody>
            <BeginRemoval>
                <Timeout>10</Timeout>
            </BeginRemoval>
    </CheckScannerBody>
</CheckScanner>
```

→endRemoval

```
<?xml version="1.0" encoding="UTF-8"?>
<CheckScanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CheckScanner/
        CheckScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        CheckScanner/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>45675674567</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="CheckScanner">1</SensorID>
    </ARTSHeader>
    <CheckScannerBody>
            <EndRemoval/>
    </CheckScannerBody>
</CheckScanner>
```

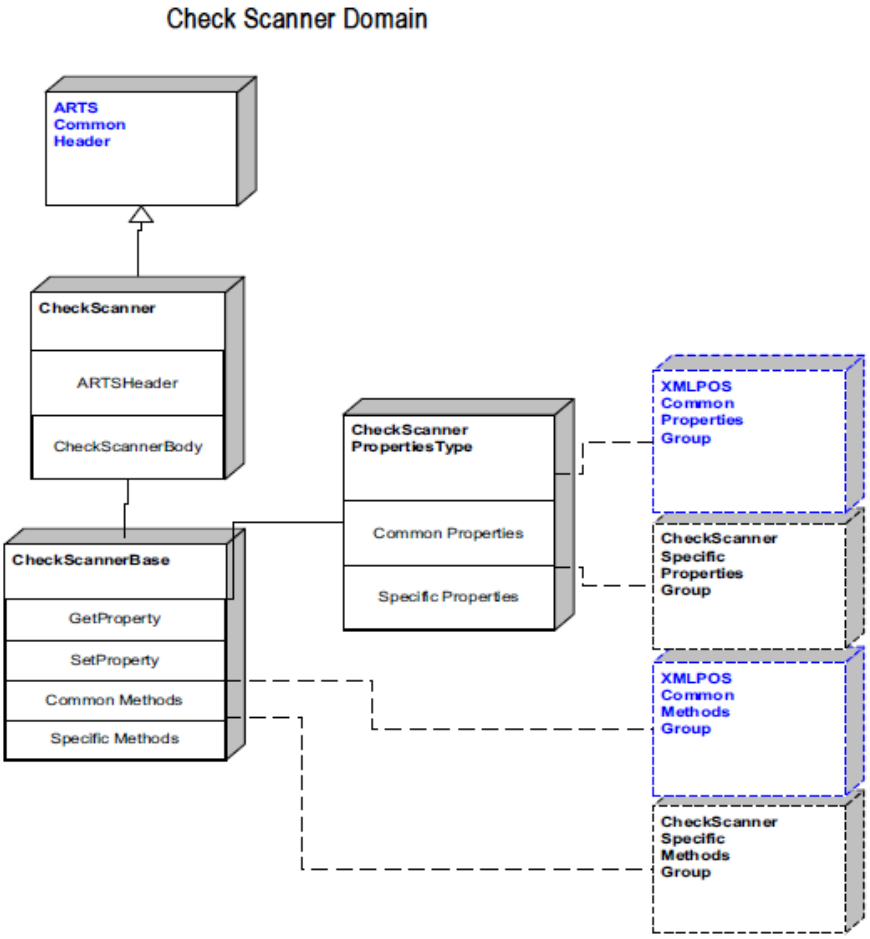


Figure 43: Check Scanner Domain View

Check Scanner Properties

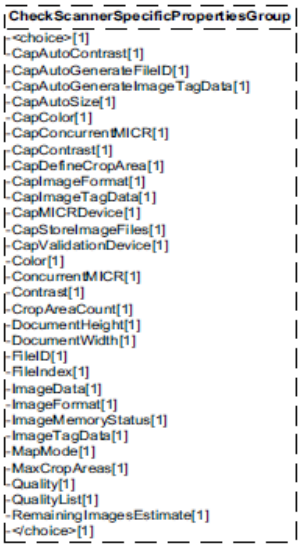


Figure 44: Check Scanner Properties Domain View

Check Scanner Methods

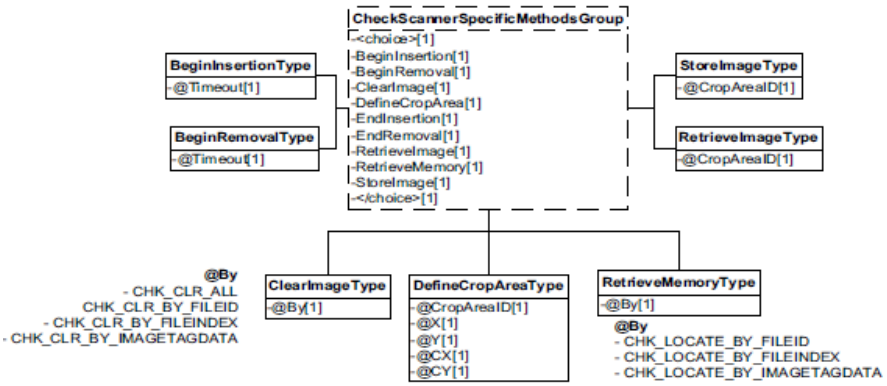


Figure 45: Check Scanner Methods Domain View

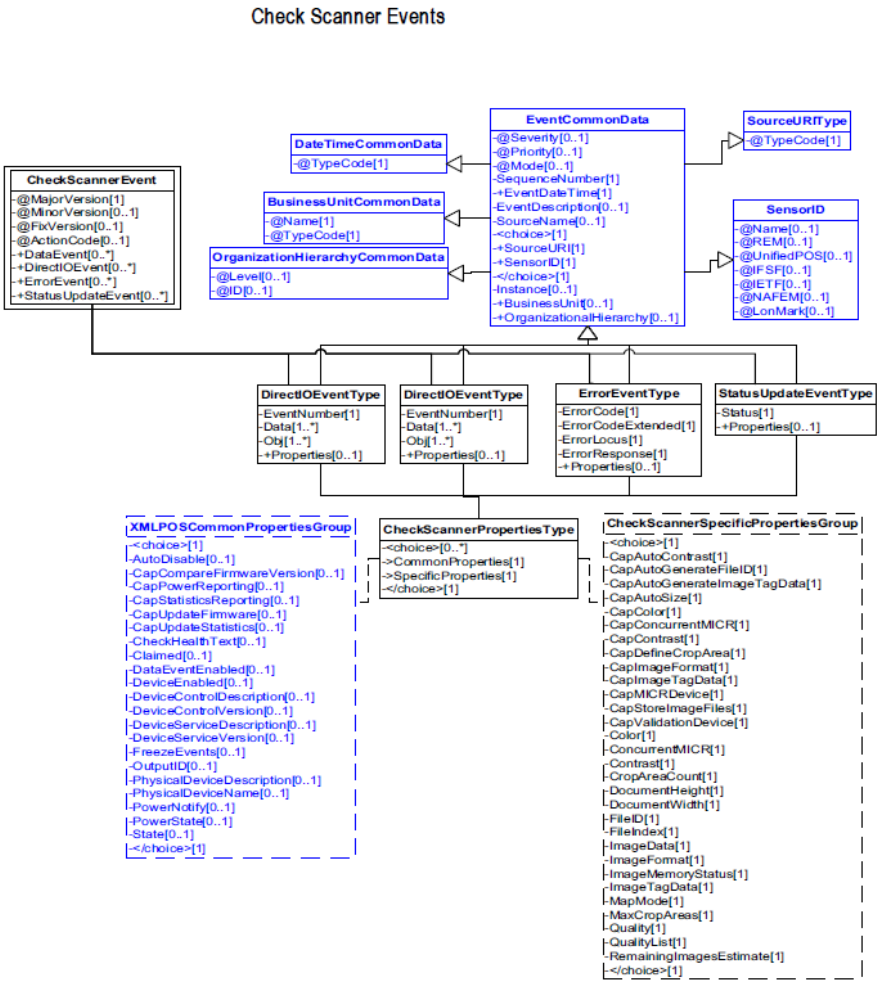


Figure 46: Check Scanner Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
beginInsertion			
	E_BUSY	Information	If the Check Scanner is a combination device, the peer device may be busy.
	E_ILLEGAL	Error	An invalid *timeout* parameter was specified.
	E_TIMEOUT	Warning	The specified time has elapsed without the check being properly inserted.
beginRemoval			
	E_BUSY	Information	If the Check Scanner is a combination device, the peer device may be busy.
	E_ILLEGAL	Error	An invalid *timeout* parameter was specified.
	E_TIMEOUT	Warning	The specified time has elapsed without the check being properly removed.
clearImage			
	E_ILLEGAL	Error	One of the following errors occurred: • Device does not support stored images • Device does not support clearing one image
	E_NOEXIST	Error	Image was not found.
endInsertion			
	E_ILLEGAL	Warning	The device is not in check insertion mode.
	ECHK_NOCHECK	Warning	The device was taken out of insertion mode without a check being inserted.
endRemoval			
	E_ILLEGAL	Warning	The device is not in check removal mode.
	ECHK_CHECK	Warning	The device was taken out of removal mode while a check is still present.
retrieveImage			
	E_ILLEGAL	Error	The following error has occurred: • Cropped area that is specified by *cropAreaID* parameter is invalid.

retrieveMemory			
	E_ILLEGAL	Error	One of the following errors occurred: • by parameter is invalid. • The image data file could not be located due to an invalid value stored in either the **FileID**, **FileIndex**, or **ImageTagData** properties that was being used with the *by* value.
storeImage			
	E_EXIST	Warning	Image already exists in the store location specified by the **FileIndex** property.
	E_ILLEGAL	Error	One of the following errors occurred: • Device does not support storing images • Cropped area that is specified by *cropAreaID* parameter is invalid.
	E_FAILURE	Error	Internal error storing image.
	ECHK_NOROOM	Error	There is no more room for the image in memory.

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
CHK_SUE_SCANCOMPLETE	Information	The process of scanning a document image has been successfully completed

Coin Acceptor

Coin Acceptor Example

→ SetRealTimeDataEvents to true

```
<?xml version="1.0" encoding="UTF-8"?>
<CoinAcceptor xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CoinAcceptor/
        CoinAcceptorV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        CoinAcceptor/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request" ActionCode="Update">
            <MessageID>asf1234</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="CoinAcceptor">1</SensorID>
    </ARTSHeader>
    <CoinAcceptorBody>
            <SetProperty>
                <RealTimeDataEnabled>true</RealTimeDataEnabled>
            </SetProperty>
    </CoinAcceptorBody>
</CoinAcceptor>
```

→:BeginDeposit()

```
<?xml version="1.0" encoding="UTF-8"?>
<CoinAcceptor xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CoinAcceptor/
        CoinAcceptorV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        CoinAcceptor/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>1234568</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="CoinAcceptor">1</SensorID>
    </ARTSHeader>
    <CoinAcceptorBody>
            <BeginDeposit/>
    </CoinAcceptorBody>
</CoinAcceptor>
```

Cash is accepted

← DataEvent is fired

```
<?xml version="1.0" encoding="UTF-8"?>
<CoinAcceptorEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        CoinAccpetorEvents/ CoinAcceptorEventV1.14.1.xsd" xmlns="http://
        www.nrf-arts.org/UnifiedPOS/CoinAccpetorEvents/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <DataEvent Severity="Information">
            <SequenceNumber>1</SequenceNumber>
            <EventDateTime>2001-12-17T09:30:47.0Z</EventDateTime>
            <SensorID UnifiedPOS="CoinAcceptor">1</SensorID>
            <Status>0</Status>
    </DataEvent>
</CoinAcceptorEvent>
```

→ EndDeposit()

```
<?xml version="1.0" encoding="UTF-8"?>
<CoinAcceptor xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CoinAcceptor/
        CoinAcceptorV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        CoinAcceptor/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>2134568</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="CoinAcceptor">1</SensorID>
    </ARTSHeader>
    <CoinAcceptorBody>
            <EndDeposit>
                    <Success>CACC_DEPOSIT_COMPLETE</Success>
            </EndDeposit>
    </CoinAcceptorBody>
</CoinAcceptor>
```

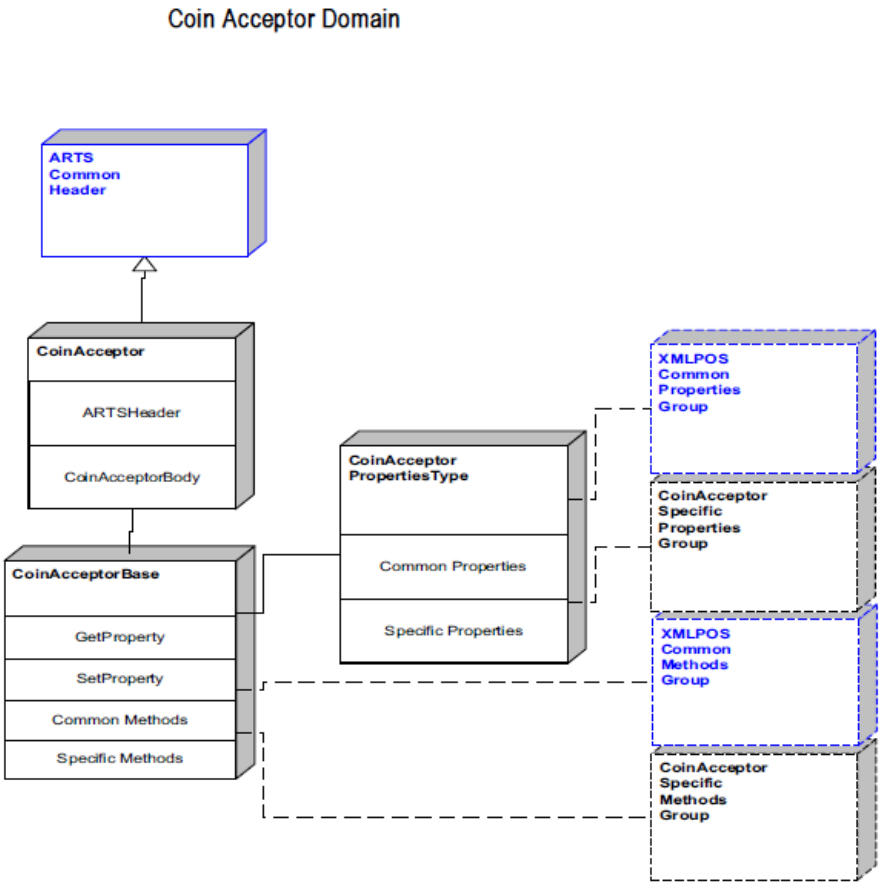


Figure 47: Coin Acceptor Domain View

Coin Acceptor Properties

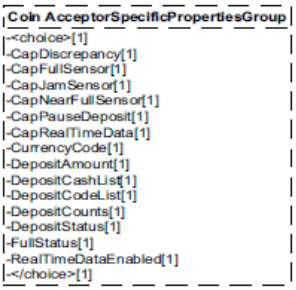


Figure 48: Coin Acceptor Properties Domain View

Coin Acceptor Methods

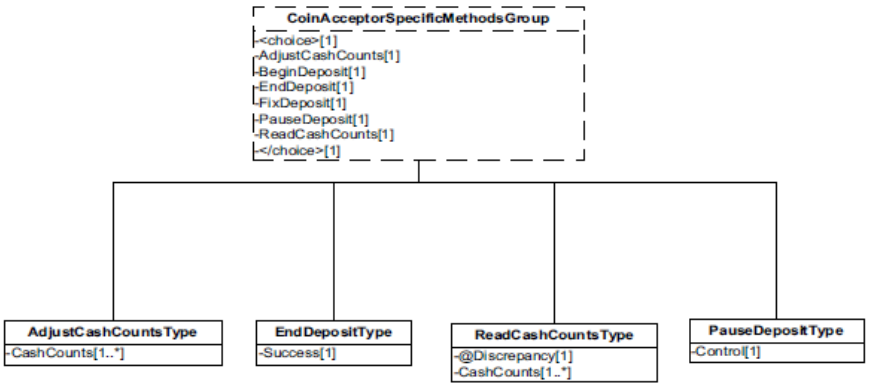


Figure 49: Coin Acceptor Methods Domain View

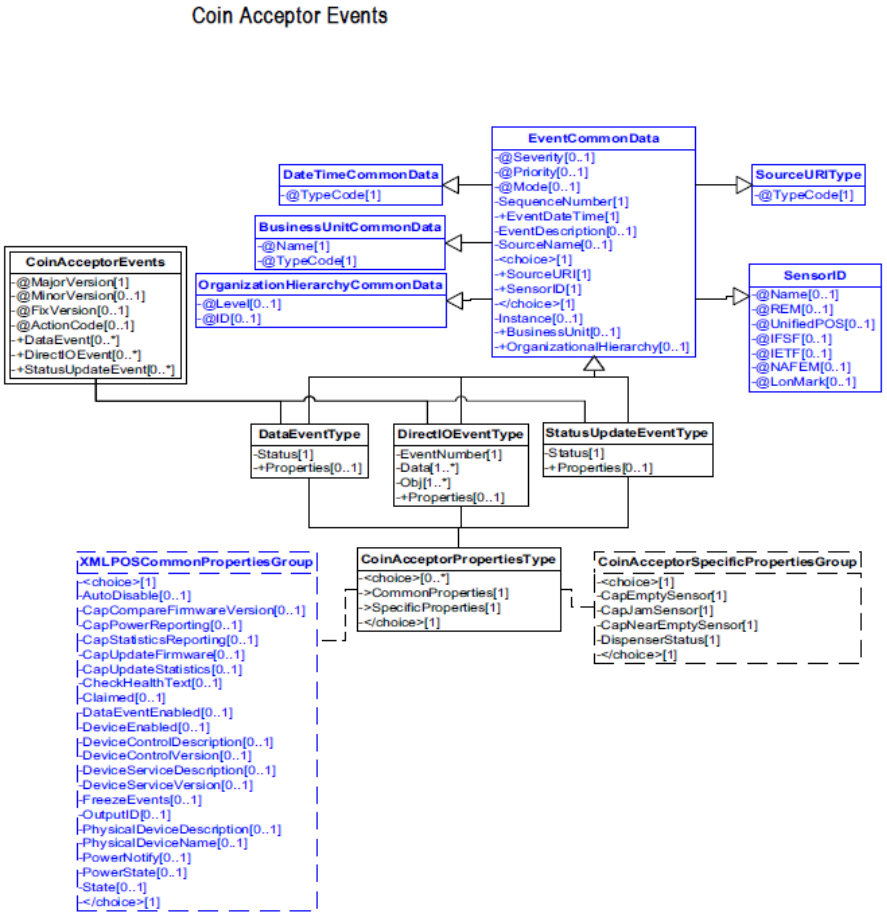


Figure 50: Coin Acceptor Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
beginDeposit			
	E_ILLEGAL	Error	The call sequence is not correct.
endDeposit			
	E_ILLEGAL	Error	One of the following errors occurred: • The call sequence is invalid. **beginDeposit** and **fixDeposit** must be called in sequence before calling this method.
fixDeposit			
	E_ILLEGAL	Error	One of the following errors occurred: • The call sequence is invalid. **beginDeposit** must be called before calling this method.
pauseDeposit			
	E_ILLEGAL	Error	One of the following errors occurred: • The call sequence is invalid. **beginDeposit** must be called before calling this method. • The deposit process is already paused and control is set to CACC_DEPOSIT_PAUSE, or the deposit process is not paused and control is set to CACC_DEPOSIT_RESTART.

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
CACC_STATUS_FULL	Error	Some cash slots are full.
CACC_STATUS_NEARFULL	Warning	Some cash slots are nearly full
CACC_STATUS_FULLOK	Information	No cash slots are either full or nearly full
CACC_STATUS_JAM	Error	A mechanical fault has occurred.
CACC_STATUS_JAMOK	Error	A mechanical fault has recovered

Coin Dispenser

Coin Dispenser Example

→DispenseChange("92")

```
<?xml version="1.0" encoding="UTF-8"?>
<CoinDispenser xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/CoinDispenser/
        CoinDispenserV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        CoinDispenser/"MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request">
            <MessageID>1</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="CoinDispenser">1</SensorID>
    </ARTSHeader>
    <CoinDispenserBody>
            <DispenseChange>
            <Amount>92</Amount>
            </DispenseChange>
    </CoinDispenserBody>
</CoinDispenser>
```

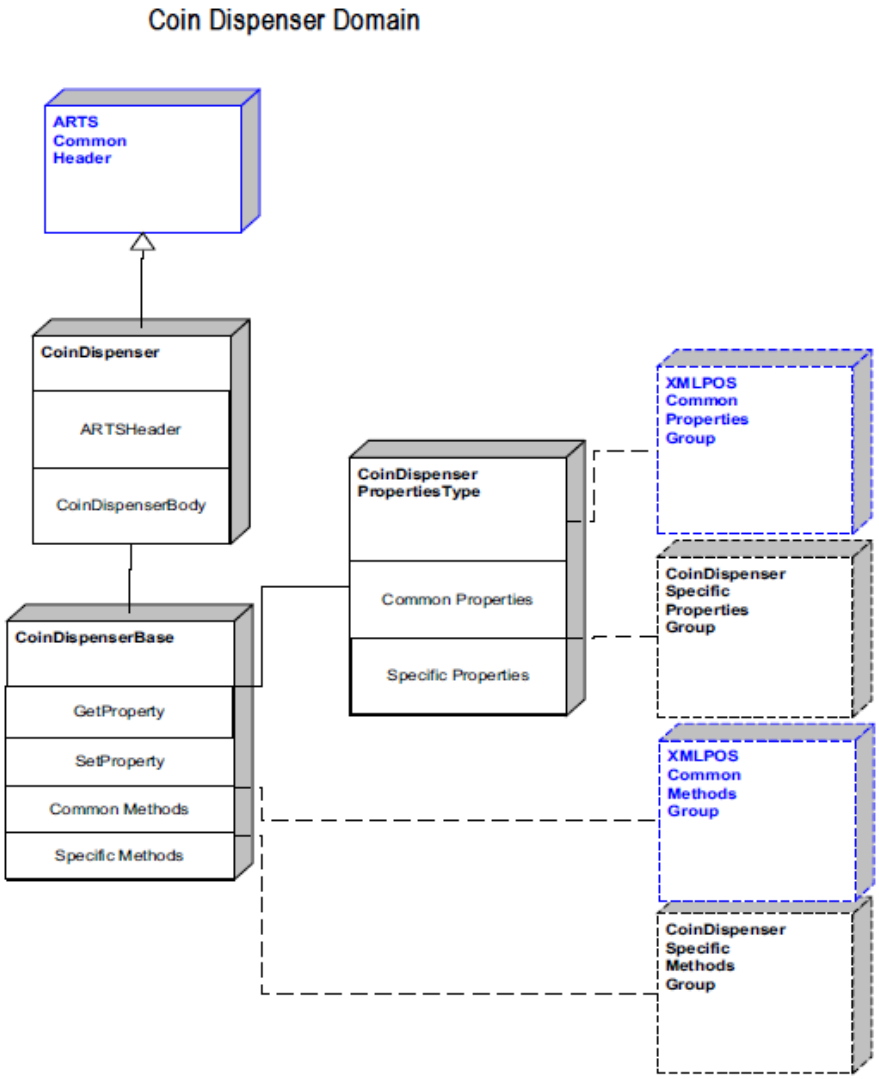


Figure 51: Coin Dispenser Domain View

Coin Dispenser Properties

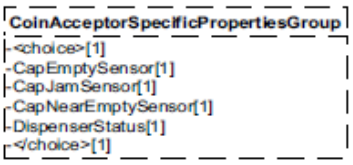


Figure 52: Coin Dispenser Properties Domain View

Coin Dispenser Methods

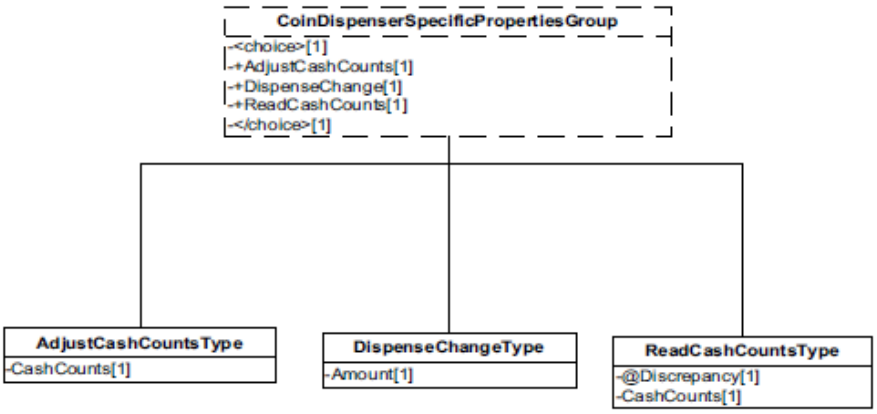


Figure 53: Coin Dispenser Methods Domain View

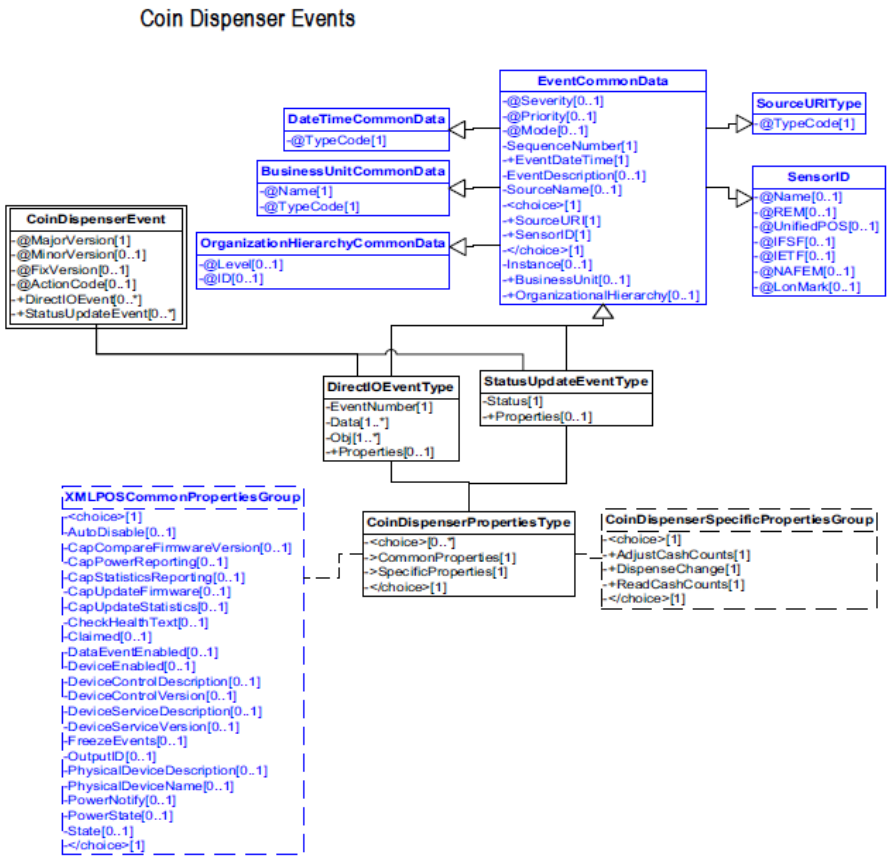


Figure 54: Coin Dispenser Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
dispenseChange			
	E_ILLEGAL	Error	An *amount* parameter value of zero was specified, or the *amount* parameter contained a negative value or a value greater than the device can dispense.

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
COIN_STATUS_OK	Information	Ready to dispense coinage. This value is also set when the dispenser is unable to detect an error condition
COIN_STATUS_EMPTY	Error	Cannot dispense coinage because the dispenser is empty.
COIN_STATUS_NEAREMPTY	Warning	Can still dispense coinage, but the dispenser is nearly empty.
COIN_STATUS_JAM	Error	A mechanical fault has occurred.

Electronic Journal

Electronic Journal Example

→queryContent("data.bin", 1, 2)

```
<?xml version="1.0" encoding="UTF-8"?>
<ElectronicJournal xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        ElectronicJournal/ ElectronicJournalV1.14.1.xsd" xmlns="http://www.nrf-
        arts.org/UnifiedPOS/ElectronicJournal/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>12341234</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="ElectronicJournal">EJ1</SensorID>
    </ARTSHeader>
    <ElectronicJournalBody>
            <QueryContent>
            <FileName>data.bin</FileName>
            <FromMarker>1</FromMarker>
            <ToMarker>2</ToMarker>
            </QueryContent>
    </ElectronicJournalBody>
</ElectronicJournal>
```

← DataEvent fired

```
<?xml version="1.0" encoding="UTF-8"?>
<ElectronicJournalEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
        instance" xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        ElectronicJournalEvents/ ElectronicJournalEventV1.14.1.xsd"
        xmlns="http://www.nrf-arts.org/UnifiedPOS/ElectronicJournalEvents/"
        MajorVersion="1" MinorVersion="14" FixVersion="1">
    <DataEvent Severity="Information">
    <SequenceNumber>4294967295</SequenceNumber>
    <EventDateTime>2001-12-17T09:30:47.0Z</EventDateTime>
    <SensorID UnifiedPOS="ElectronicJournal">EJ1</SensorID>
    <Status>0</Status>
    </DataEvent>
</ElectronicJournalEvent>
```

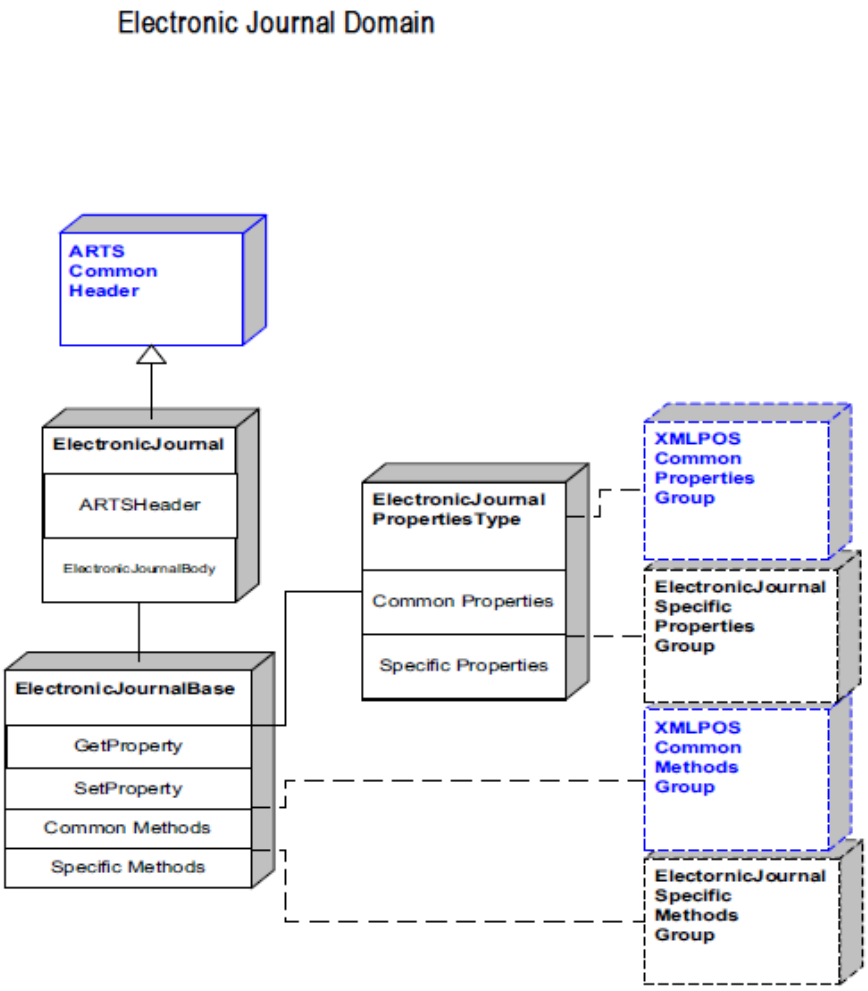


Figure 55: Electronic Journal Domain View

Electronic Journal Properties

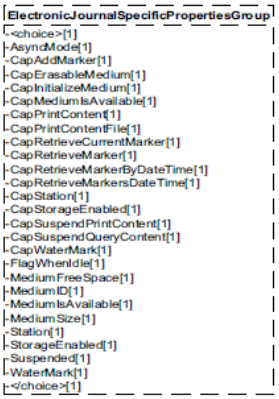


Figure 56: Electronic Journal Properties Domain View

Electronic Journal Methods

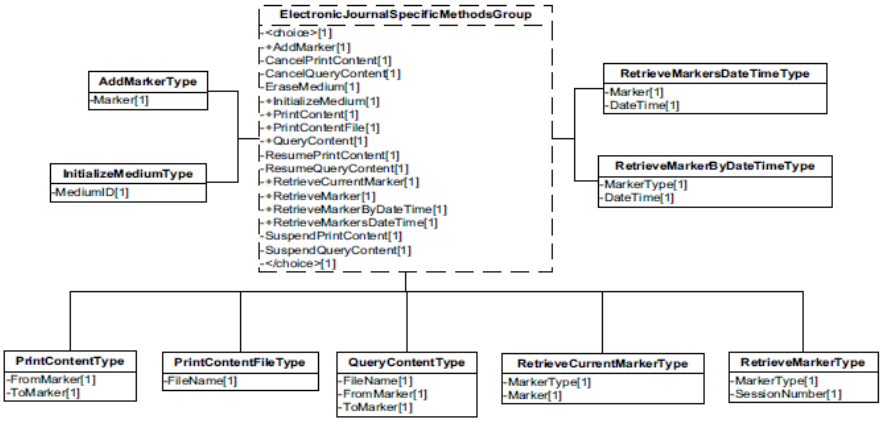


Figure 57: Electronic Journal Method Domain View

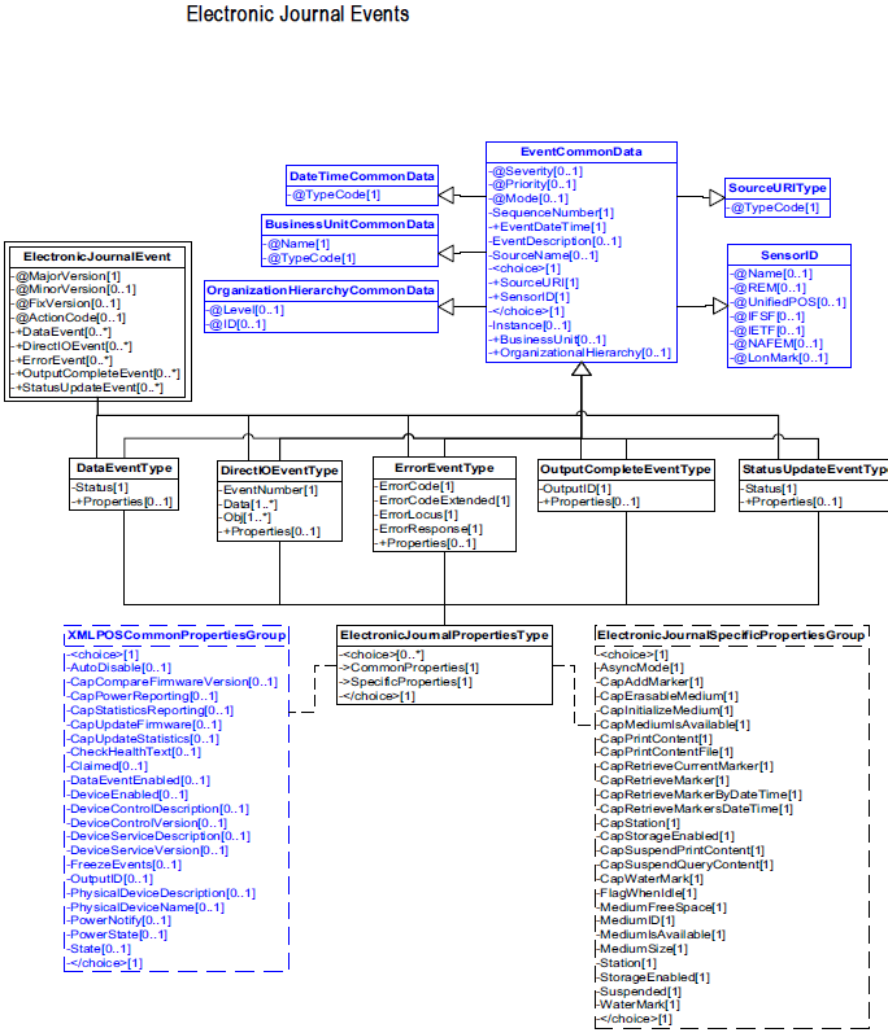


Figure 58: Electronic Journal Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
addMarker			
	E_ILLEGAL	Error	Characters that cannot be used as marker are included, or the character string is too long to be used as the marker.
	E_BUSY	Warning	Request cannot be performed while output is in progress. (This includes when the POSPrinter or FiscalPrinter is busy printing.)
	EEJ_EXISTING	Error	The marker name is already specified in current medium.
	EEJ_MEDIUM_-FULL	Error	There is not enough free space to add a marker in current medium.
eraseMedium			
	E_FAILURE	Error	Failed to erase data.
initializeMedium			
	E_BUSY	Warning	Cannot perform while output is in progress. (This includes when the POSPrinter or FiscalPrinter is busy printing.)
printContentFile			
	E_BUSY	Warning	Cannot perform while output is in progress. (This includes when the POSPrinter or FiscalPrinter is busy printing.)
	E_ILLEGAL	Error	fileName contains invalid characters.
	E_EXISTS	Error	The file defined in **fileName** already exists.
retrieveCurrentMarker			
	E_ILLEGAL	Error	The parameter **markerType** contains an invalid value.
	E_NOEXIST	Error	A marker does not exist for the specified marker type.
retrieveMarker			
	E_ILLEGAL	Error	One of the parameters is invalid. Either the value in **markerType** does not exist.

	E_NOEXIST	Error	A marker does not exist for the specified parameter values.
retrieveMarkerByDateTime			
	E_ILLEGAL	Error	One of the parameters is invalid. The value in **markerType** does not exist, **dateTime** is invalid, or the **markerNumber** does not exist for the specified time period.
	E_NOEXIST	Error	A marker does not exist for the specified time period.
	EEJ_MULTI-PLE_MARKER	Error	More than one marker exists for the specified time period.

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
EJ_SUE_MEDIUM_NEAR_FULL	Warning	The medium is nearly full (i.e., its free space is low
EJ_SUE_MEDIUM_FULL	Error	Storage medium is full.
EJ_SUE_MEDIUM_REMOVED	Information	Medium was removed from the device.
EJ_SUE_MEDIUM_INSERTED	Information	Medium was inserted into the device.
EJ_SUE_SUSPENDED	Warning	Data printing or transfer was suspended

Electronic Value Reader / Writer

Electronic Value Reader / Writer Example

→beginDetection

```
<?xml version="1.0" encoding="UTF-8"?>
<ElectronicValueRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        ElectronicValueRW/ ElectronicValueRWV1.14.1.xsd" xmlns="http://
        www.nrf-arts.org/UnifiedPOS/ElectronicValueRW/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>1</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="ElectronicValueReaderWriter">EVR1
            </SensorID>
    </ARTSHeader>
    <ElectronicValueRWBody>
            <BeginDetection Type="EVRW_BD_ANY">
                    <Timeout>30000</Timeout>
            </BeginDetection>
    </ElectronicValueRWBody>
</ElectronicValueRW>
```

→endDetection

```
<?xml version="1.0" encoding="UTF-8"?>
<ElectronicValueRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        ElectronicValueRW/ ElectronicValueRWV1.14.1.xsd" xmlns="http://
        www.nrf-arts.org/UnifiedPOS/ElectronicValueRW/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>2</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="ElectronicValueReaderWriter">EVR1
            </SensorID>
    </ARTSHeader>
    <ElectronicValueRWBody>
            <EndDetection/>
    </ElectronicValueRWBody>
</ElectronicValueRW>
```

→set DataEventEnabled to true

```
<?xml version="1.0" encoding="UTF-8"?>
<ElectronicValueRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        ElectronicValueRW/ ElectronicValueRWV1.14.1.xsd" xmlns="http://
        www.nrf-arts.org/UnifiedPOS/ElectronicValueRW/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request" ActionCode="Begin">
            <MessageID Timestamp="2001-12-17T09:30:47.0Z">1234
            </MessageID>
            <DateTime TypeCode="Message">2001-12-17T09:30:47.0Z
            </DateTime>
            <SensorID UnifiedPOS="ElectronicValueReaderWriter">EVR1
            </SensorID>
    </ARTSHeader>
    <ElectronicValueRWBody>
            <SetProperty>
                    <DataEventEnabled>true</DataEventEnabled>
            </SetProperty>
    </ElectronicValueRWBody>
</ElectronicValueRW>
```

→beginRemoval

```
<?xml version="1.0" encoding="UTF-8"?>
<ElectronicValueRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        ElectronicValueRW/ ElectronicValueRWV1.14.1.xsd" xmlns="http://
        www.nrf-arts.org/UnifiedPOS/ElectronicValueRW/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>6</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="ElectronicValueReaderWriter">EVR1
            </SensorID>
    </ARTSHeader>
    <ElectronicValueRWBody>
            <BeginRemoval>
                    <Timeout>30000</Timeout>
            </BeginRemoval>
    </ElectronicValueRWBody>
</ElectronicValueRW>
```

→endRemoval

```
<<?xml version="1.0" encoding="UTF-8"?>
<ElectronicValueRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        ElectronicValueRW/ ElectronicValueRWV1.14.1.xsd" xmlns="http://
        www.nrf-arts.org/UnifiedPOS/ElectronicValueRW/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>7</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="ElectronicValueReaderWriter">EVR1
             </SensorID>
    </ARTSHeader>
    <ElectronicValueRWBody>
            <EndRemoval/>
    </ElectronicValueRWBody>
</ElectronicValueRW>
```

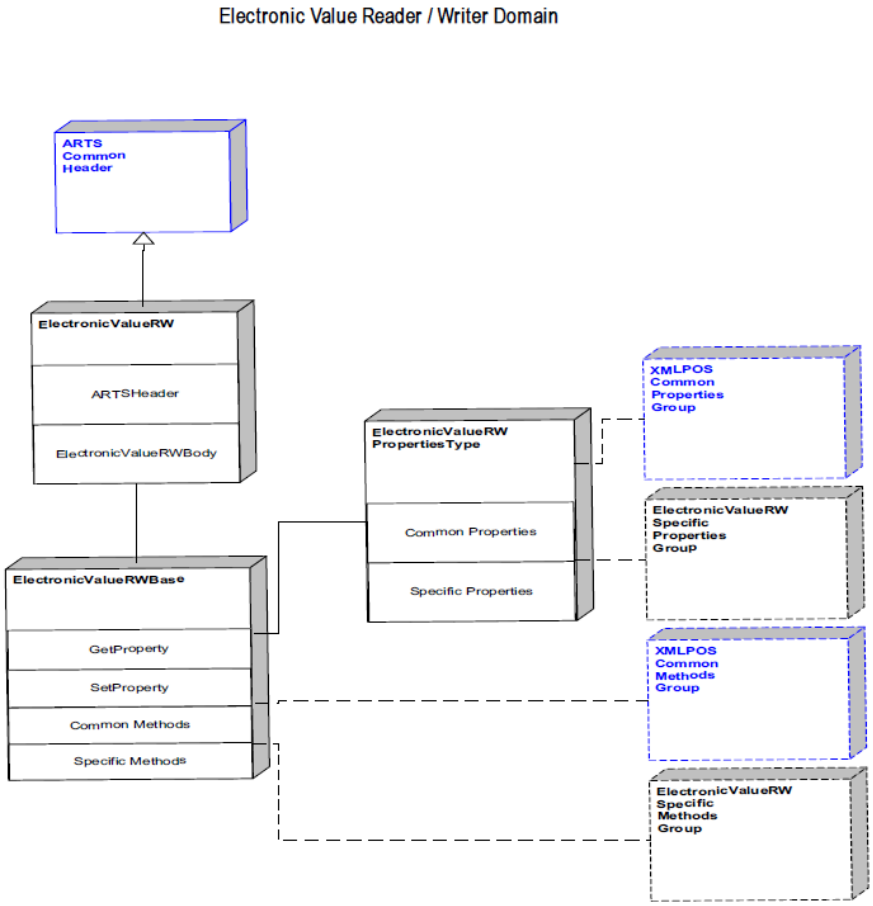


Figure 59: Electronic Value Reader / Writer Domain View

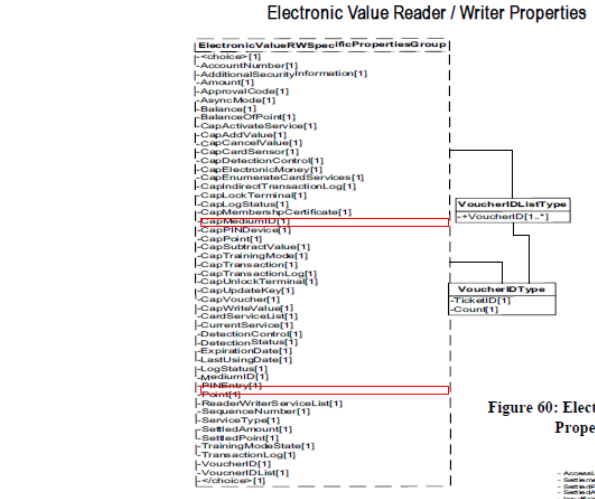


Figure 60: Electronic Value Reader / Writer Properties Domain View

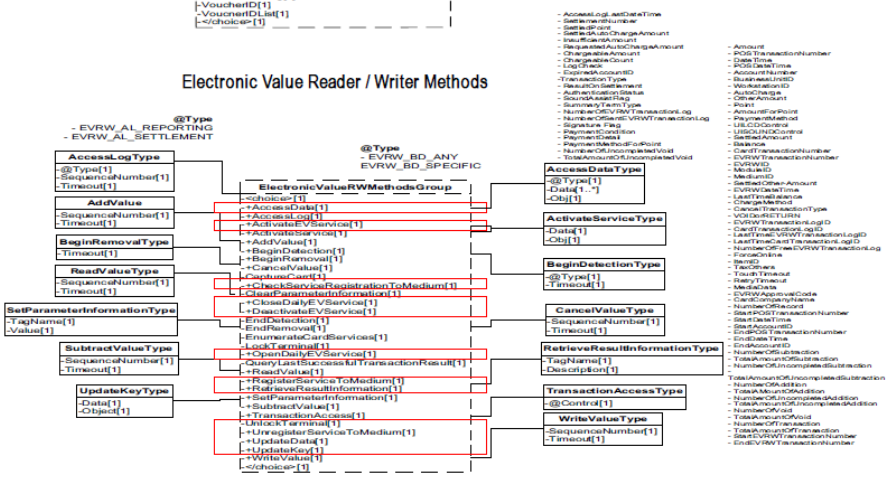


Figure 61: Electronic Value Reader / Writer Methods Domain View

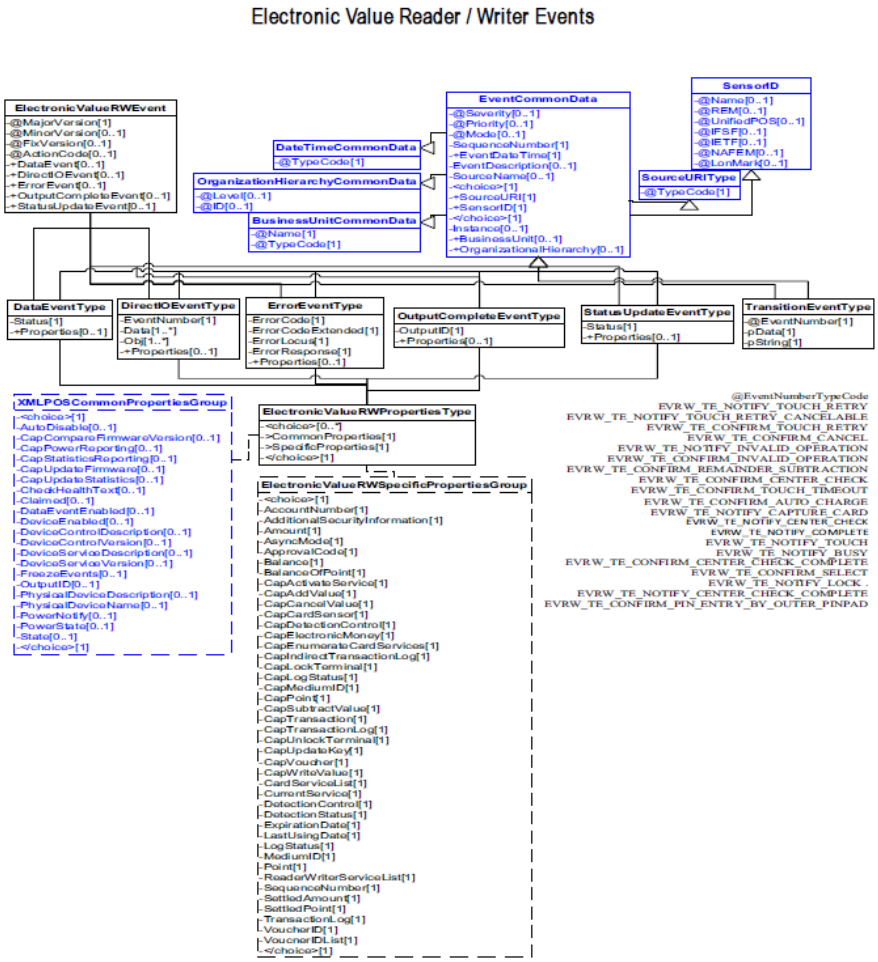


Figure 62: Electronic Value Reader / Writer Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning

Hard Totals

Hard Totals Example

→ write(1, data, 100, 256)

```
<?xml version="1.0" encoding="UTF-8"?>
<HardTotals xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/HardTotals/
        HardTotalsV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        HardTotals/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>1234235423452345</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="HardTotals">HT1</SensorID>
    </ARTSHeader>
    <HardTotalsBody>
            <Write>
                    <HTotalsFile>1</HTotalsFile>
                    <Data>\u005b\u0029\u003e\u001e\u0030\u0000\u001d
                            \u0030\u0030\u0030\u0031\u001d\u0032
                            \u001d\u0033\u001d\u0031\u0032\u0033
                    </Data>
                    <Offset>100</Offset>
                    <Count>256</Count>
            </Write>
    </HardTotalsBody>
</HardTotals>
```

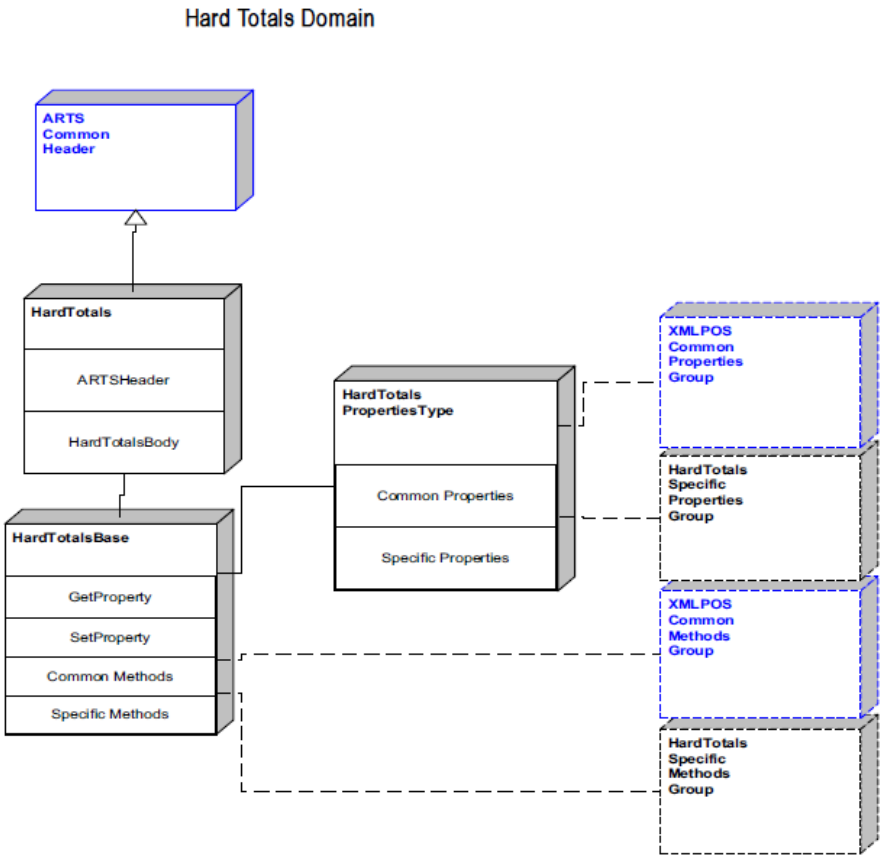


Figure 71: Hard Totals Domain View

Hard Totals Properties

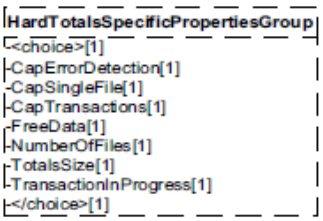


Figure 72: Hard Totals Properties Domain View

Hard Totals Methods

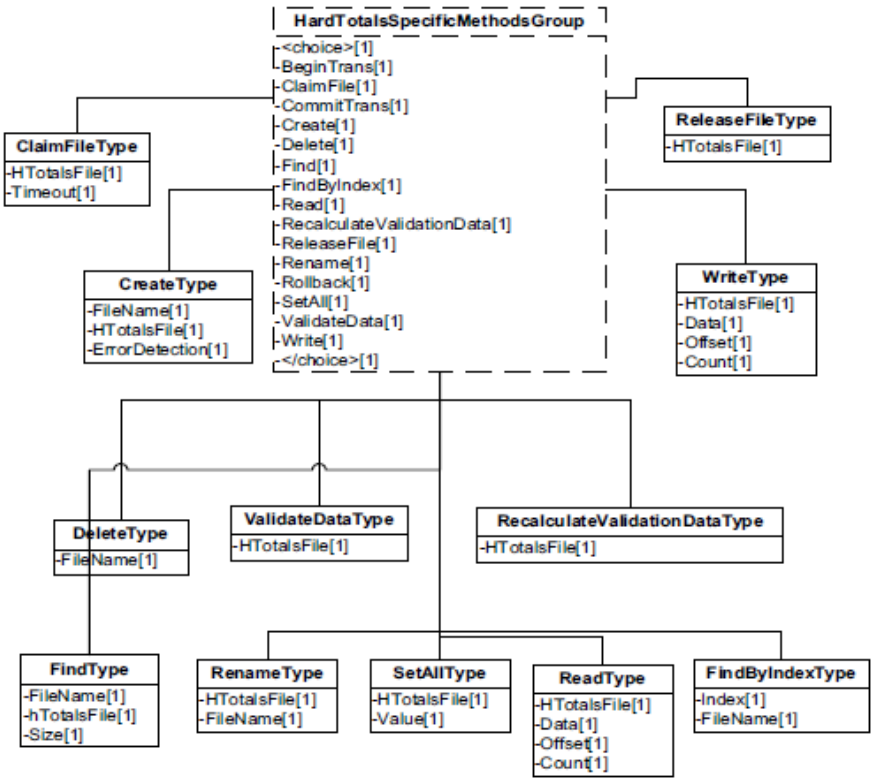


Figure 73: Hard Totals Methods Domain View

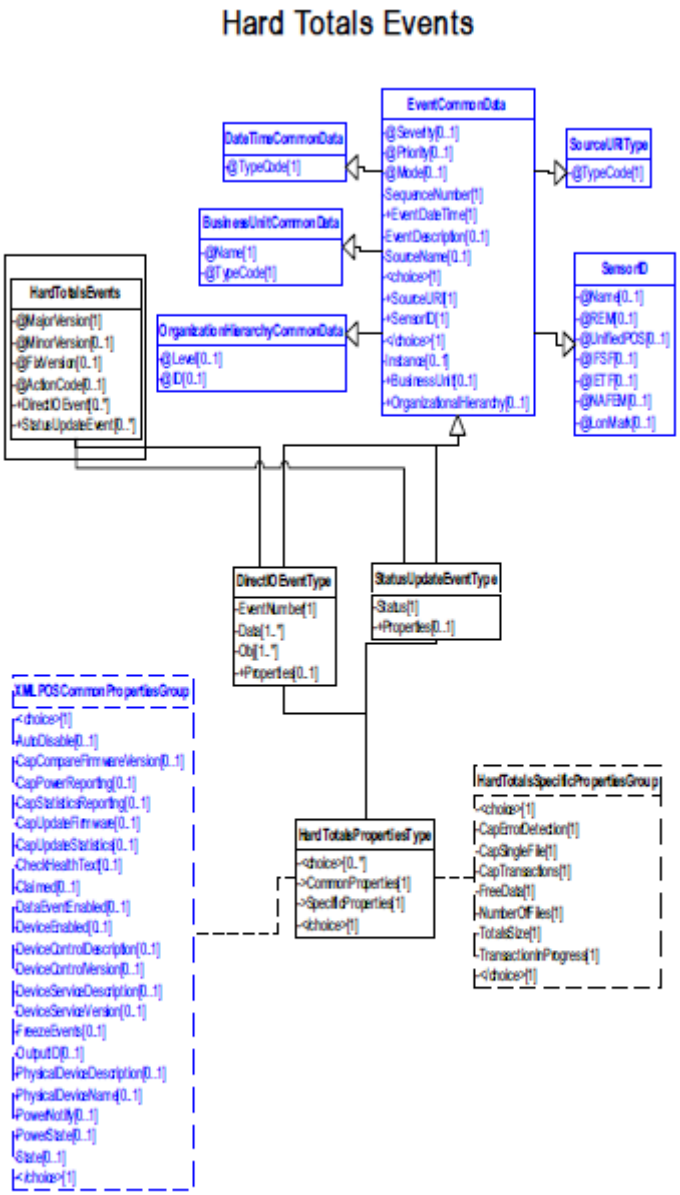


Figure 74: Hard Totals Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
beginTrans			
	E_ILLEGAL	Error	Transactions are not supported by this device.
claim			
	E_ILLEGAL	Error	An invalid timeout parameter was specified.
	E_TIMEOUT	Warning	Another application has exclusive access to the device or one or more of its files and did not relinquish control before timeout milliseconds expired.
claimFile			
	E_ILLEGAL	Error	The handle is invalid, or an invalid timeout parameter was specified.
	E_TIMEOUT	Warning	The timeout value expired before another application released exclusive access of either the requested totals file or the entire totals area.
commitTrans			
	E_ILLEGAL	Error	Transactions are not supported by this device, or no transaction is in progress.
create			
	E_CLAIMED	Warning	Cannot create because the entire totals file area is claimed by another application.
	E_ILLEGAL	Error	The *fileName* is too long or contains invalid characters
	E_EXISTS	Error	fileName already exists.
	ETOT_NOROOM	Error	There is insufficient room in the totals area to create the file.
delete			
	E_CLAIMED	Warning	Cannot delete because either the totals file or the entire totals area is claimed by another application.
	E_ILLEGAL	Error	The fileName is too long or contains invalid characters.
	E_NOEXIST	Error	fileName was not found.
find			
	E_CLAIMED	Warning	Cannot find because the entire totals file area is claimed by another application.
	E_ILLEGAL	Error	The fileName contains invalid characters.
	E_NOEXIST	Error	fileName was not found.
findByIndex			

	E_CLAIMED	Warning	Cannot find because the entire totals file area is claimed by another application.
	E_ILLEGAL	Error	The index is greater than the largest file index that is currently defined
read			
	E_CLAIMED	Warning	Cannot read because either the totals file or the entire totals area is claimed by another application.
	E_ILLEGAL	Error	The handle is invalid, part of the data range is outside the bounds of the totals file, or data array length is less than count
	ETOT_VALIDATION	Error	A validation error has occurred while reading data.
recalculateValidationData			
	E_CLAIMED	Warning	Cannot recalculate because either the totals file or the entire totals area is claimed by another application.
	E_ILLEGAL	Error	The handle is invalid, or advanced error detection is either not supported by the Service or by this file
release			
	E_ILLEGAL	Warning	The application does not have exclusive access to the device.
releaseFile			
	E_ILLEGAL	Error	The handle is invalid, or the specified file is not claimed by this application.
rename			
	E_CLAIMED	Error	Cannot rename because either the totals file or the entire totals area is claimed by another application.
	E_ILLEGAL	Error	The handle is invalid, the fileName contains invalid characters, or the **CapSingleFile** property is true.
	E_EXISTS	Error	fileName already exists.
rollback			
	E_ILLEGAL	Error	Transactions are not supported by this device, or no transaction is in progress.
setAll			
	E_CLAIMED	Warning	Cannot set because either the totals file or the entire totals area is claimed by another application.
	E_ILLEGAL	Error	The handle is invalid.
validateData			
	E_CLAIMED	Warning	Cannot validate because either the totals file or the entire totals area is claimed by another application.

	E_ILLEGAL	Error	The handle is invalid, or advanced error detection is either not supported by the Service or by this file.
write			
	E_CLAIMED	Warning	Cannot write because either the totals file or the entire totals area is claimed by another application.
	E_ILLEGAL	Error	The handle is invalid, or part of or all of the data range is outside the bounds of the totals file.
	ETOT_NOROOM	Error	Cannot write because a transaction is in progress, and there is not enough free space to prepare for the transaction commit.
	ETOT_VALIDATION	Error	A validation error has occurred while reading data.

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
N/A		

Image Scanner

Image Scanner Example

→ set ImageMode to IMG_STILL_ONLY

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageScanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/ImageScanner/
        ImageScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        ImageScanner/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish" ActionCode="Update">
            <MessageID>asdfasfsdf</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="ImageScanner">IS1</SensorID>
    </ARTSHeader>
    <ImageScannerBody>
            <SetProperty>
                    <ImageMode>IMG_STILL_ONLY</ImageMode>
            </SetProperty>
    </ImageScannerBody>
</ImageScanner>
```

→ set DeviceEnabled to true

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageScanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/ImageScanner/
        ImageScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        ImageScanner/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish" ActionCode="Update">
            <MessageID>sdhgdfg</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="ImageScanner">IS1</SensorID>
    </ARTSHeader>
    <ImageScannerBody>
            <SetProperty>
                    <DeviceEnabled>false</DeviceEnabled>
            </SetProperty>
    </ImageScannerBody>
</ImageScanner>
```

(acquire image)

← fire DataEvent

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageScannerEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        ImageScannerEvents/ ImageScannerEventV1.14.1.xsd" xmlns="http://
        www.nrf-arts.org/UnifiedPOS/ImageScannerEvents/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <DataEvent>
            <SequenceNumber>4294967295</SequenceNumber>
            <EventDateTime>2001-12-17T09:30:47.0Z</EventDateTime>
```

```
            <SensorID UnifiedPOS="ImageScanner">IS1</SensorID>
            <Status>0</Status>
        </DataEvent>
    </ImageScannerEvent>
```

Application services event

Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageScanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/ImageScanner/
        ImageScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        ImageScanner/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request" ActionCode="Read">
            <MessageID>1</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="ImageScanner">IS1</SensorID>
    </ARTSHeader>
    <ImageScannerBody>
            <GetProperty>
                    <FrameData xsi:nil="true"/>
                    <FrameType xsi:nil="true"/>
                    <ImageHeight xsi:nil="true"/>
                    <ImageType xsi:nil="true"/>
                    <ImageWidth xsi:nil="true"/>
            </GetProperty>
    </ImageScannerBody>
</ImageScanner>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageScanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/ImageScanner/
        ImageScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        ImageScanner/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Response" ActionCode="Read">
            <MessageID>1</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="ImageScanner">IS1</SensorID>
    </ARTSHeader>
    <ImageScannerBody>
            <GetProperty>
                    <FrameData>
        \u005b\u0029\u003e\u001e\u0030\u0000\u001d\u0030\u0030\u0030\u0031
        \u001d\u0032\u001d\u0033\u001d\u0031\u0032\u0033
                    </FrameData>
                    <FrameType>IMG_FRAME_STILL</FrameType>
                    <ImageHeight>8</ImageHeight>
                    <ImageType>IMG_TYP_GIF</ImageType>
                    <ImageWidth>19</ImageWidth>
            </GetProperty>
    </ImageScannerBody>
</ImageScanner>
```

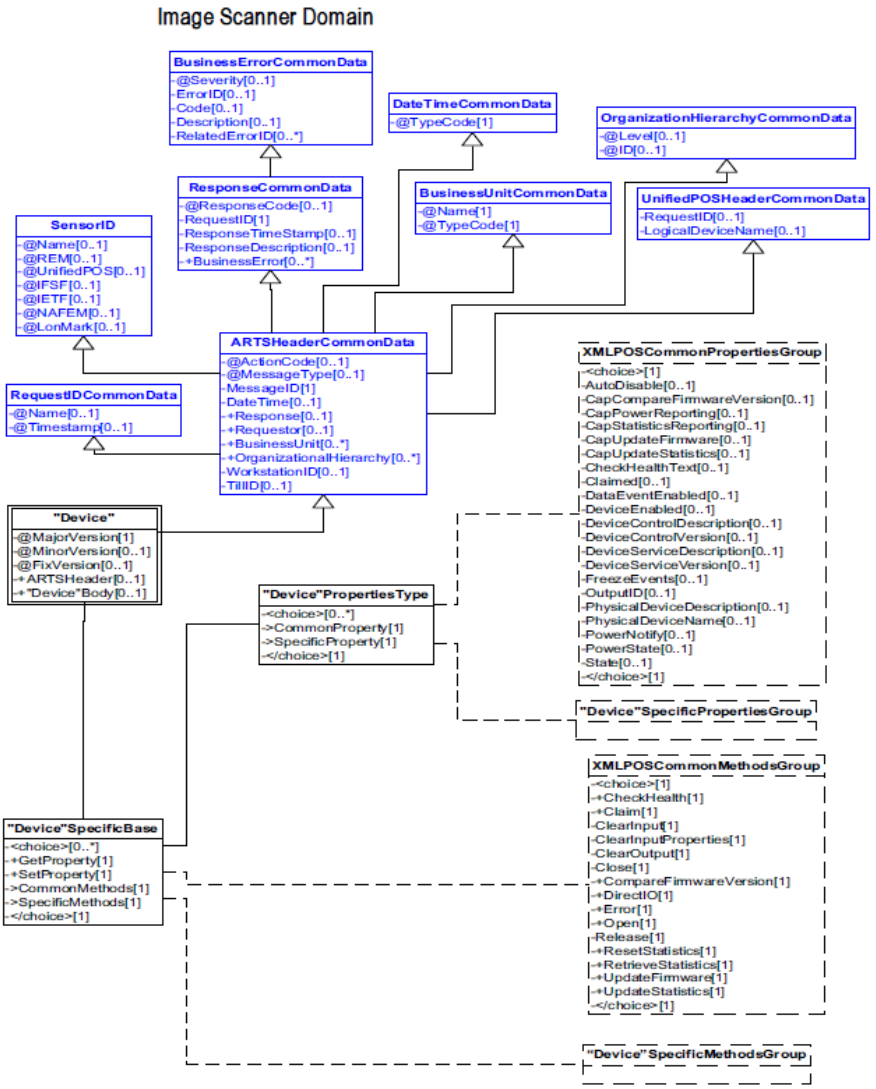


Figure 75: Image Scanner Domain View

Image Scanner Properties

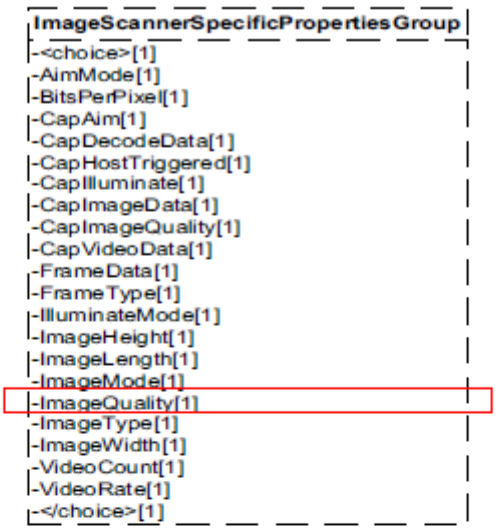


Figure 76: Image Scanner Properties Domain View

Image Scanner Methods

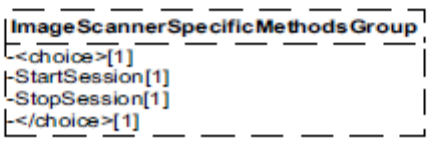


Figure 77: Image Scanner Methods Domain View

Image Scanner Events

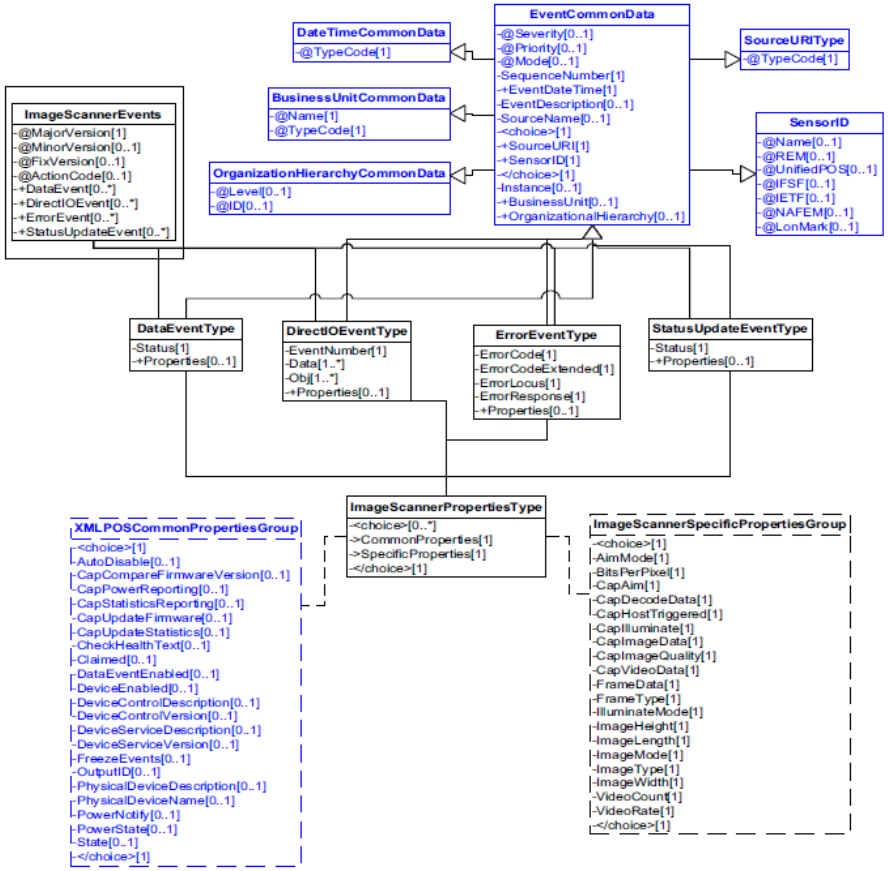


Figure 78: Image Scanner Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
startSession			
	E_ILLEGAL	Error	An attempt was made to call the **startSession** method when the **CapHostTriggered** property is false.
stopSession			
	E_ILLEGAL	Error	An attempt was made to call the **stopSession** method when the **CapHostTriggered** property is false.

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
N/A		

Keylock

Keylock Example

→ waitForKeylockChange(LOCK_KP_LOCK, 30000)

```
<?xml version="1.0" encoding="UTF-8"?>
<Keylock xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/Keylock/
        KeylockV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        Keylock/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>1241234</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="Keylock">1</SensorID>
    </ARTSHeader>
    <KeylockBody>
            <WaitForKeylockChange>
            <KeyPosition>LOCK_KP_LOCK</KeyPosition>
            <Timeout>30000</Timeout>
            </WaitForKeylockChange>
    </KeylockBody>
</Keylock>
```

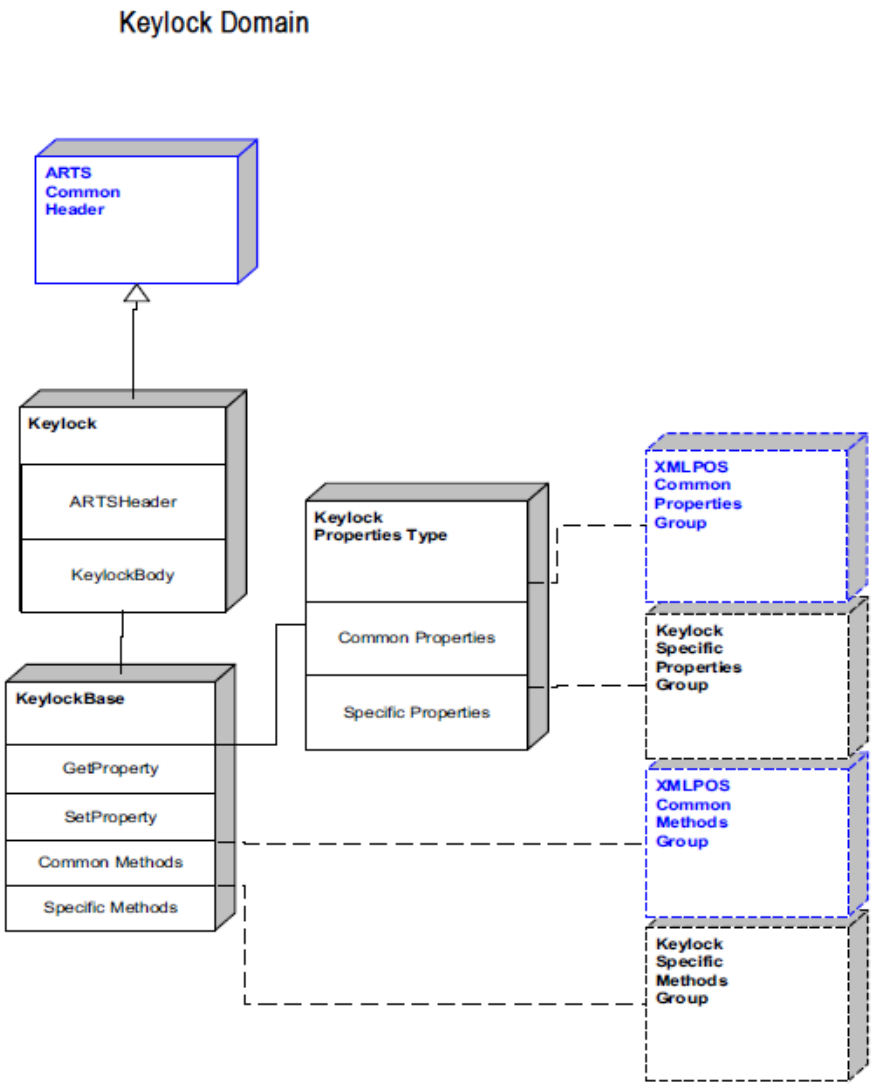


Figure 83: Keylock Domain View

Keylock Properties

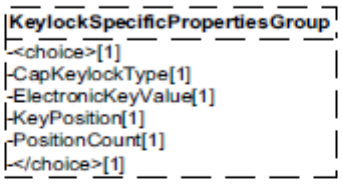


Figure 84: Keylock Properties Domain View

Keylock Methods

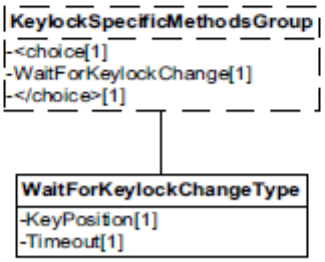


Figure 85: Keylock Methods Domain View

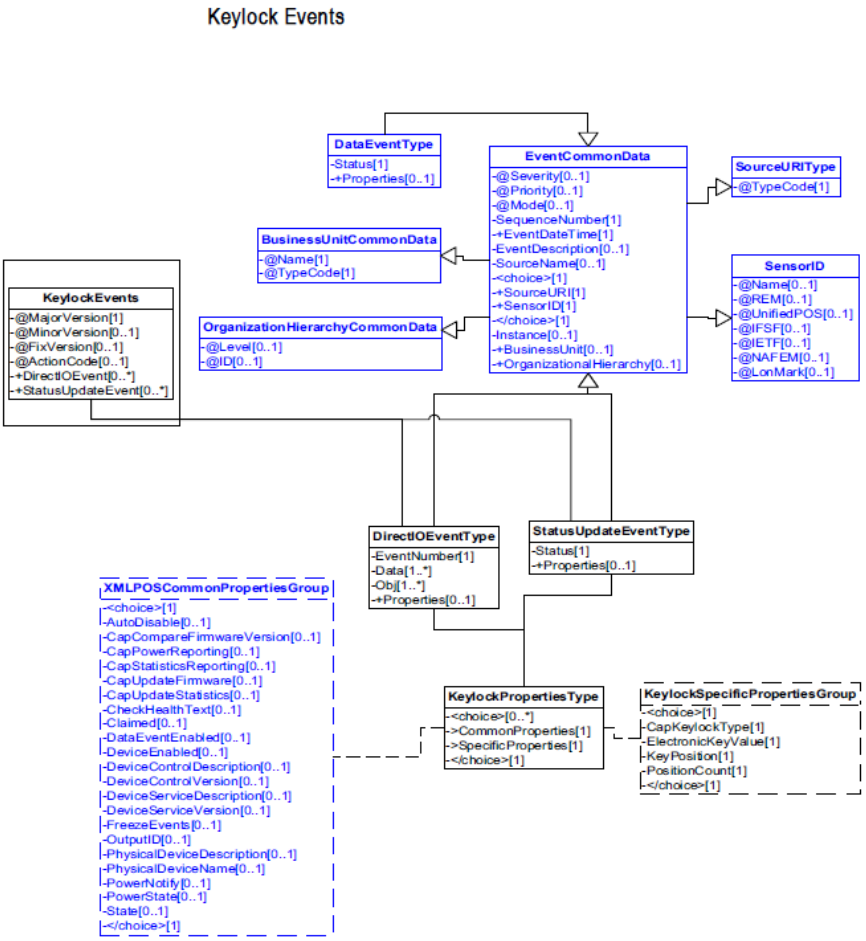


Figure 86: Keylock Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
waitForKeylockChange			
	E_ILLEGAL	Error	An invalid parameter value was specified.
	E_TIMEOUT	Warning	The timeout period expired before the requested keylock positioning occurred.

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
LOCK_KP_ELECTRONIC	Information	Electronic Keylock value.
LOCK_KP_LOCK	Information	Keylock is in the “locked” position.
LOCK_KP_NORM	Information	Keylock is in the “normal” position
LOCK_KP_SUPR	Information	Keylock is in the “supervisor” position.

Line Display

Line Display Example

→ displayText("Hello, World", DISP_DP_NORMAL)

```
<?xml version="1.0" encoding="UTF-8"?>
<LineDisplay xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/LineDisplay/
        LineDisplayV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        LineDisplay/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>asdf1234asfd</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="LineDisplay">1</SensorID>
    </ARTSHeader>
    <LineDisplayBody>
            <DisplayText>
                    <Data>Hello World</Data>
                    <Attribute>DISP_DT_NORMAL</Attribute>
            </DisplayText>
    </LineDisplayBody>
</LineDisplay>
```

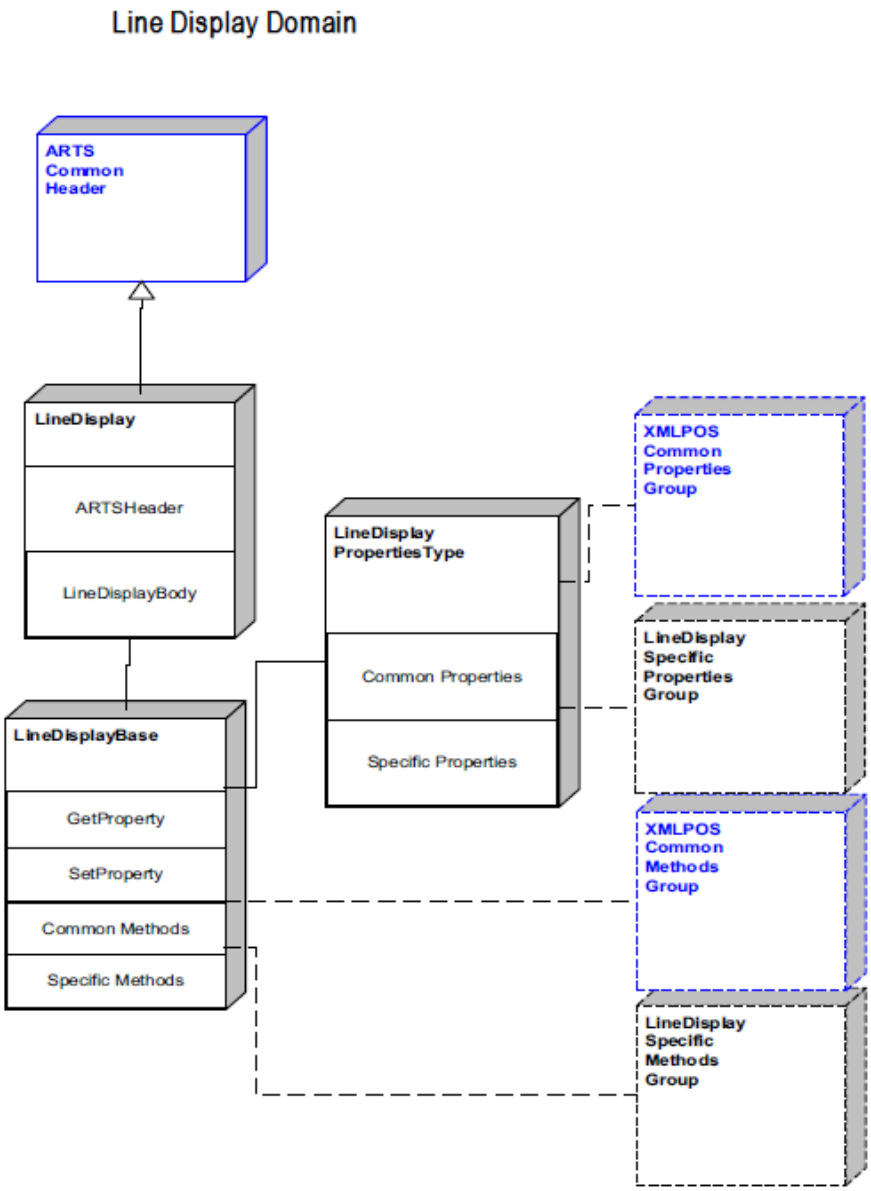


Figure 91: Line Display Domain View

Line Display Properties

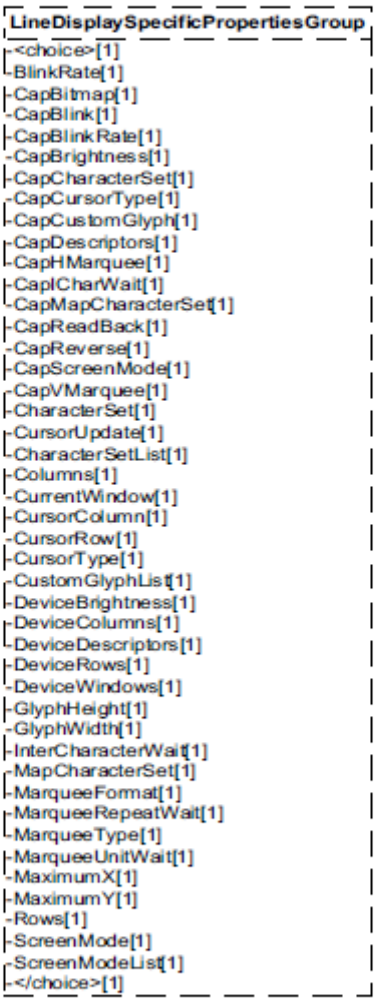


Figure 92: Line Display Properties Domain View

Line Display Methods

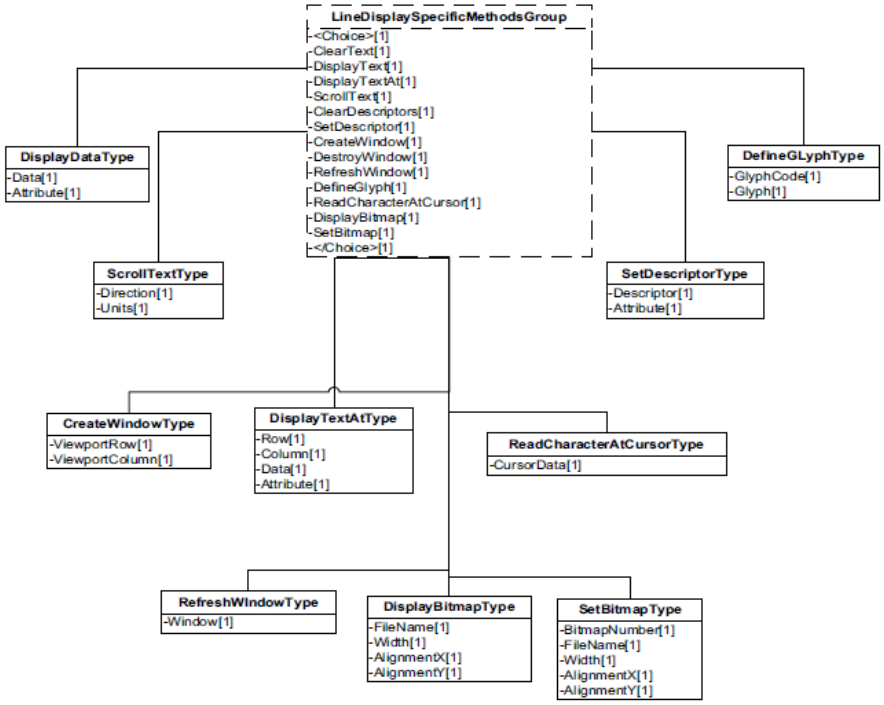


Figure 93: Line Display Methods Domain View

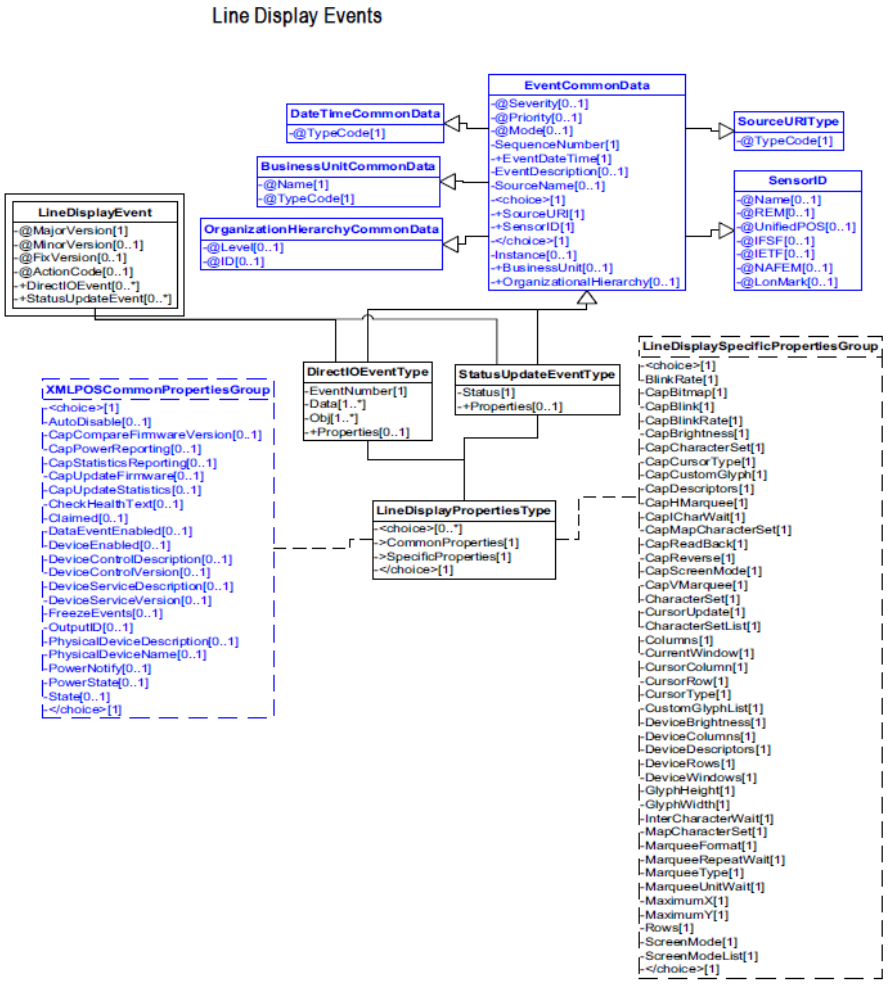


Figure 94: Line Display Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
clearDescriptors			
	E_ILLEGAL	Error	The device does not support descriptors.
clearText			
	E_ILLEGAL	Error	In Marquee On Mode
createWindow			
	E_ILLEGAL	Error	One or more parameters are out of their valid ranges, or all available windows are already in use.
defineGlyph			
	E_ILLEGAL	Error	**CapCustomGlyph** is false, or **glyphCode** is an unsupported character code for glyph definition.
destroyWindow			
	E_ILLEGAL	Error	The current window is 0. This window may not be destroyed.
displayBitmap			
	E_ILLEGAL	Error	One of the following errors occurred: • The LineDisplay does not support bitmap display. • The width parameter is invalid or too big. • The *alignmentX* / alignmentY parameter is invalid or too big. • The window is not in Immediate Mode. • The window size does not match its viewport size. • The bitmap is too large to display at the requested location.
	E_NOEXIST	Error	The fileName was not found.
	EDISP_TOO BIG	Error	The bitmap is either too wide to display without transformation, or it is too big to transform.
	EDISP_BAD FORMAT	Error	The specified file is either not a bitmap file or it is an unsupported format.
displayText			
	E_ILLEGAL	Error	attribute is illegal, or the display is in Marquee On Mode.
displayTextAt			
	E_ILLEGAL	Error	row or column are out or range, attribute is illegal, or in Marquee On Mode.

readCharacterAtCursor			
	E_ILLEGAL	Error	**CapReadBack** is DISP_CRB_NONE.
refreshWindow			
	E_ILLEGAL	Error	window is larger than DeviceWindows or has not been created, or in Marquee On Mode.
scrollText			
	E_ILLEGAL	Error	direction is illegal, or in Teletype Mode or Marquee Mode.
setBitmap			
	E_ILLEGAL	Error	One of the following errors occurred: • The bitmapNumber parameter is invalid. • The LineDisplay does not support bitmap display. • The width parameter is invalid or too big. • The alignmentX or alignmentY parameter is invalid or too big.
	E_NOEXIST	Error	The fileName was not found.
	EDISP_TOO BIG	Error	The bitmap is either too wide to display without transformation, or it is too big to transform.
	EDISP_BAD FORMAT	Error	The specified file is either not a bitmap file or it is an unsupported format.
setDescriptor			
	E_ILLEGAL	Error	The device does not support descriptors, or one of the parameters contained an illegal value.

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
N/A		

MSR

MSR Example

←fire DataEvent

```
<?xml version="1.0" encoding="UTF-8"?>
<MSREvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/MSREvents/
        MSREventV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        MSREvents/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <DataEvent>
            <SequenceNumber>4294967295</SequenceNumber>
            <EventDateTime>2001-12-17T09:30:47.0Z</EventDateTime>
            <SensorID UnifiedPOS="MagneticStripeReader">MSR1
            </SensorID>
            <Status>0</Status>
    </DataEvent>
</MSREvent>
```

→get Track1Data

```
<?xml version="1.0" encoding="UTF-8"?>
<MSR xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/MSR/
        MSRV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/MSR/"
        MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request" ActionCode="Read">
            <MessageID>123123123</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="MagneticStripeReader">MSR1
            </SensorID>
    </ARTSHeader>
    <MSRBody>
            <GetProperty>

        <Track1Data>\u005b\u0029\u003e\u001e\u0030\u0000\u001d\u0030
        \u0030\u0030\u0031\u001d\u0032\u001d\u0033\u001d\u0031\u0032\u0033
        </Track1Data>

            </GetProperty>
    </MSRBody>
</MSR>
```

→return Track1Data

```
<?xml version="1.0" encoding="UTF-8"?>
<MSR xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/MSR/
        MSRV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/MSR/"
        MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Response" ActionCode="Read">
            <MessageID>12344444</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <Response>
                    <!-- this is the ID of the original request message -->
                    <RequestID>123123123</RequestID>
                    <ResponderID>MSR Service</ResponderID>
            </Response>
            <SensorID UnifiedPOS="MagneticStripeReader">MSR1
            </SensorID>
    </ARTSHeader>
    <MSRBody>
            <GetProperty>

        <Track1Data>\u005b\u0029\u003e\u001e\u0030\u0000\u001d\u0030
        \u0030\u0030\u0031\u001d\u0032\u001d\u0033\u001d\u0031\u0032\u0033
        </Track1Data>

            </GetProperty>
    </MSRBody>
</MSR>
```

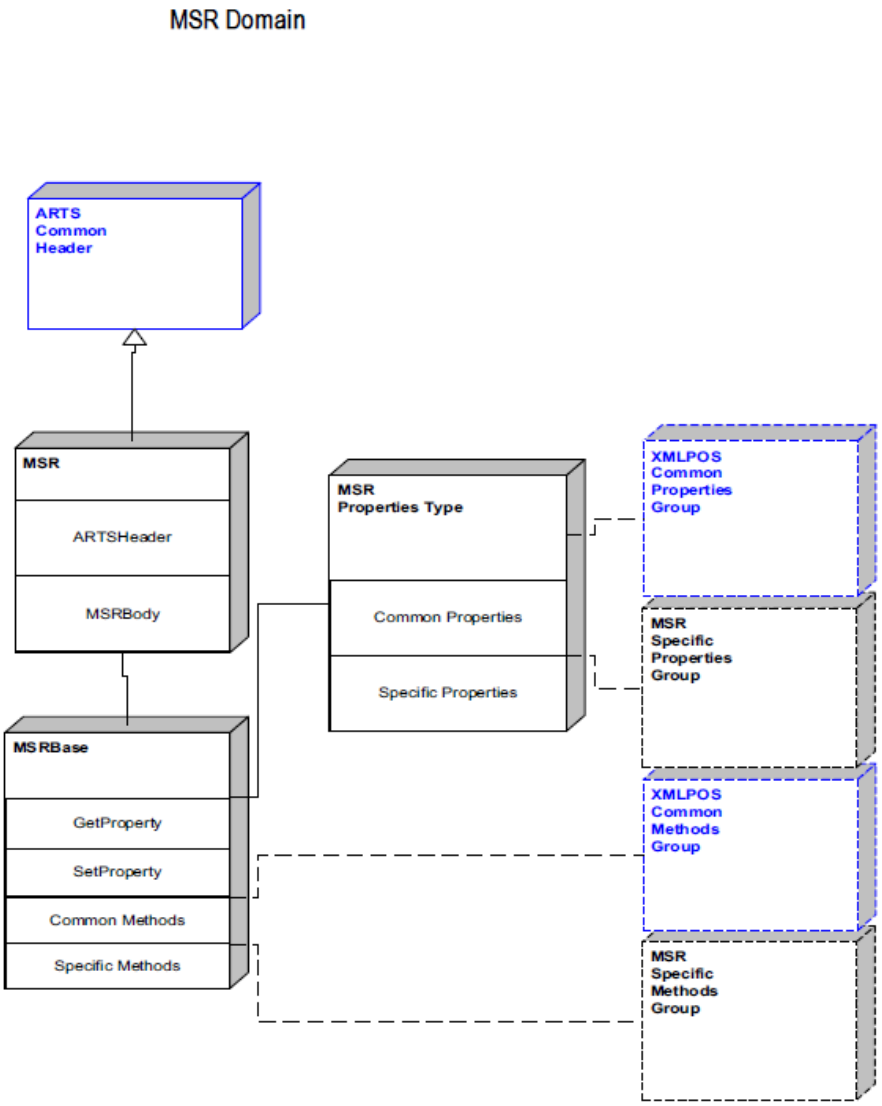


Figure 103: MSR Domain View

MSR Properties

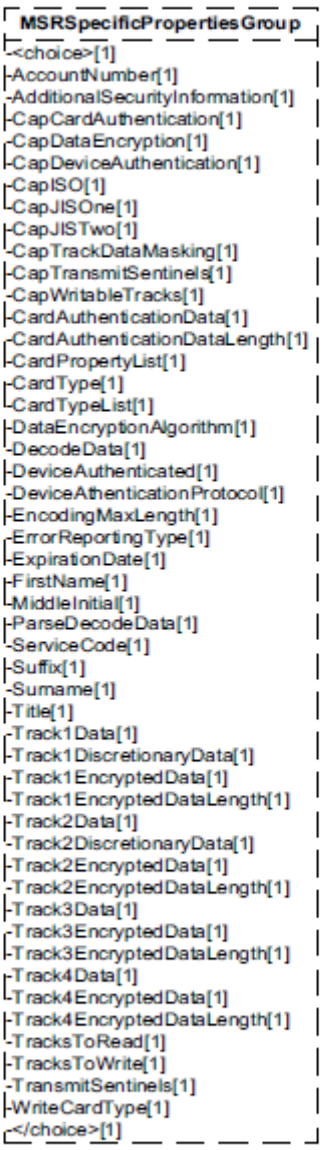


Figure 104: MSR Properties Domain View

MSR Methods

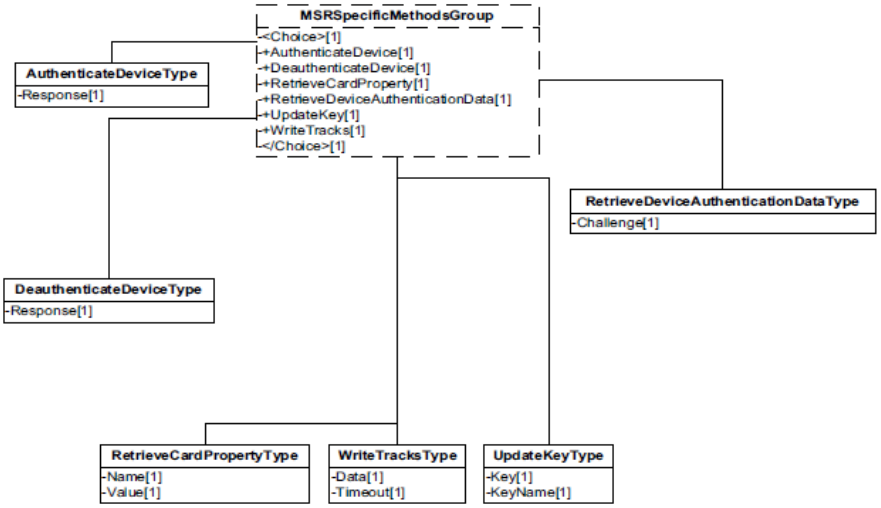


Figure 105: MSR Methods Domain View

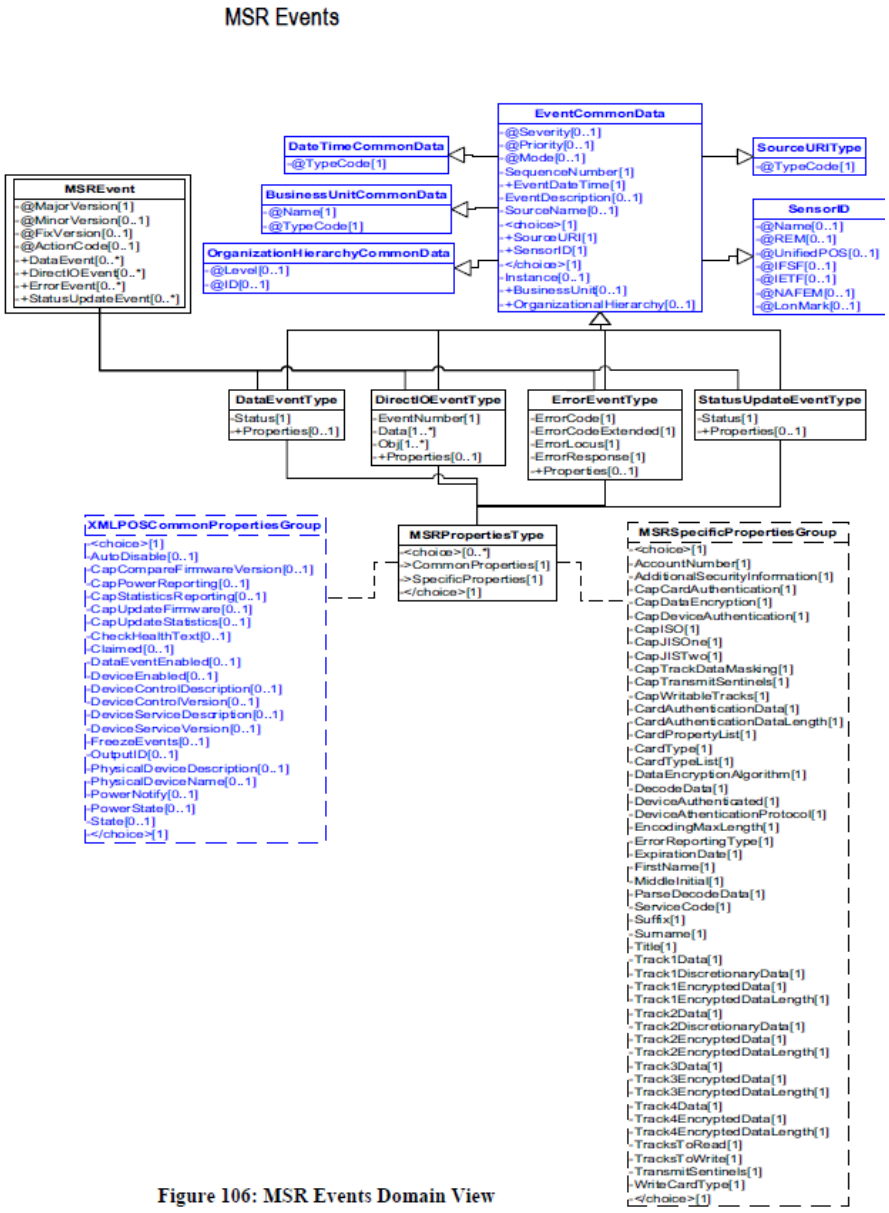


Figure 106: MSR Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
writeTracks			
	E_ILLEGAL	Error	The data to be written exceeds the EncodingMaxLength property for the selected TracksToWrite, or CapWritableTracks is set to MSR_TR_NONE.
	E_FAILURE	Error	A card was swiped within the allotted timeout, but that card or track specified by TracksToWrite is not writable
	E_TIMEOUT	Warning	A card was not swiped within the allotted timeout period

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
N/A		

PIN Pad

PIN Pad Example

→ beginEFTTransaction

```
<?xml version="1.0" encoding="UTF-8"?>
<PINPad xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/PINPad/
        PINPadV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/PINPad/"
        MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>1234</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="PINPad">PP1</SensorID>
    </ARTSHeader>
    <PINPadBody>
            <BeginEFTTransaction>
                    <PINPadSystem>M/S</PINPadSystem>
                    <TransactionHost>0</TransactionHost>
            </BeginEFTTransaction>
    </PINPadBody>
</PINPad>
```

→enablePINEntry

```
<?xml version="1.0" encoding="UTF-8"?>
<PINPad xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/PINPad/
        PINPadV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/PINPad/"
        MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>1235</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="PINPad">PP1</SensorID>
    </ARTSHeader>
    <PINPadBody>
            <EnablePINEntry/>
    </PINPadBody>
</PINPad>
```

← fire DataEvent

```
<?xml version="1.0" encoding="UTF-8"?>
<PINPadEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/PINPadEvents/
        ../PINPadEventV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        PINPadEvents/" MajorVersion="1" MinorVersion="14" FixVersion="1"">
    <DataEvent>
            <SequenceNumber>1236</SequenceNumber>
            <EventDateTime>2001-12-17T09:30:47.0Z</EventDateTime>
            <SensorID UnifiedPOS="PINPad">PP1</SensorID>
            <Status>PPAD_SUCCESS</Status>
    </DataEvent>
</PINPadEvent>
```

→ computeMAC(in, out)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<PINPad xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/PINPad/
        PINPadV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/PINPad/"
        MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>1237</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="PINPad">PP1</SensorID>
    </ARTSHeader>
    <PINPadBody>
            <ComputeMAC>
                    <InMsg>in</InMsg>

        <OutMsg>\u005b\u0029\u003e\u001e\u0030\u0000\u001d\u0030\u0030
        \u0030\u0031\u001d\u0032\u001d\u0033\u001d\u0031\u0032\u0033
        </OutMsg>
            </ComputeMAC>
    </PINPadBody>
</PINPad>
```

→ verifyMAC(message)

```
<?xml version="1.0" encoding="UTF-8"?>
<PINPad xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/PINPad/
        PINPadV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/PINPad/"
        MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>1238</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="PINPad">PP1</SensorID>
    </ARTSHeader>
    <PINPadBody>
            <VerifyMAC>
                    <Message>message</Message>
            </VerifyMAC>
    </PINPadBody>
</PINPad>
```

→endEFTTransaction(PPA_EFT_NORMAL)

```
<?xml version="1.0" encoding="UTF-8"?>
<PINPad xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/PINPad/
        PINPadV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/PINPad/"
        MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>1239</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="PINPad">PP1</SensorID>
    </ARTSHeader>
    <PINPadBody>
            <EndEFTTransaction>
                    <CompletionCode>PPAD_EFT_NORMAL
                    </CompletionCode>
            </EndEFTTransaction>
    </PINPadBody>
</PINPad>
```

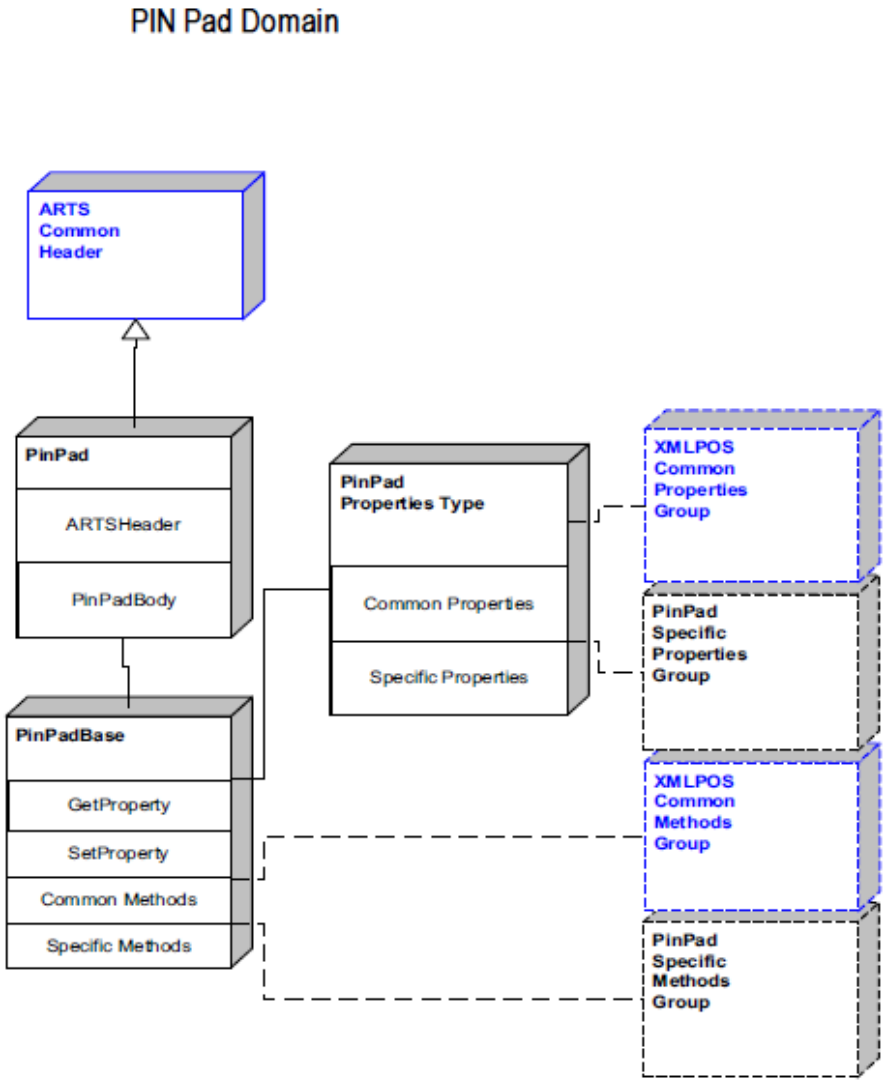


Figure 107: PIN Pad Domain View

PIN Pad Properties

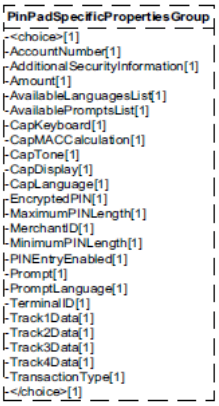


Figure 108: PIN Pad Properties Domain View

PIN Pad Methods

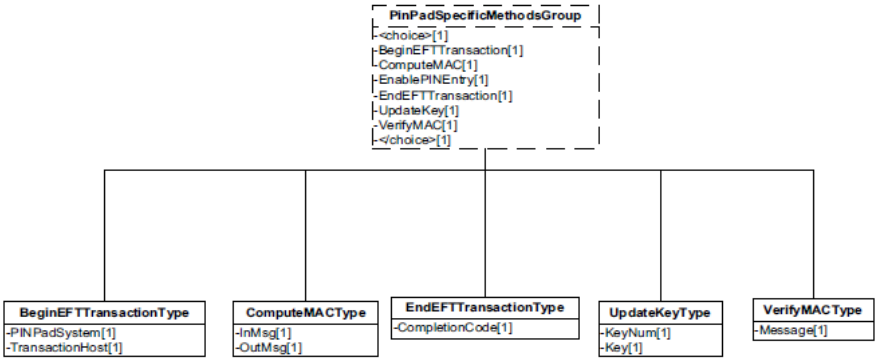


Figure 109: PIN Pad Methods Domain View

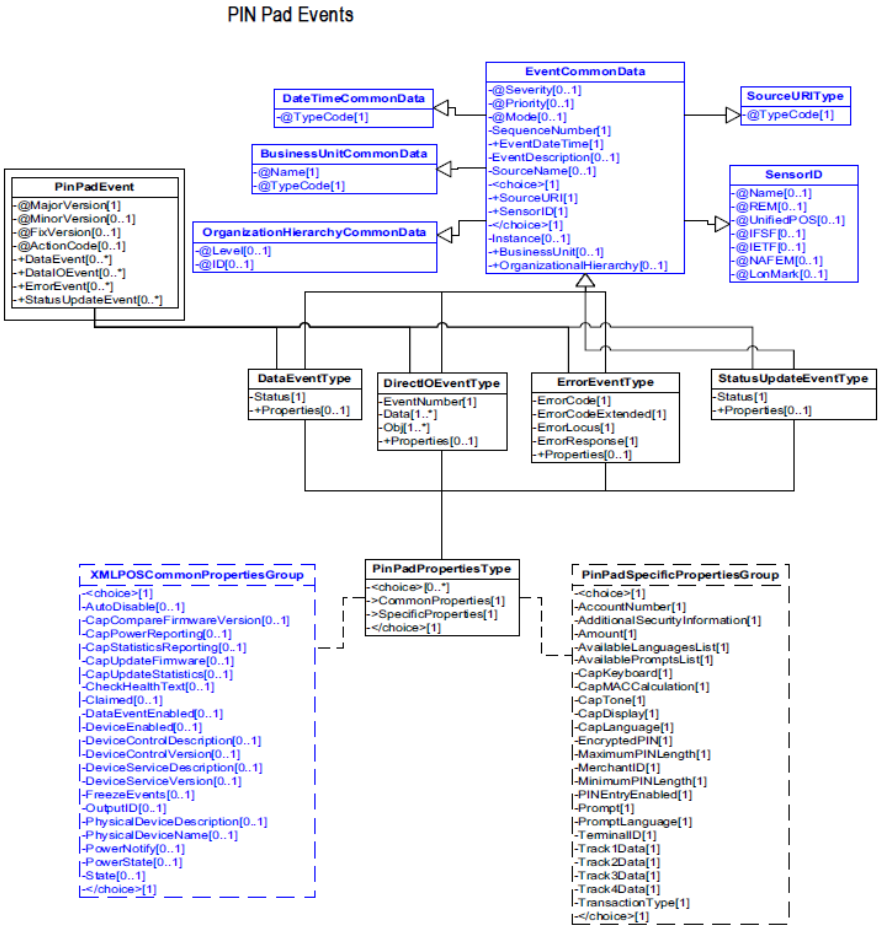


Figure 110: PIN Pad Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
beginEFTTransaction			
	E_ILLEGAL	Error	The requested PIN Pad Management System is not supported by the Control, or the requested EFT Transaction Host is an illegal value for the selected PIN Pad Management System.
	E_BUSY	Warning	The PIN Pad is already performing an EFT transaction.
computeMAC			
	E_DISABLED	Warning	A beginEFTTransaction method has not been performed
	E_BUSY	Warning	**PINEntryEnabled** is true. The PIN Pad cannot perform a MAC calculation during PIN Entry.
enablePINEntry			
	E_DISABLED	Warning	A beginEFTTransaction method has not been performed.
updateKey			
	E_ILLEGAL	Error	One of the following conditions occurred. * The selected PIN Pad Management System does not support this function. * The keyNum specifies an unacceptable key number. * The key contains a bad key (not Hex-ASCII or wrong length or bad parity).
verifyMAC			
	E_BUSY	Warning	PINEntryEnabled is true. The PIN Pad cannot perform a MAC verification during PIN Entry
	E_DISABLED	Warning	A beginEFTTransaction method has not been performed.
	E_FAILURE	Error	The Service failed to verify the MAC value in message.

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
N/A		

Point Card Reader/Writer

Point Card Reader Example

→beginInsertion

```
<?xml version="1.0" encoding="UTF-8"?>
<PointCardRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/PointCardRW/
        PointCardRWV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        PointCardRW/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>1</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="PointcardReaderWriter">PCR1
            </SensorID>
    </ARTSHeader>
    <PointCardRWBody>
            <BeginInsertion>
                    <Timeout>30000</Timeout>
            </BeginInsertion>
    </PointCardRWBody>
</PointCardRW>
```

→endInsertion

```
<?xml version="1.0" encoding="UTF-8"?>
<PointCardRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/PointCardRW/
        PointCardRWV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        PointCardRW/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>2</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="PointcardReaderWriter">PCR1
            </SensorID>
    </ARTSHeader>
    <PointCardRWBody>
            <EndInsertion/>
    </PointCardRWBody>
</PointCardRW>
```

→set DataEventEnabled to true

```
<?xml version="1.0" encoding="UTF-8"?>
<PointCardRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/PointCardRW/
        PointCardRWV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        PointCardRW/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request" ActionCode="Begin">
            <MessageID Name="String"
                    Timestamp="2001-12-17T09:30:47.0Z">12341234
            </MessageID>
            <DateTime TypeCode="Message">2001-12-17T09:30:47.0Z
            </DateTime>
            <SensorID UnifiedPOS="PointcardReaderWriter">PCR1
            </SensorID>
    </ARTSHeader>
```

```
    <PointCardRWBody>
        <SetProperty>
            <DataEventEnabled>true</DataEventEnabled>
        </SetProperty>
    </PointCardRWBody>
</PointCardRW>
```

←fire DataEvent

```
<?xml version="1.0" encoding="UTF-8"?>
<PointCardRWEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/PoinCardEvents/
        PointCardRWEventV1.14.1.xsd" xmlns="http://www.nrf-arts.org/
        UnifiedPOS/PoinCardEvents/" MajorVersion="1" MinorVersion="14"
        FixVersion="1">
    <DataEvent>
        <SequenceNumber>4</SequenceNumber>
        <EventDateTime>2001-12-17T09:30:47.0Z</EventDateTime>
        <SensorID UnifiedPOS="PointcardReaderWriter">PCR1
        </SensorID>
        <Status>0</Status>
    </DataEvent>
</PointCardRWEvent>
```

→printWrite(1, 0, 0, "1000 points")

```
<?xml version="1.0" encoding="UTF-8"?>
<PointCardRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/PointCardRW/
        PointCardRWV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        PointCardRW/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
        <MessageID>5</MessageID>
        <DateTime>2001-12-17T09:30:47.0Z</DateTime>
        <SensorID UnifiedPOS="PointcardReaderWriter">PCR1
        </SensorID>
    </ARTSHeader>
    <PointCardRWBody>
        <PrintWrite>
            <Kind>Print</Kind>
            <HPosition>0</HPosition>
            <VPosition>0</VPosition>
            <Data>1000 points</Data>
        </PrintWrite>
    </PointCardRWBody>
</PointCardRW>
```

→beginRemoval

```
<?xml version="1.0" encoding="UTF-8"?>
<PointCardRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/PointCardRW/
        PointCardRWV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        PointCardRW/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>6</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="PointcardReaderWriter">PCR1
            </SensorID>
    </ARTSHeader>
    <PointCardRWBody>
            <BeginRemoval>
                    <Timeout>30000</Timeout>
            </BeginRemoval>
    </PointCardRWBody>
</PointCardRW>
```

→endRemoval

```
<?xml version="1.0" encoding="UTF-8"?>
<PointCardRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/PointCardRW/
        PointCardRWV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        PointCardRW/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>7</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="PointcardReaderWriter">PCR1
            </SensorID>
    </ARTSHeader>
    <PointCardRWBody>
            <EndRemoval/>
    </PointCardRWBody>
</PointCardRW>
```

Point Card Reader Domain

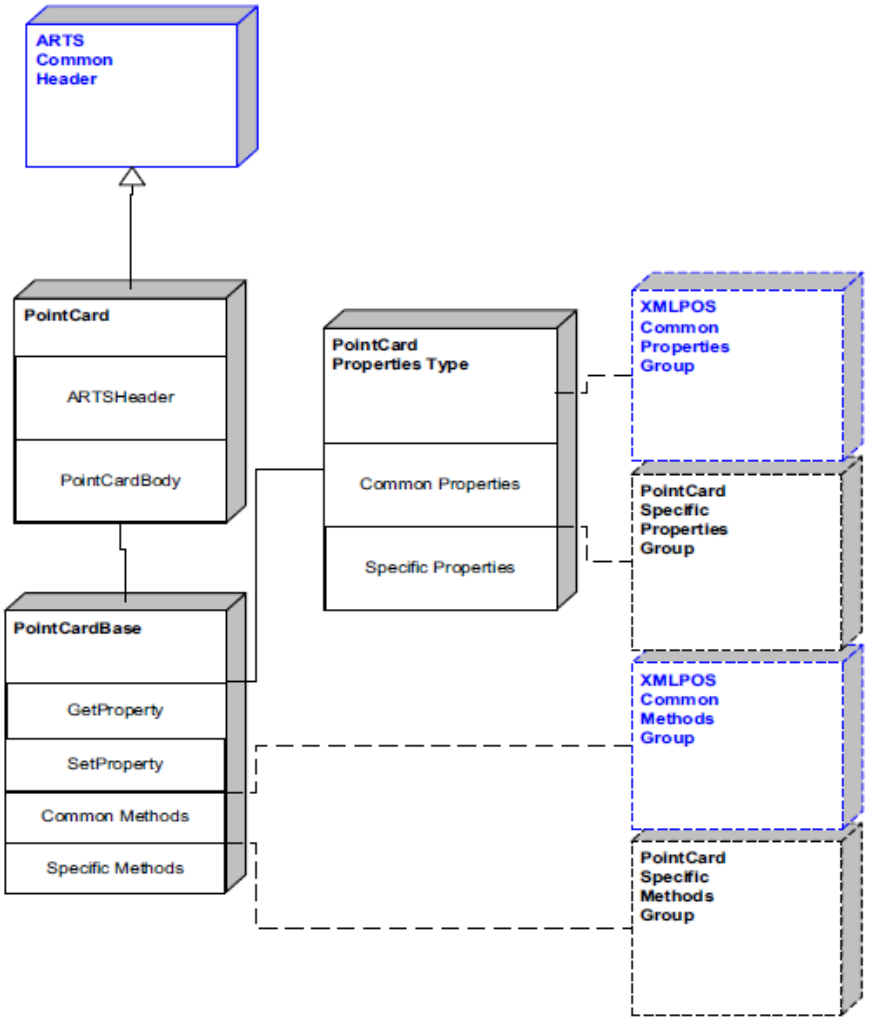


Figure 111: Point Card RW Domain View

Point Card Reader Properties

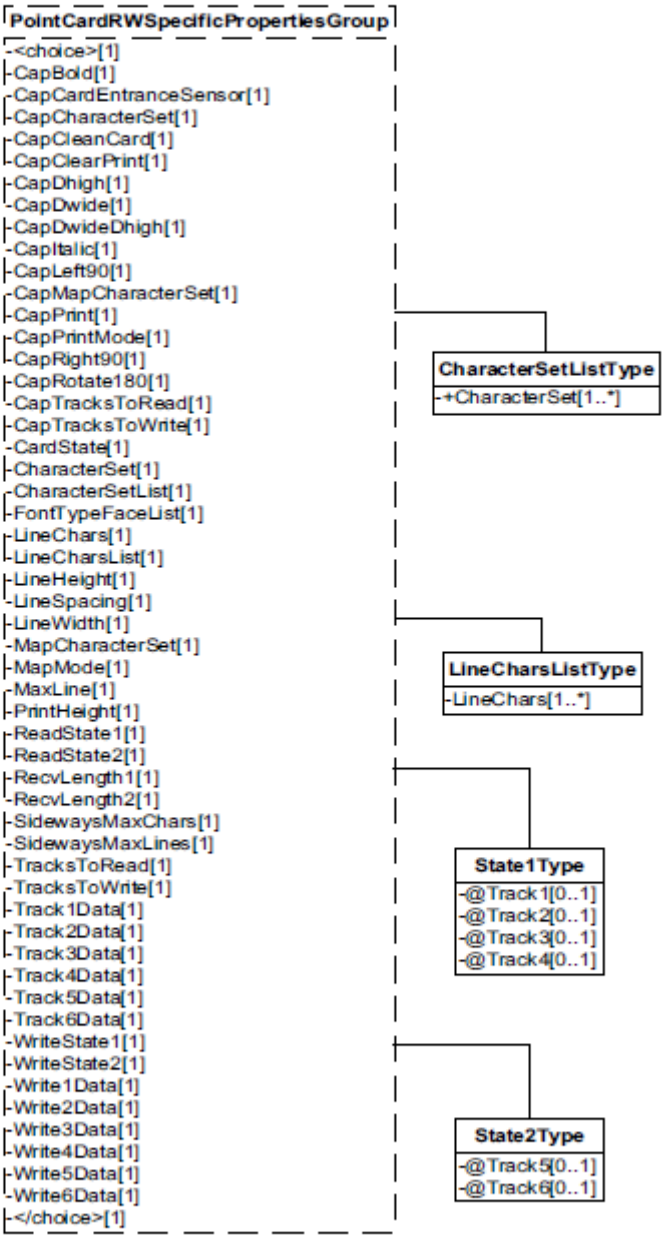


Figure 112: Point Card RW Properties Domain View

Point Card Reader Methods

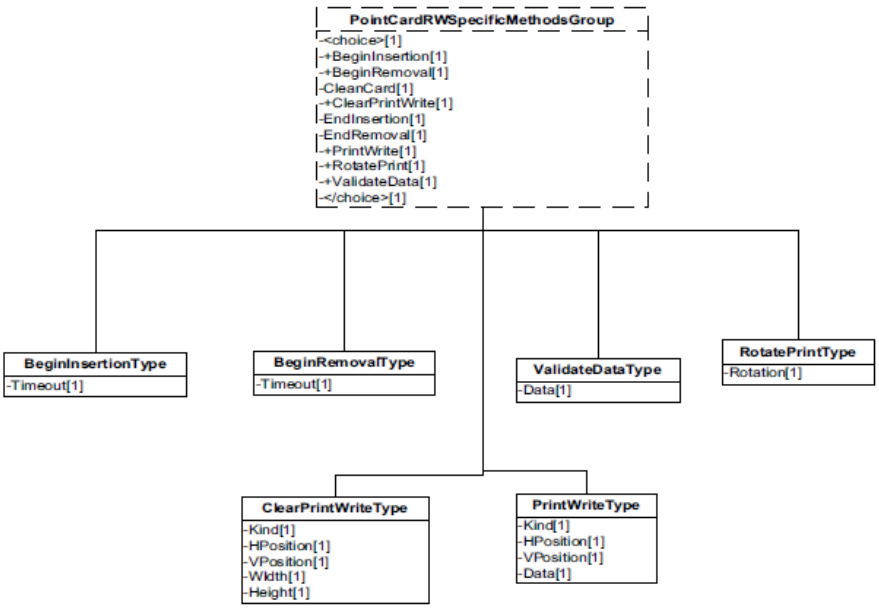


Figure 113: Point Card RW Methods Domain View

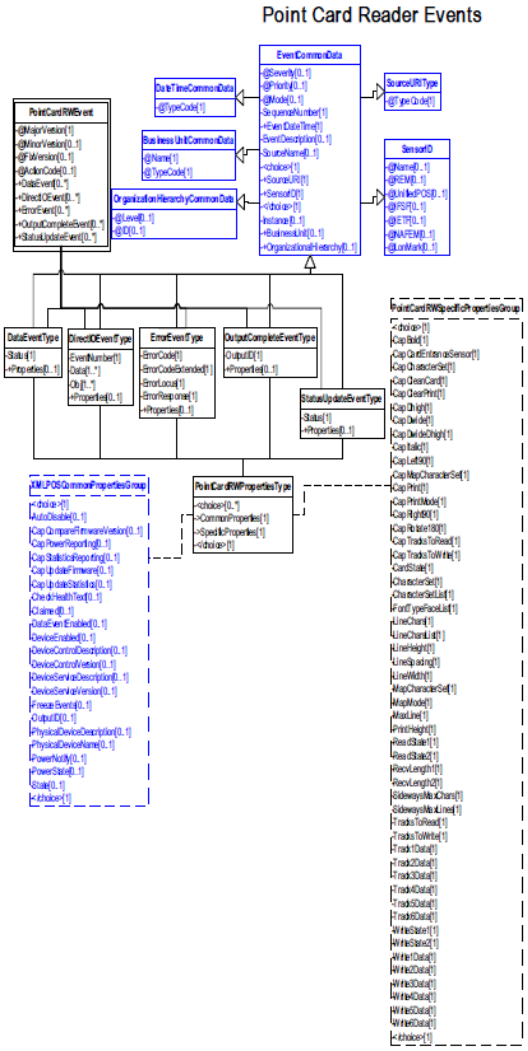


Figure 114: Point Card RW Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
beginInsertion			
	E_BUSY	Warning	This operation cannot be performed because asynchronous output is in progress.
	E_ILLEGAL	Error	The Point Card Reader Writer does not exist or an invalid timeout parameter was specified.
	E_TIMEOUT	Warning	The specified time has elapsed without the point card being properly inserted.
beginRemoval			
	E_BUSY	Warning	This operation cannot be performed because asynchronous output is in progress.
	E_ILLEGAL	Error	The Point Card Reader Writer does not exist or an invalid timeout parameter was specified.
	E_TIMEOUT	Warning	The specified time has elapsed without the point card being properly inserted.
cleanCard			
	E_ILLEGAL	Error	The Point Card Reader Writer does not exist or CapCleanCard is false.
clearPrintWrite			
	E_BUSY	Warning	This operation cannot be performed because asynchronous output is in progress.
endInsertion			
	E_ILLEGAL	Warning	The Point Card Reader Writer is not in point card insertion mode.
	E_FAILURE	Warning	A card is not inserted in the Point Card Reader Writer.
endRemoval			
	E_ILLEGAL	Error	The Point Card Reader Writer is not in point card removal mode.
	E_FAILURE	Warning	There is a card in the Point Card Reader Writer.
printWrite			
	E_ILLEGAL	Warning	There is no card in the Point Card Reader Writer.
rotatePrint			

	E_BUSY	Warning	This operation cannot be performed because asynchronous output is in progress.
	E_ILLEGAL	Error	The Point Card Reader Writer does not support the specified rotation.
validateData			
	E_ILLEGAL	Warning	Some of the data is not precisely supported by the device, but the Control can select valid alternatives.
	E_FAILURE	Error	Some of the data is not supported. No alternatives can be selected.
E_EXTENDED			
	EPCRW_READ	Error	There was a read error
	EPCRW_WRITE	Error	There was a write error
	EPCRW_JAM	Error	There was a card jam
	EPCRW_MOTOR	Error	There was a conveyance motor error
	EPCRW_COVER	Error	The conveyance motor cover was open
	EPCRW_PRINTER	Error	The printer has an error
	EPCRW_RELEASE	Warning	There is a card remaining in the entrance
	EPCRW_DISPLAY	Error	There was a display indicator error
	EPCRW_NOCARD	Warning	There is no card in the reader

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
PCRW_SUE_NOCARD	Warning	No card or card sensor position indeterminate
PCRW_SUE_REMAINING	Warning	Card remaining in the entrance
PCRW_SUE_INRW	Warning	There is a card in the device

POS Keyboard

POS Keyboard Example

← fire DataEvent

```
<?xml version="1.0" encoding="UTF-8"?>
<POSKeyboardEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        POSKeyboardEvents/ POSKeyboardEventV1.14.1.xsd" xmlns="http://
        www.nrf-arts.org/UnifiedPOS/POSKeyboardEvents/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <DataEvent>
            <SequenceNumber>4294967295</SequenceNumber>
            <EventDateTime>2001-12-17T09:30:47.0Z</EventDateTime>
            <SensorID UnifiedPOS="POSKeyboard">KBD1</SensorID>
            <Status>0</Status>
    </DataEvent>
</POSKeyboardEvent>
```

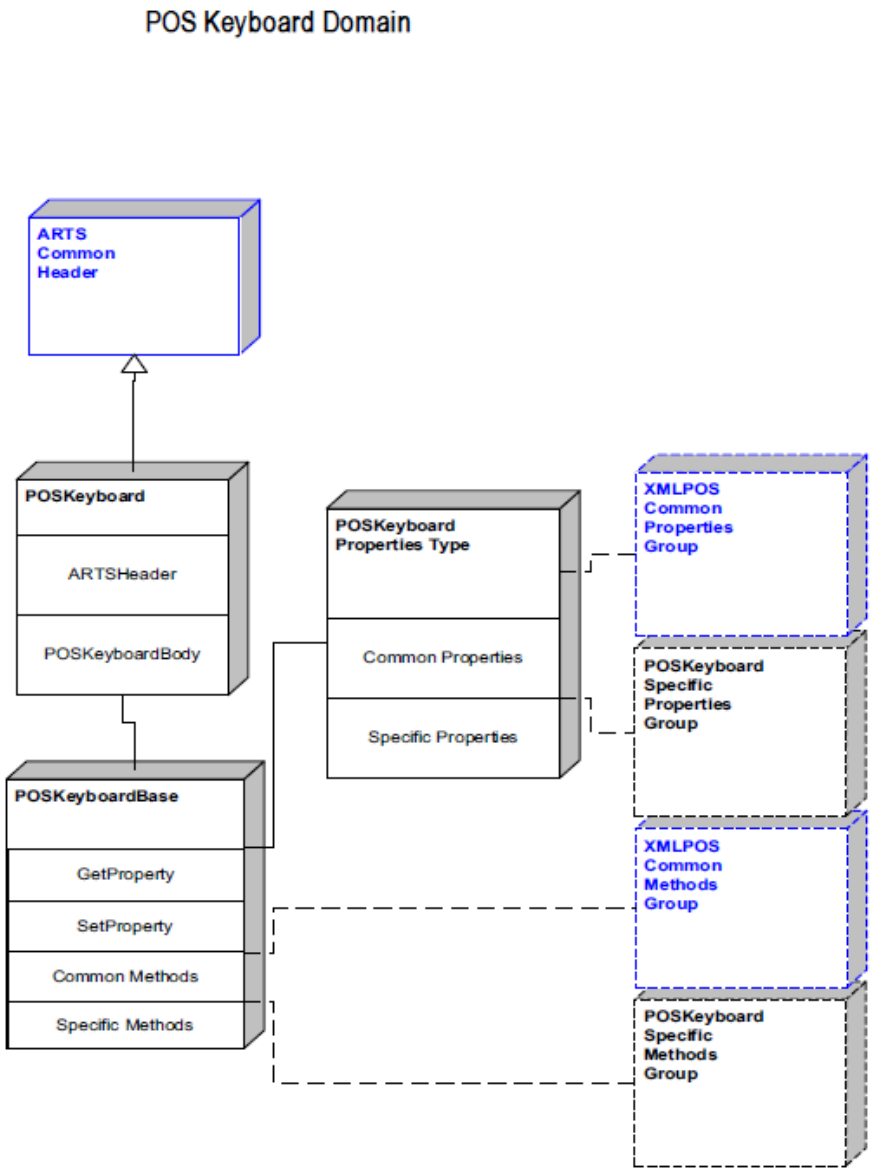


Figure 115: POS Keyboard Domain View

POS Keyboard Properties

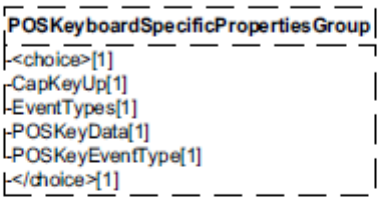


Figure 116: POS Keyboard Properties Domain View

POS Keyboard Methods

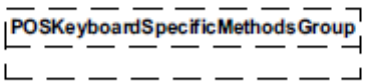


Figure 117: POS Keyboard Methods Domain View

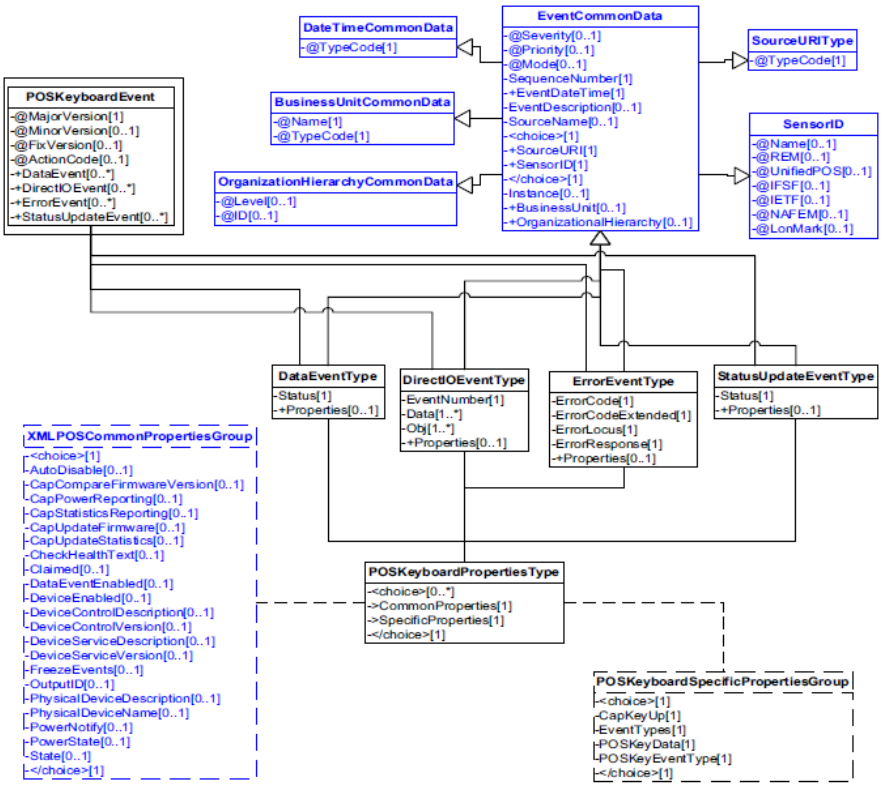


Figure 118: POS Keyboard Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Status Codes to Message Severity Codes

Method	Value	Severity	Meaning
N/A			

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
N/A		

POS Power

POS Power Example

→ set PowerNotify to true

```
<?xml version="1.0" encoding="UTF-8"?>
<POSPower xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/POSPower/
        POSPowerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        POSPower/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request" ActionCode="Begin">
            <MessageID>1234</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="POSPower">Power1</SensorID>
    </ARTSHeader>
    <POSPowerBody>
            <SetProperty>
                    <PowerNotify>PN_ENABLED</PowerNotify>
            </SetProperty>
    </POSPowerBody>
</POSPower>
```

← fire StatusUpdateEvent (power low)

```
<?xml version="1.0" encoding="UTF-8"?>
<POSPowerEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        POSPowerEvents/ POSPowerEventV1.14.1.xsd" xmlns="http://www.nrf-
        arts.org/UnifiedPOS/POSPowerEvents/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <StatusUpdateEvent>
            <SequenceNumber>4294967295</SequenceNumber>
            <EventDateTime TypeCode="Message">2001-12-17T09:30:47.0Z
            </EventDateTime>
            <SensorID UnifiedPOS="POSPower">Pwr1</SensorID>
            <Status>PWR_SUE_UPS_LOW</Status>
    </StatusUpdateEvent>
</POSPowerEvent>
```

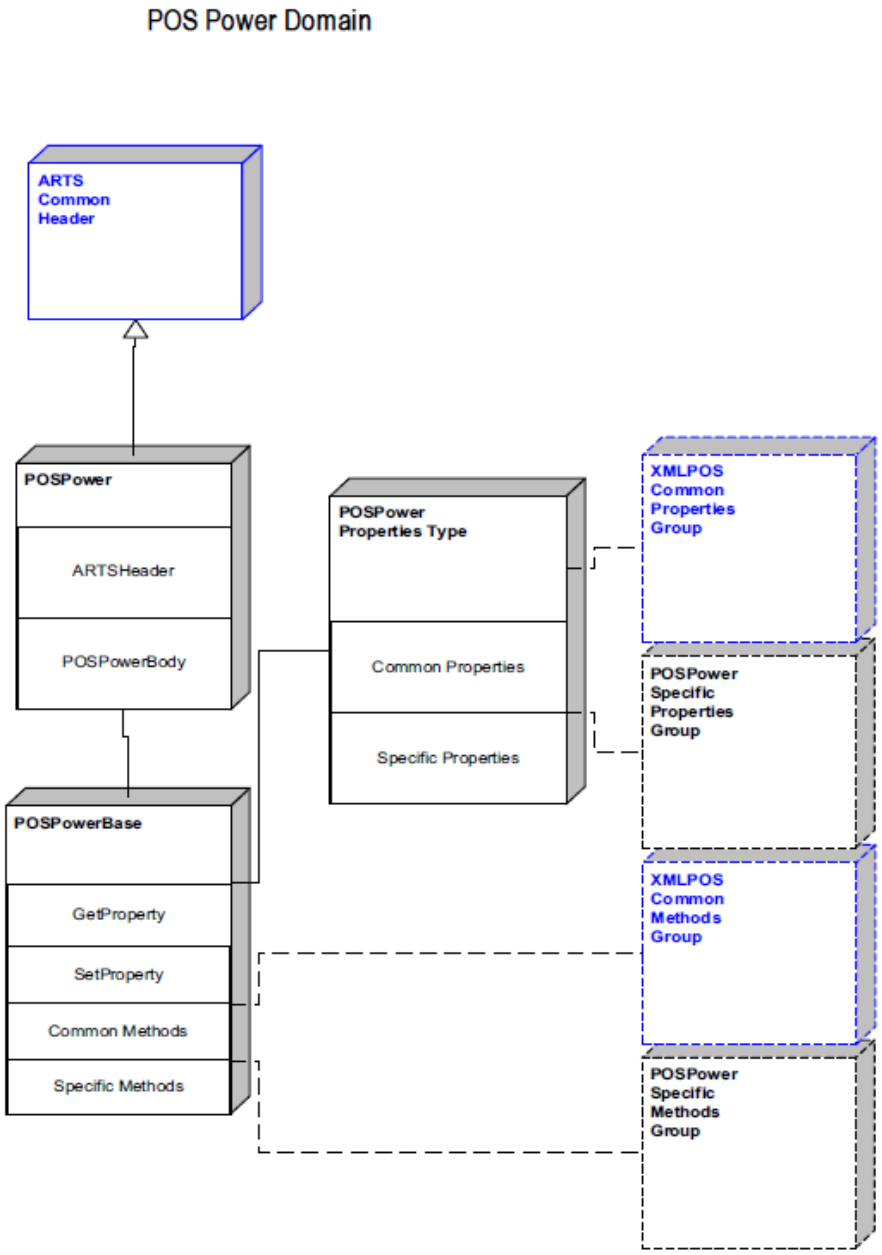


Figure 119: POS Power Domain View

POS Power Properties

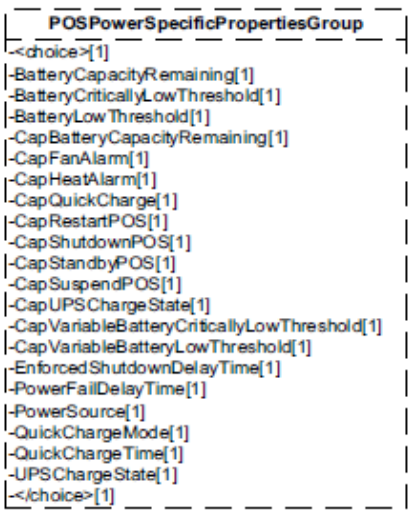


Figure 120: POS Power Properties Domain View

POS Power Methods

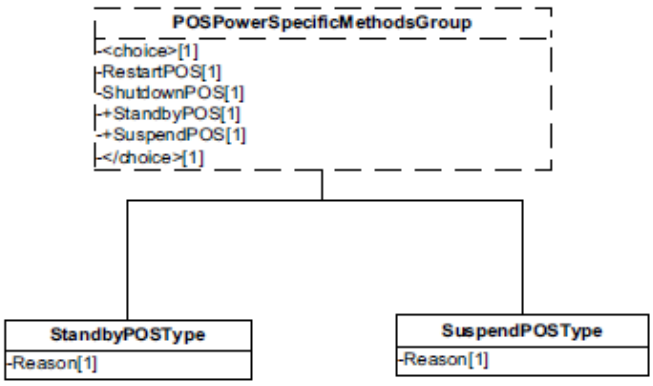


Figure 121: POS Power Methods Domain View

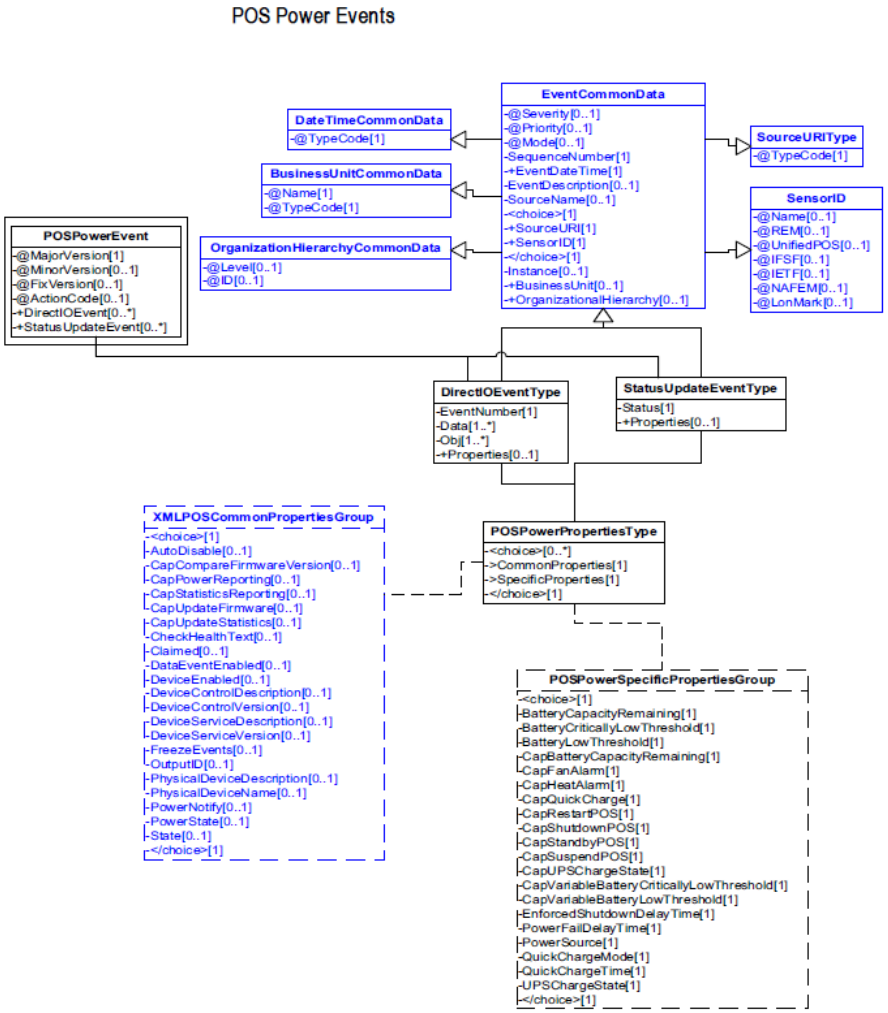


Figure 122: POS Power Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
restartPOS			
	E_ILLEGAL	Error	This method is not supported
shutdownPOS			
	E_ILLEGAL	Error	This method is not supported
standbyPOS			
	E_ILLEGAL	Error	This method is not supported
suspendPOS			
	E_ILLEGAL	Error	This method is not supported

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
PWR_SUE_UPS_FULL UPS	Information	battery is near full charge. Can be returned if **CapUPSChargeState** contains PWR_UPS_FULL
PWR_SUE_UPS_WARNING	Warning	UPS battery is near 50% charge. Can be returned if **CapUPSChargeState** contains PWR_UPS_WARNING
PWR_SUE_UPS_LOW	Warning	UPS battery is near empty. Application shutdown should be started to ensure that it can be completed before the battery charge is depleted. A minimum of 2 minutes of normal system operation can be assumed when this state is entered unless this is the first charge state reported upon entering the "Off" state. Can be returned if **CapUPSChargeState** contains PWR_UPS_LOW.
PWR_SUE_UPS_CRITICAL	Warning	UPS is in critical state, and will in short time be disconnected. Can be returned if **CapUPSChargeState** contains PWR_UPS_CRITICAL
PWR_SUE_FAN_STOPPED	Error	The CPU fan is stopped. Can be returned if **CapFanAlarm** is true.
PWR_SUE_FAN_RUNNING	Information	The CPU fan is running. Can be returned if **CapFanAlarm** is true.
PWR_SUE_TEMPERATURE_HIGH	Error	The CPU is running on high temperature. Can be returned if **CapHeatAlarm** is true.
PWR_SUE_TEMPERATURE_OK	Information	The CPU is running on normal temperature. Can be returned if **CapHeatAlarm** is true.

PWR_SUE_SHUTDOWN	Error	The system will shutdown immediately
PWR_SUE_BAT_LOW	Warning	The system remaining battery capacity is at or below the low battery threshold and the system is operating from the battery
PWR_SUE_BAT_CRITICAL	Error	The system remaining battery capacity is at or below the critically low battery threshold and the system is operating from the battery.
PWR_SUE_BAT_CAPACITY_REMAINING	Information	The **BatteryCapacityRemaining** property has been updated
PWR_SUE_RESTART	Warning	The system will restart immediately.
PWR_SUE_STANDBY	Information	The system is requesting a transition to the Standby state
PWR_SUE_USER_STANDBY	Information	The system is requesting a transition to the Standby state as a result of user input.
PWR_SUE_SUSPEND	Information	The system is requesting a transition to the Suspend state.
PWR_SUE_USER_SUSPEND	Information	The system is requesting a transition to the Suspend state as a result of user input
PWR_SUE_PWR_SOURCE	Information	The **PowerSource** property has been updated

POS Printer

POS Printer Example

→ changePrintSide(PTR_PS_SIDE1)

```
<?xml version="1.0" encoding="UTF-8"?>
<POSPrinter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/POSPrinter/
        POSPrinterV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        POSPrinter/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>1234</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="POSPrinter">PTR1</SensorID>
    </ARTSHeader>
    <POSPrinterBody>
            <ChangePrintSide>
                    <Side>PTR_PS_SIDE1</Side>
            </ChangePrintSide>
    </POSPrinterBody>
</POSPrinter>
```

→ printNormal(PTR_S_SLIP, "Some String Data")

```
<?xml version="1.0" encoding="UTF-8"?>
<POSPrinter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/POSPrinter/
        POSPrinterV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        POSPrinter/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>587689</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="POSPrinter">Prt1</SensorID>
    </ARTSHeader>
    <POSPrinterBody>
            <PrintNormal>
                    <Station>PTR_S_SLIP</Station>
                    <Data>Some String Data</Data>
            </PrintNormal>
    </POSPrinterBody>
</POSPrinter>
```

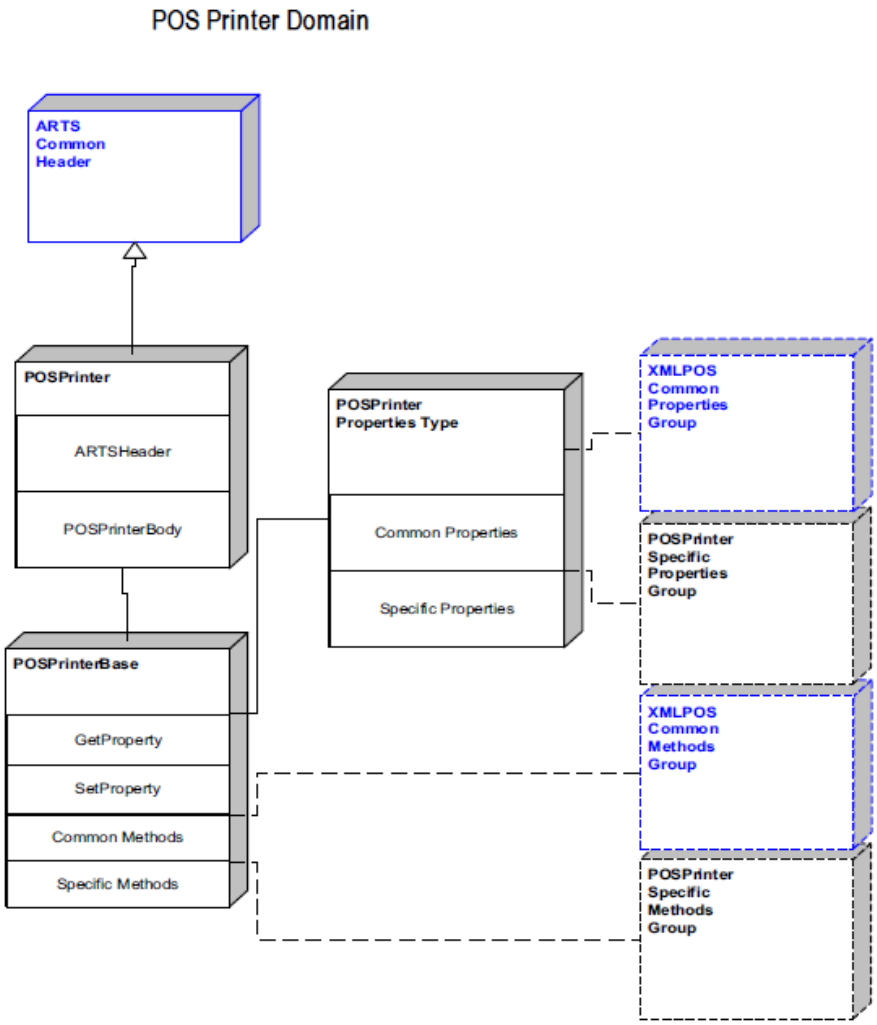


Figure 123: POS Printer Domain View

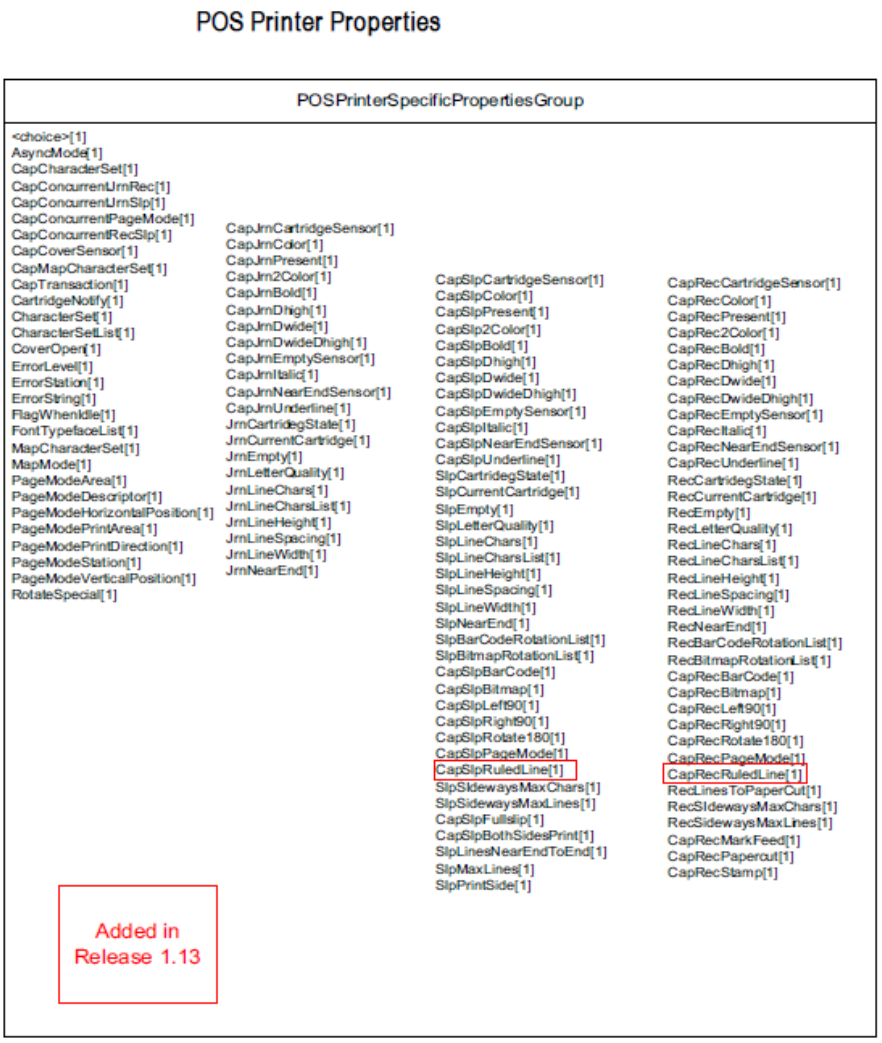


Figure 124: POS Printer Properties Domain View

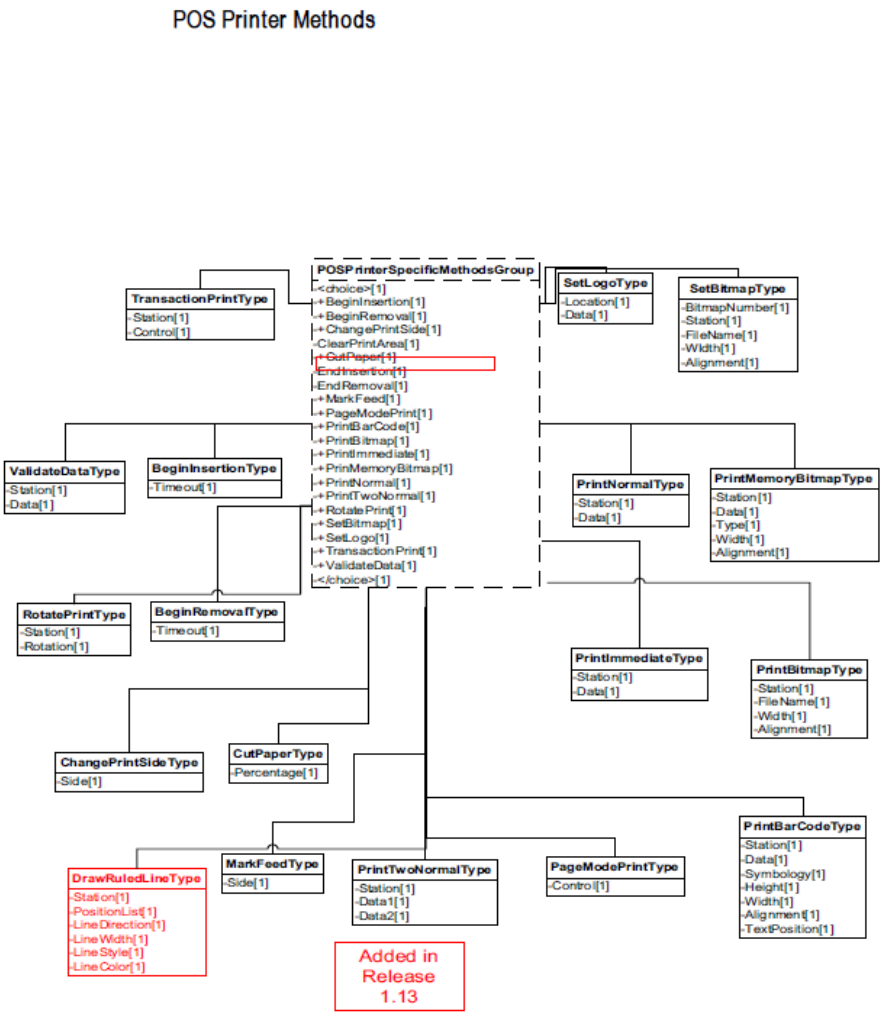


Figure 125: POS Printer Methods Domain View

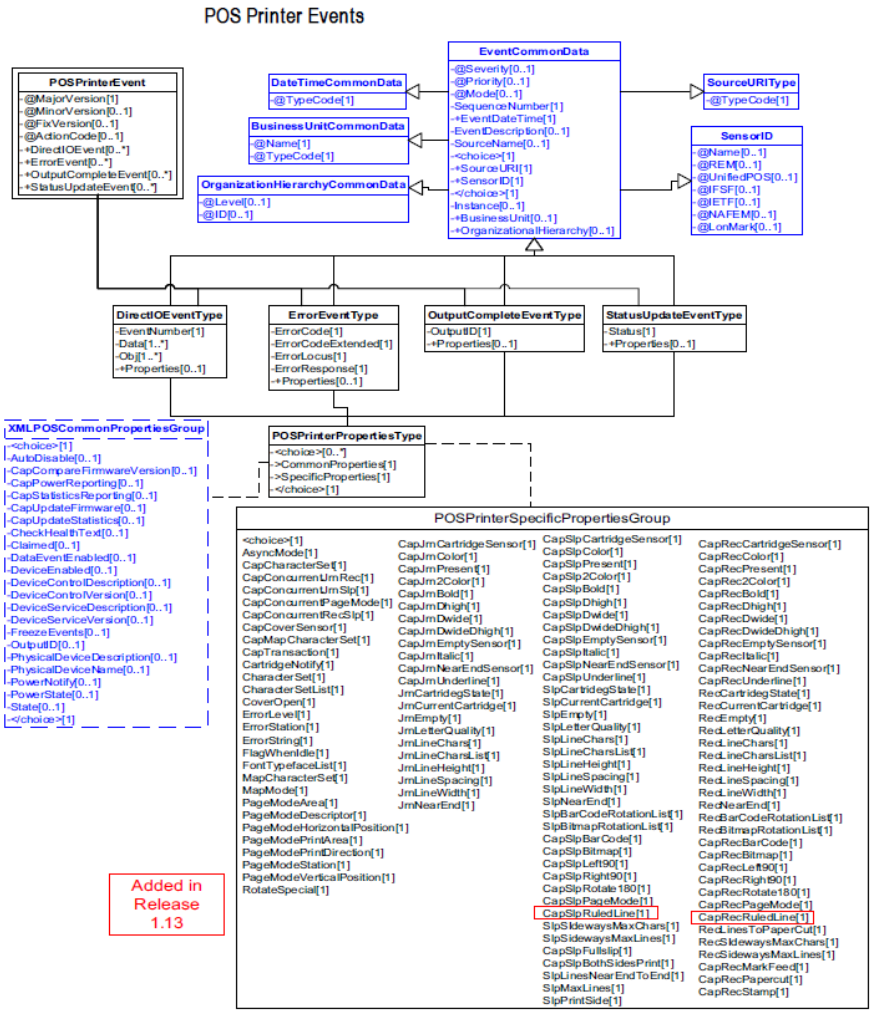


Figure 126: POS Printer Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
beginInsertion			
	E_BUSY	Warning	Cannot perform request while output is in progress.
	E_ILLEGAL	Error	The slip station does not exist or an invalid timeout parameter was specified.
	E_TIMEOUT	Warning	The specified time has elapsed without the form being properly inserted
beginRemoval			
	E_BUSY	Warning	Cannot perform request while output is in progress.
	E_ILLEGAL	Information	The Fiscal Printer does not have a slip station or an invalid timeout parameter was specified.
	E_TIMEOUT	Warning	The specified time has elapsed without the form being properly removed.
changePrintSide			
	E_BUSY	Warning	Cannot perform request while output is in progress.
	E_ILLEGAL	Error	One of the following errors occurred: * The slip station does not exist * the printer does not support both sides printing * an invalid side parameter was specified
	EPTR_COVER_OPEN	Error	The printer cover is open.
	EPTR_SLP_EMPTY	Warning	The slip station was specified, but a form is not inserted.
	EPTR_SLP_CARTRIDGE_EMPTY	Error	A slip station cartridge is empty.
	EPTR_SLP_CARTRIDGE_REMOVED	Error	A slip station cartridge has been removed.
	EPTR_SLP_HEAD_CLEANING	Warning	A slip station head is being cleaned.
cutPaper			
	E_BUSY	Warning	Cannot perform request while output is in progress.
	E_ILLEGAL	Error	An invalid percentage was specified, the receipt station does not exist, the receipt printer does not have paper cutting ability, or Page Mode for the receipt station is active.
	EPTR_COVER_OPEN	Error	The printer cover is open.
	EPTR_REC_EMPTY	Error	The receipt station is out of paper.

endInsertion			
	E_BUSY	Warning	Cannot perform request while output is in progress.
	E_ILLEGAL	Warning	The Fiscal Printer is not in slip insertion mode.
	EPTR_COVER_OPEN	Warning	The device was taken out of insertion mode while the Printer cover was open.
	EPTR_SLP_EMPTY	Warning	The device was taken out of insertion mode without a form being inserted.
endRemoval			
	E_BUSY	Warning	Cannot perform request while output is in progress.
	E_ILLEGAL	Warning	The Printer is not in slip removal mode.
	EFPTR_SLP_FORM	Warning	The device was taken out of removal mode while a form was still present.
markFeed			
	E_BUSY	Warning	Cannot perform request while output is in progress.
	E_ILLEGAL	Error	The receipt print station does not support the given mark sensed paper handling function.
	EPTR_COVER_OPEN	Error	The printer cover is open.
	EPTR_REC_EMPTY	Error	The receipt paper is empty.
pageModePrint			
	E_BUSY	Warning	Cannot perform request while output is in progress.
	E_ILLEGAL	Error	The specified PageModeStation does not exist, or CapxxxPageMode is false, or the specified PageModeStation is not in Page Mode and control is set to PTR_PM_NORMAL, PTR_PM_PRINT_SAVE, or PTR_PM_CANCEL
printBarCode			
	E_BUSY	Warning	Cannot perform request while output is in progress.

	E_ILLEGAL	Error	One of the following parameter errors occured: * station does not exist * station does not support bar code printing * height or width is zero or too big * symbology is not supported * not all characters in data are supported by symbology * alignment is invalid or too big * Code Set is not specified for PTR_BCS_Code128_Parsed at start of data * textPosition is invalid, or * the RotateSpecial rotation is not supported
	EPTR_COVER_OPEN	Error	The printer cover is open.
	EPTR_REC_EMPTY	Error	The receipt paper is empty.
	EPTR_REC_CARTRIDGE_REMOVED	Error	A receipt cartridge has been removed.
	EPTR_REC_CARTRIDGE_EMPTY	Error	A receipt cartridge is empty.
	EPTR_REC_HEAD_CLEANING	Warning	A receipt cartridge head is being cleaned.
	EPTR_SLP_EMPTY	Warning	The slip station was specified, but a form is not inserted.
	EPTR_SLP_CARTRIDGE_REMOVED	Error	A slip cartridge has been removed.
	EPTR_SLP_CARTRIDGE_EMPTY	Error	A slip cartridge is empty.
	EPTR_SLP_HEAD_CLEANING	Warning	A slip cartridge head is being cleaned.
printBitmap			
	E_BUSY	Warning	Cannot perform while output is in progress.
	E_ILLEGAL	Warning	The Fiscal Printer does not support duplicate receipts or there is no buffered transaction to print
	EFPTR_WRONG_STATE	Warning	The Fiscal Printer is not currently in the Monitor state
	EFPTR_JRN_EMPTY	Error	The journal station is out of paper.
	EFPTR_REC_EMPTY	Error	The receipt station is out of paper
printFiscalDocumentLine			
	E_BUSY	Warning	Cannot perform while output is in progress.

	E_ILLEGAL	Error	One of the following parameter errors occurred: * station does not exist * station does not support bitmap printing * width parameter is invalid or too big * alignment is invalid or too big
	EPTR_TOOBIG	Error	The bitmap is either too wide to print without transformation, or it is too big to transform
	EPTR_COVER_OPEN	Error	The printer cover is open.
	EPTR_BADFORMAT	Error	The specified file is either not a bitmap file, or it is in an unsupported format.
	EPTR_REC_EMPTY	Error	The receipt station was specified but is out of paper.
	EPTR_REC_CARTRIDGE_REMOVED	Error	A receipt cartridge has been removed.
	EPTR_REC_CARTRIDGE_EMPTY	Error	A receipt cartridge is empty.
	EPTR_REC_HEAD_CLEANING	Warning	A receipt cartridge head is being cleaned.
	EPTR_SLP_EMPTY	Warning	The slip station was specified, but a form is not inserted.
	EPTR_SLP_CARTRIDGE_REMOVED	Error	A slip cartridge has been removed.
	EPTR_SLP_CARTRIDGE_EMPTY	Error	A slip cartridge is empty.
	EPTR_SLP_HEAD_CLEANING	Warning	A slip cartridge head is being cleaned.
printImmediate			
	E_ILLEGAL	Error	The specified station does not exist, or the station is in Page Mode and the device does not support direct printing in Page Mode.
	EPTR_COVER_OPEN	Error	The printer cover is open.
	EPTR_JRN_EMPTY	Error	The journal station was specified but is out of paper.
	EPTR_JRN_CARTRIDGE_REMOVED	Error	A journal cartridge has been removed.
	EPTR_JRN_CARTRIDGE_EMPTY	Error	A journal cartridge is empty.
	EPTR_JRN_HEAD_CLEANING	Warning	A journal cartridge head is being cleaned
	EPTR_REC_EMPTY	Error	The receipt station was specified but is out of paper

	EPTR_REC_CARTRIDGE_REMOVED	Error	A receipt cartridge has been removed.
	EPTR_REC_CARTRIDGE_EMPTY	Error	A receipt cartridge is empty
	EPTR_REC_HEAD_CLEANING	Warning	A receipt cartridge head is being cleaned.
	EPTR_SLP_EMPTY	Warning	The slip station was specified, but a form is not inserted.
	EPTR_SLP_CARTRIDGE_REMOVED	Error	A slip cartridge has been removed.
	EPTR_SLP_CARTRIDGE_EMPTY	Error	A slip cartridge is empty
	EPTR_SLP_HEAD_CLEANING	Warning	A slip cartridge head is being cleaned.
printMemoryBitmap			
	E_BUSY	Warning	Cannot perform while output is in progress.
	E_ILLEGAL	Error	One of the following parameter errors occurred: * station does not exist * station does not support bitmap printing * width parameter is invalid or too big * alignment is invalid or too big
	EPTR_TOOBIG	Error	The bitmap is either too wide to print without transformation, or it is too big to transform.
	EPTR_COVER_OPEN	Error	The printer cover is open.
	EPTR_BADFORMAT	Error	The specified file is either not a bitmap file, or it is in an unsupported format.
	EPTR_REC_EMPTY	Error	The receipt station was specified but is out of paper.
	EPTR_REC_CARTRIDGE_REMOVED	Error	A receipt cartridge has been removed.
	EPTR_REC_CARTRIDGE_EMPTY	Error	A receipt cartridge is empty.
	EPTR_REC_HEAD_CLEANING	Warning	A receipt cartridge head is being cleaned
	EPTR_SLP_EMPTY	Warning	The slip station was specified, but a form is not inserted.
	EPTR_SLP_CARTRIDGE_REMOVED	Error	A slip cartridge has been removed.
	EPTR_SLP_CARTRIDGE_EMPTY	Error	A slip cartridge is empty.
	EPTR_SLP_HEAD_CLEANING	Warning	A slip cartridge head is being cleaned.

printNormal			
	E_ILLEGAL	Error	The specified station does not exist.
	E_BUSY	Warning	Cannot perform while output is in progress.
	EPTR_COVER_OPEN	Error	The printer cover is open.
	EPTR_JRN_EMPTY	Error	The journal station was specified but is out of paper.
	EPTR_JRN_CARTRIDGE_REMOVED	Error	A journal cartridge has been removed.
	EPTR_JRN_CARTRIDGE_EMPTY	Error	A journal cartridge is empty.
	EPTR_JRN_HEAD_CLEANING	Warning	A journal cartridge head is being cleaned.
	EPTR_REC_EMPTY	Error	The receipt station was specified but is out of paper.
	EPTR_REC_CARTRIDGE_REMOVED	Error	A receipt cartridge has been removed.
	EPTR_REC_CARTRIDGE_EMPTY	Error	A receipt cartridge is empty.
	EPTR_REC_HEAD_CLEANING	Warning	A receipt cartridge head is being cleaned
	EPTR_SLP_EMPTY	Warning	The slip station was specified, but a form is not inserted.
	EPTR_SLP_CARTRIDGE_REMOVED	Error	A slip cartridge has been removed.
	EPTR_SLP_CARTRIDGE_EMPTY	Error	A slip cartridge is empty.
	EPTR_SLP_HEAD_CLEANING	Warning	A slip cartridge head is being cleaned.
printTwoNormal			
	E_ILLEGAL	Error	The specified stations do not support concurrent printing, or Page Mode is active for either station specified in stations.
	E_BUSY	Warning	Cannot perform while output is in progress.
	EPTR_COVER_OPEN	Error	The printer cover is open.
	EPTR_JRN_EMPTY	Error	The journal station was specified but is out of paper.
	EPTR_JRN_CARTRIDGE_REMOVED	Error	A journal cartridge has been removed.
	EPTR_JRN_CARTRIDGE_EMPTY	Error	A journal cartridge is empty.
	EPTR_JRN_HEAD_CLEANING	Warning	A journal cartridge head is being cleaned.

	EPTR_REC_EMPTY	Error	The receipt station was specified but is out of paper.
	EPTR_REC_CARTRIDGE_REMOVED	Error	A receipt cartridge has been removed.
	EPTR_REC_CARTRIDGE_EMPTY	Error	A receipt cartridge is empty.
	EPTR_REC_HEAD_CLEANING	Warning	A receipt cartridge head is being cleaned
	EPTR_SLP_EMPTY	Warning	The slip station was specified, but a form is not inserted.
	EPTR_SLP_CARTRIDGE_REMOVED	Error	A slip cartridge has been removed.
	EPTR_SLP_CARTRIDGE_EMPTY	Error	A slip cartridge is empty.
	EPTR_SLP_HEAD_CLEANING	Warning	A slip cartridge head is being cleaned.
rotatePrint			
	E_ILLEGAL	Error	The specified station does not exist, or the station does not support the specified rotation
	E_BUSY	Warning	Cannot perform while output is in progress.
	EPTR_COVER_OPEN	Error	The printer cover is open.
	EPTR_REC_EMPTY	Error	The receipt station was specified but is out of paper.
	EPTR_REC_CARTRIDGE_REMOVED	Error	A receipt cartridge has been removed.
	EPTR_REC_CARTRIDGE_EMPTY	Error	A receipt cartridge is empty.
	EPTR_REC_HEAD_CLEANING	Warning	A receipt cartridge head is being cleaned
	EPTR_SLP_EMPTY	Warning	The slip station was specified, but a form is not inserted.
	EPTR_SLP_CARTRIDGE_REMOVED	Error	A slip cartridge has been removed.
	EPTR_SLP_CARTRIDGE_EMPTY	Error	A slip cartridge is empty.
	EPTR_SLP_HEAD_CLEANING	Warning	A slip cartridge head is being cleaned.
setBitmap			

	E_ILLEGAL	Error	One of the following errors occurred: * bitmapNumber is invalid * station does not exist * station does not support bitmap printing * width is too big * alignment is invalid or too big
	E_NOEXIST	Error	fileName was not found.
	EPTR_TOOBIG	Error	The bitmap is either too wide to print without transformation, or it is too big to transform.
	EPTR_BADFORMAT	Error	The specified file is either not a bitmap file, or it is in an unsupported format.
setLogo			
	E_ILLEGAL	Error	An invalid location was specified
transactionPrint			
	E_ILLEGAL	Error	The specified station does not exist, or **CapTransaction** is false.
	E_BUSY	Warning	Cannot perform while output is in progress.
	EPTR_COVER_OPEN	Error	The printer cover is open.
	EPTR_JRN_EMPTY	Error	The journal station was specified but is out of paper.
	EPTR_JRN_CARTRIDGE_REMOVED	Error	A journal cartridge has been removed.
	EPTR_JRN_CARTRIDGE_EMPTY	Error	A journal cartridge is empty.
	EPTR_JRN_HEAD_CLEANING	Warning	A journal cartridge head is being cleaned.
	EPTR_REC_EMPTY	Error	The receipt station was specified but is out of paper.
	EPTR_REC_CARTRIDGE_REMOVED	Error	A receipt cartridge has been removed.
	EPTR_REC_CARTRIDGE_EMPTY	Error	A receipt cartridge is empty.
	EPTR_REC_HEAD_CLEANING	Warning	A receipt cartridge head is being cleaned
	EPTR_SLP_EMPTY	Error	The slip station was specified, but a form is not inserted.
	EPTR_SLP_CARTRIDGE_REMOVED	Error	A slip cartridge has been removed.
	EPTR_SLP_CARTRIDGE_EMPTY	Error	A slip cartridge is empty.
	EPTR_SLP_HEAD_CLEANING	Warning	A slip cartridge head is being cleaned.
validateData			

	E_ILLEGAL	Error	Some of the data is not precisely supported by the printer station, but the Service can select valid alternatives. This exception can also be thrown if an escape sequence is not supported while either Page Mode or rotate sideways is active.
	E_FAILURE	Error	Some of the data is not supported. No alternatives can be selected.

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
PTR_SUE_COVER_OPEN	Error	Printer cover is open.
PTR_SUE_COVER_OK	Error	Printer cover is closed.
PTR_SUE_JRN_EMPTY	Error	No journal paper
PTR_SUE_JRN_NEAREMPTY	Warning	Journal paper is low
PTR_SUE_JRN_PAPEROK	Information	Journal paper is ready
PTR_SUE_REC_EMPTY	Error	No receipt paper.
PTR_SUE_REC_NEAREMPTY	Warning	Receipt paper is low
PTR_SUE_REC_PAPEROK	Information	Receipt paper is ready.
PTR_SUE_SLP_EMPTY	Error	No slip form is inserted, and no slip form has been detected at the entrance to the slip station.
PTR_SUE_SLP_NEAREMPTY	Warning	Almost at the bottom of the slip form.
PTR_SUE_SLP_PAPEROK	Information	Slip form is inserted
PTR_SUE_IDLE	Information	All asynchronous output has finished, either successfully or because output has been cleared.
PTR_SUE_JRN_CARTRIDGE_EMPTY	Warning	A journal cartridge needs to be replaced. Cartridge is empty or not present.
PTR_SUE_JRN_HEAD_CLEANING	Information	A journal cartridge has begun cleaning.
PTR_SUE_JRN_CARTRIDGE_NEAREMPTY	Warning	A journal cartridge is near end
PTR_SUE_JRN_CARTRIDGE_OK	Information	All journal cartridges are ready. It gives no indication of the amount of media in the cartridge
PTR_SUE_REC_CARTRIDGE_EMPTY	Warning	A receipt cartridge needs to be replaced. Cartridge is empty or not present.
PTR_SUE_REC_HEAD_CLEANING	Information	A receipt cartridge has begun cleaning.
PTR_SUE_REC_CARTRIDGE_NEAREMPTY	Warning	A receipt cartridge is near end.

PTR_SUE_REC_CARTRIDGE_OK	Information	All receipt cartridges are ready. It gives no indication of the amount of media in the cartridge
PTR_SUE_SLP_CARTRIDGE_EMPTY	Warning	A slip cartridge needs to be replaced. Cartridge is empty or not present
PTR_SUE_SLP_HEAD_CLEANING	Information	A slip cartridge has begun cleaning.
PTR_SUE_SLP_CARTRIDGE_NEAREMPTY	Warning	A slip cartridge is near end.
PTR_SUE_SLP_CARTRIDGE_OK	Information	All slip cartridges are ready. It gives no indication of the amount of media in the cartridge.
PTR_SUE_JRN_COVER_OPEN	Error	Journal station cover is open
PTR_SUE_JRN_COVER_OK	Information	Journal station cover is closed
PTR_SUE_REC_COVER_OPEN	Error	Receipt station cover is open.
PTR_SUE_REC_COVER_OK	Information	Receipt station cover is closed
PTR_SUE_SLP_COVER_OPEN	Error	Slip station cover is open
PTR_SUE_SLP_COVER_OK	Information	Slip station cover is closed.

RFID Scanner

RFID Scanner Example

Retrieve CapMultipleProtocols Property Response

```
<?xml version="1.0" encoding="UTF-8"?>
<RFIDScanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/RFIDScanner/
        RFIDScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        RFIDScanner/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Response" ActionCode="Read">
            <MessageID>1234</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <Response ResponseCode="OK">
                    <RequestID>98765</RequestID>
                    <ResponseTimestamp>2001-12-17T09:30:47.0Z
                    </ResponseTimestamp>
            </Response>
            <SensorID UnifiedPOS="RFIDScanner">String</SensorID>
    </ARTSHeader>
    <RFIDScannerBody>
            <GetProperty>
                    <CapMultipleProtocols>RFID_CMP_EPC0
                    </CapMultipleProtocols>
                    <CapMultipleProtocols>RFID_CMP_0PLUS
                    </CapMultipleProtocols>
            </GetProperty>
    </RFIDScannerBody>
</RFIDScanner>
```

Set ProtocolMask Property

```
<?xml version="1.0" encoding="UTF-8"?>
<RFIDScanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/RFIDScanner/
        RFIDScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        RFIDScanner/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish" ActionCode="Update">
            <SensorID UnifiedPOS="RFIDScanner">POS1Scanner</SensorID>
    </ARTSHeader>
    <RFIDScannerBody>
            <SetProperty>
                    <ProtocolMask>RFID_SDT_EPC0</ProtocolMask>
            </SetProperty>
    </RFIDScannerBody>
</RFIDScanner>
```

RFID Scanner Data Event

```
<?xml version="1.0" encoding="UTF-8"?>
<RFIDScannerEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        RFIDScannerEvents/ RFIDScannerEventV1.14.1.xsd" xmlns="http://
        www.nrf-arts.org/UnifiedPOS/RFIDScannerEvents/" MajorVersion="1"
        MinorVersion="14" FixVersion="1" ActionCode="Begin">
    <DataEvent>
            <SequenceNumber>4294967295</SequenceNumber>
            <EventDateTime>2001-12-17T09:30:47.0Z</EventDateTime>
            <SourceName>POS1Scanner</SourceName>
            <SensorID UnifiedPOS="RFIDScanner">POS1Scanner</SensorID>
            <Status>0</Status>
            <Properties>
                    <CheckHealthText>String</CheckHealthText>
                    <DataCount>0</DataCount>
                    <CurrentTagID>\u005b\u0029\u003e\u001e\u0030\u0000
                     </CurrentTagID>
                    <CurrentTagIDLength>0</CurrentTagIDLength>
                    <CurrentTagProtocol>0</CurrentTagProtocol>
                    <CurrentTagUserData>
                    \u005b\u0029\u003e\u001e\u0030\u0000\u001d\u0030
                    \u0030\u0030\u0031\u001d\u0032\u001d\u0033
                    \u001d\u0031\u0032\u0033
                    </CurrentTagUserData>
            </Properties>
    </DataEvent>
</RFIDScannerEvent>
```

Read Tags Request

```
<?xml version="1.0" encoding="UTF-8"?>
<RFIDScanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/RFIDScanner/
        RFIDScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        RFIDScanner/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish" ActionCode="Initiate">
            <SensorID UnifiedPOS="RFIDScanner">POS1Scanner</SensorID>
    </ARTSHeader>
    <RFIDScannerBody>

            <ReadTags>
                    <Cmd>RFID_RT_ID</Cmd>
                    <FilterID>
                            \u005b\u0029\u003e\u001e\u0030\u0000
                    </FilterID>
                    <FilterMask>
                            \u0029\u003e\u001e\u0030\u0000\u005b
                    </FilterMask>
                    <Start>0</Start>
                    <Length>0</Length>
                    <Timeout>0</Timeout>
                    <Password>
                            \u005b\u0029\u003e\u001e\u0030\u0000
                    </Password>
            </ReadTags>
    </RFIDScannerBody>
</RFIDScanner>
```

Write Tags Request

```
<?xml version="1.0" encoding="UTF-8"?>
<RFIDScanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/RFIDScanner/
        RFIDScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        RFIDScanner/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request" ActionCode="Begin">
            <DateTime TypeCode="Message">2001-12-17T09:30:47.0Z
            </DateTime>
            <SensorID UnifiedPOS="RFIDScanner">POS1Scanner
            </SensorID>
    </ARTSHeader>
    <RFIDScannerBody>
            <WriteTagData>
                    <TagID>
                            \u0029\u003e\u001e\u0030\u0000\u005b
                    </TagID>
                    <UserData>
                            \u005b\u0029\u003e\u001e\u0030\u0000
                    </UserData>
                    <Start>0</Start>
                    <Timeout>0</Timeout>
                    <Password>
                            \u022b\u0029\u003e\u001e\u0030\u0000
                    </Password>
            </WriteTagData>
    </RFIDScannerBody>
</RFIDScanner>
```

RFID Scanner Domain

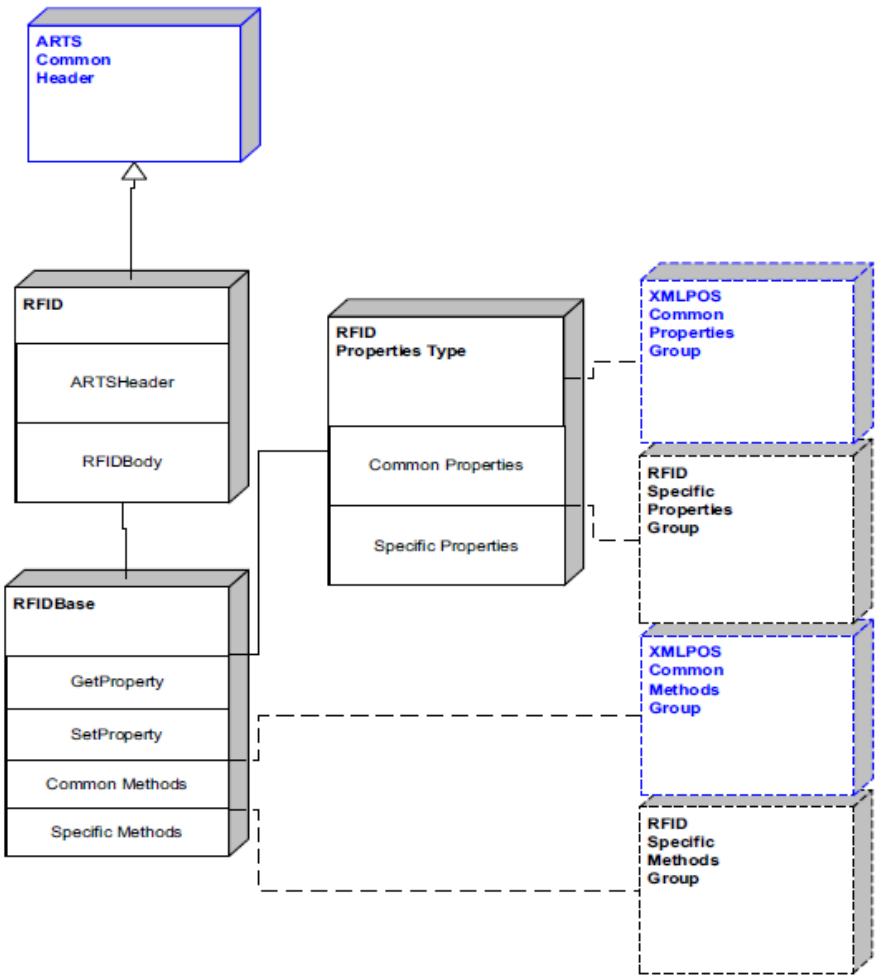


Figure 131: RFID Scanner Domain View

RFID Scanner Properties

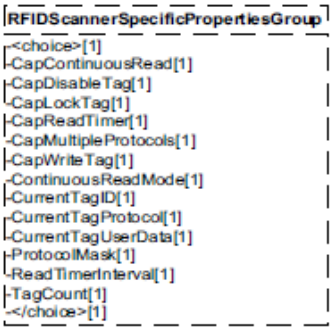
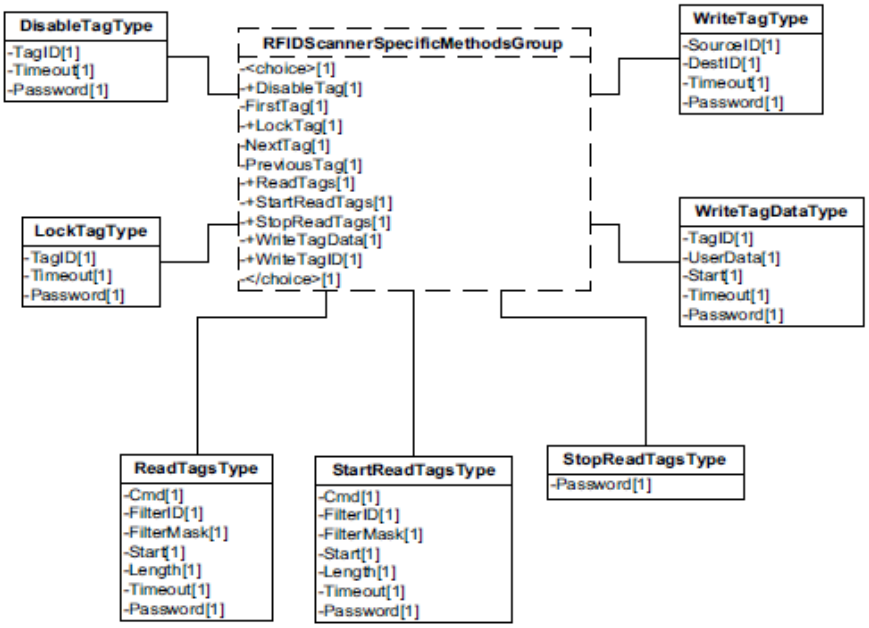


Figure 132: RFID Scanner Properties Domain View



RFID Scanner Methods

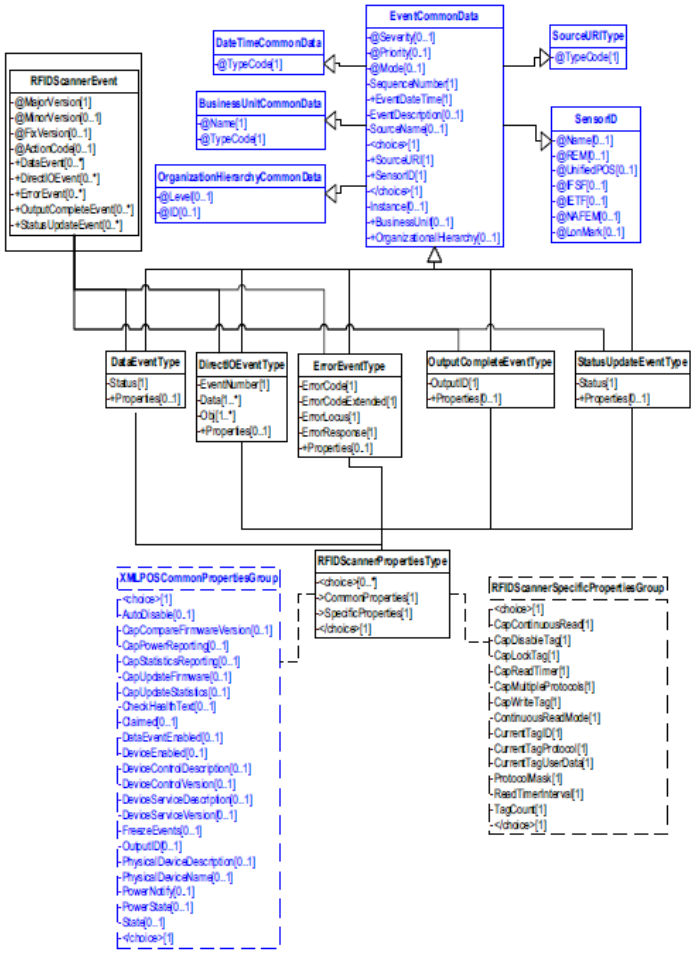


Figure 133: RFID Scanner Methods Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
disableTag			
	E_TIMEOUT	Error	Allowed execution time has expired.

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
N/A		

Scanner Device

Scanner Device Example

Application sets DecodeData = True

```
<?xml version="1.0" encoding="UTF-8"?>
<Scanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/Scanner/
        ScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/Scanner/
        " MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish" ActionCode="Update">
            <MessageID>1</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="Scanner">Scanner1</SensorID>
    </ARTSHeader>
    <ScannerBody>
            <SetProperty>
                <DecodeData>true</DecodeData>
            </SetProperty>
<    /ScannerBody>
</Scanner>
```

Application sets DataEventEnabled = True

```
<?xml version="1.0" encoding="UTF-8"?>
<Scanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/Scanner/
        ScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/Scanner/
        " MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish" ActionCode="Update">
            <MessageID>2</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="Scanner">Scanner1</SensorID>
    </ARTSHeader>
    <ScannerBody>
            <SetProperty>
                <DataEventEnabled>false</DataEventEnabled>
            </SetProperty>
    </ScannerBody>
</Scanner>
```

User scans bar code with data "5000174289657<CR>"

Scanner decodes bar code, sends data to scanner service

Scanner service sets ScanData property = "5000174289657"

Scanner service sets ScanDataType = "SCAN_SDT_EAN13"

Scanner service notifies application (through control) of asynchronous data event

```
<?xml version="1.0" encoding="UTF-8"?>
<ScannerEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/ScannerEvents/
        ScannerEventV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        ScannerEvents/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <DataEvent>
            <SequenceNumber>3</SequenceNumber>
            <EventDateTime>2001-12-17T09:30:47.0Z</EventDateTime>
            <SensorID UnifiedPOS="Scanner">Scanner1</SensorID>
            <Status>0</Status>
    </DataEvent>
</ScannerEvent>
```

Application services event (reads ScanData or ScanDataLabel, and ScanDataType)

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<Scanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/Scanner/
        ScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/Scanner/
        " MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request" ActionCode="Read">
            <MessageID>4</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="Scanner">Scanner1</SensorID>
    </ARTSHeader>
    <ScannerBody>
            <GetProperty>
                    <ScanData/>
                    <ScanDataLabel/>
            </GetProperty>
    </ScannerBody>
</Scanner>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Scanner xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/Scanner/
        ScannerV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/Scanner/
        " MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Response" ActionCode="Read">
            <MessageID>4</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="Scanner">Scanner1</SensorID>
    </ARTSHeader>
    <ScannerBody>
            <GetProperty>
                    <ScanData>\u005b\u0029\u003e\u001e\u0030\u0000
                    </ScanData>
                    <ScanDataType>SCAN_SDT_EAN13</ScanDataType>
            </GetProperty>
    </ScannerBody>
</Scanner>
```

Scanner Domain

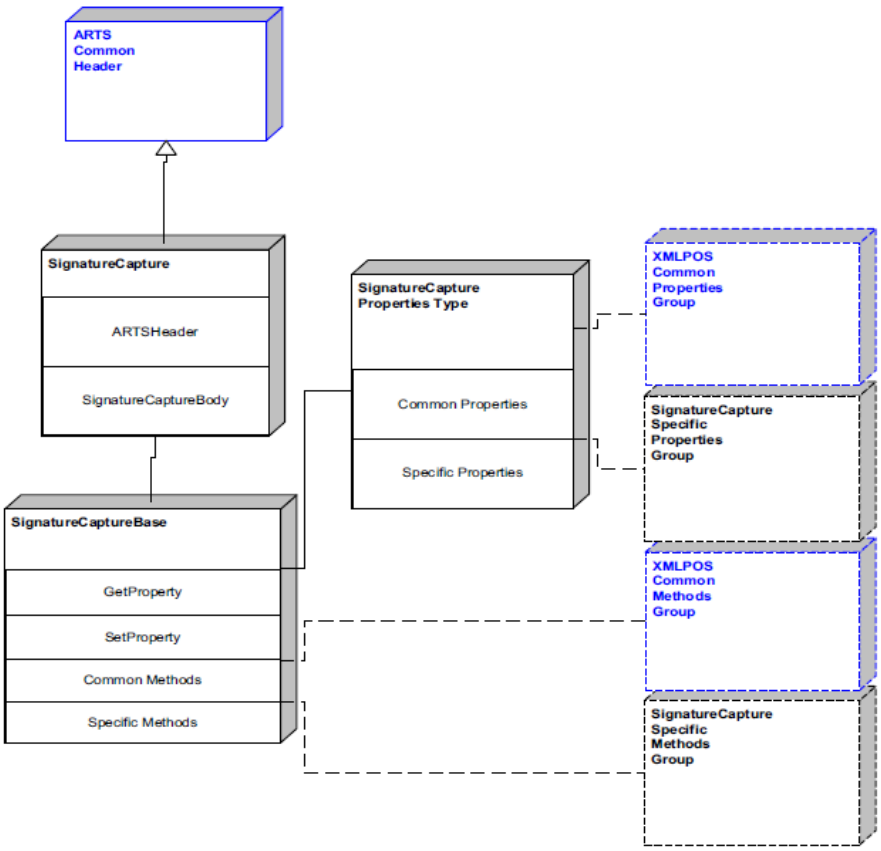


Figure 139: Scanner Domain View

Scanner Properties

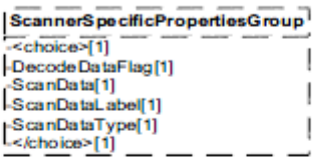


Figure 140: Scanner Properties Domain View

Scanner Methods

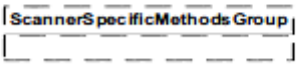
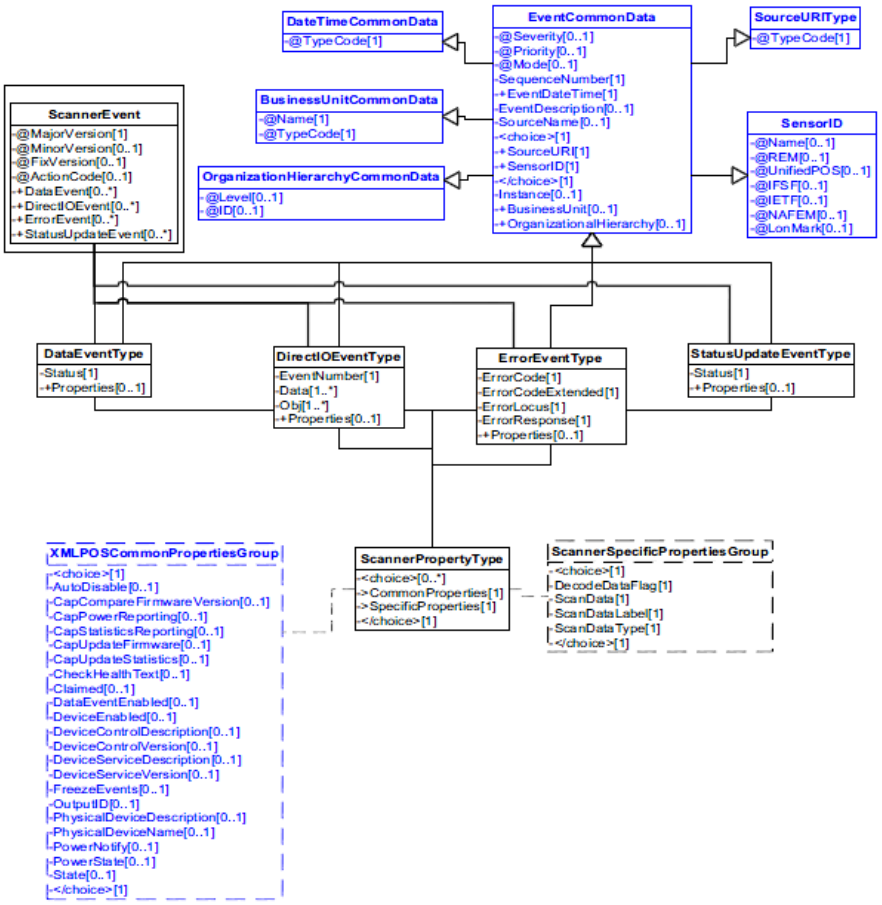


Figure 141: Scanner Methods Domain View

Scanner Events



Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
N/A			

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
N/A		

Smart Card Reader

Smart Card Reader Example

→beginInsertion

```
<?xml version="1.0" encoding="UTF-8"?>
<SmartCardRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/SmartCardRW/
        SmartCardRWV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        SmartCardRW/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>1</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="SmartCardRW">100</SensorID>
    </ARTSHeader>
    <SmartCardRWBody>
            <BeginInsertion>
                    <Timeout>30000</Timeout>
            </BeginInsertion>
    </SmartCardRWBody>
</SmartCardRW>
```

→endInsertion

```
<?xml version="1.0" encoding="UTF-8"?>
<SmartCardRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/SmartCardRW/
        SmartCardRWV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        SmartCardRW/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>2</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="SmartCardRW">2</SensorID>
    </ARTSHeader>
    <SmartCardRWBody>
            <EndInsertion/>
    </SmartCardRWBody>
</SmartCardRW>
```

←fire DataEvent

```
<?xml version="1.0" encoding="UTF-8"?>
<SmartCardRWEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/
        SmartCardEvents/ SmartCardRWEventV1.14.1.xsd" xmlns="http://
        www.nrf-arts.org/UnifiedPOS/SmartCardEvents/" MajorVersion="1"
        MinorVersion="14" FixVersion="1">
    <DataEvent Severity="Information" Mode="Production" Priority="-0">
            <SequenceNumber>3</SequenceNumber>
            <EventDateTime>2001-12-17T09:30:47.0Z</EventDateTime>
            <SensorID UnifiedPOS="SmartCardRW">100</SensorID>
            <Status>0</Status>
    </DataEvent>
</SmartCardRWEvent>
```

→readData

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SmartCardRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/SmartCardRW/
        SmartCardRWV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        SmartCardRW/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Request">
            <MessageID>4</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="SmartCardRW">100</SensorID>
    </ARTSHeader>
    <SmartCardRWBody>
            <ReadData>
                    <Action>SC_READ_DATA</Action>
                    <Count/>
                    <Data/>
            </ReadData>
    </SmartCardRWBody>
</SmartCardRW>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SmartCardRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/SmartCardRW/
        SmartCardRWV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        SmartCardRW/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Response">

            <MessageID>4</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="SmartCardRW">100</SensorID>
     </ARTSHeader>
     <SmartCardRWBody>
            <ReadData>
                    <Action>SC_READ_DATA</Action>
                    <Count>10</Count>
                    <Data>1234568790</Data>
            </ReadData>
     </SmartCardRWBody>
</SmartCardRW>
```

→beginRemoval

```
<?xml version="1.0" encoding="UTF-8"?>
<SmartCardRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/SmartCardRW/
        SmartCardRWV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        SmartCardRW/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>5</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="SmartCardRW">100</SensorID>
    </ARTSHeader>
    <SmartCardRWBody>
            <BeginRemoval>
                    <Timeout>30000</Timeout>
            </BeginRemoval>
    </SmartCardRWBody>
</SmartCardRW>
```

→endRemoval

```
<?xml version="1.0" encoding="UTF-8"?>
<SmartCardRW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/SmartCardRW/
        SmartCardRWV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        SmartCardRW/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>6</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="SmartCardRW">100</SensorID>
    </ARTSHeader>
    <SmartCardRWBody>
            <EndRemoval/>
    </SmartCardRWBody>
</SmartCardRW>
```

Smart Card Reader Domain

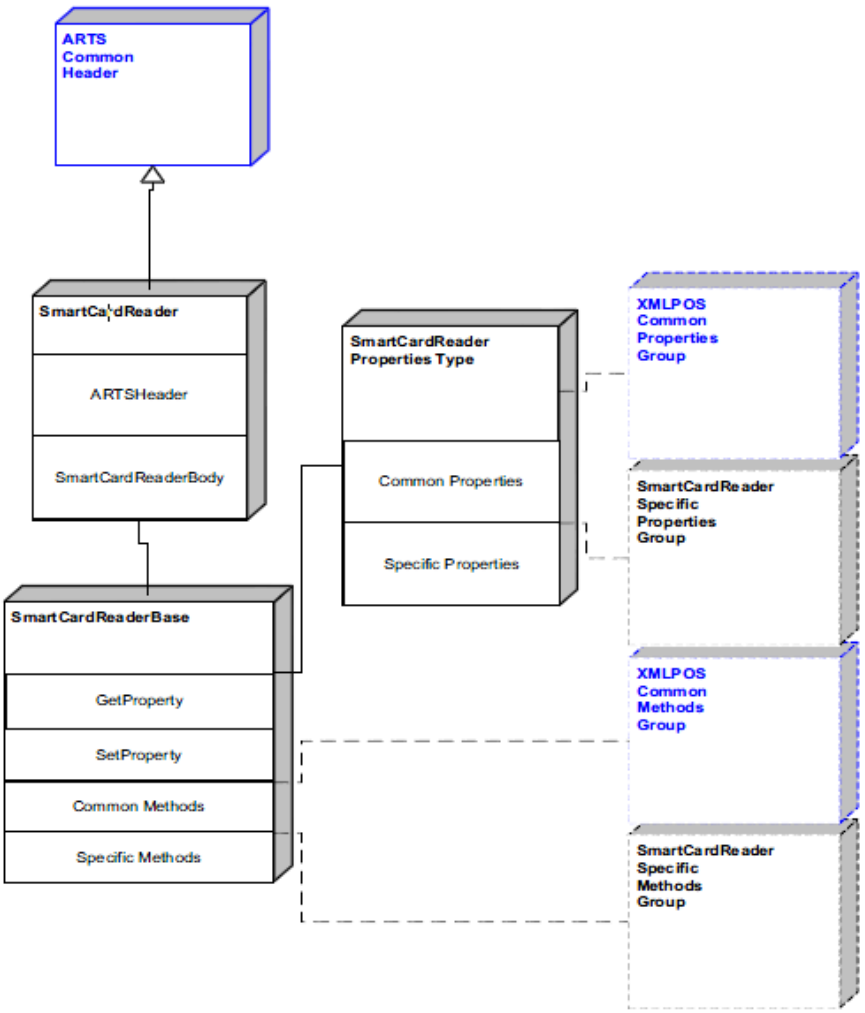


Figure 147: Smart Card Reader Domain View

Smart Card Reader Properties

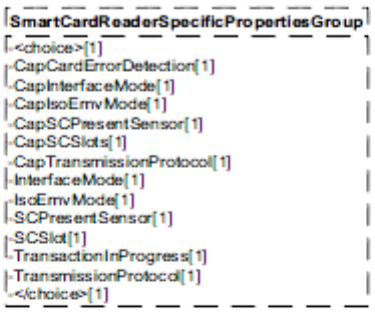


Figure 148: Smart Card Reader Properties Domain View

Smart Card Reader Methods

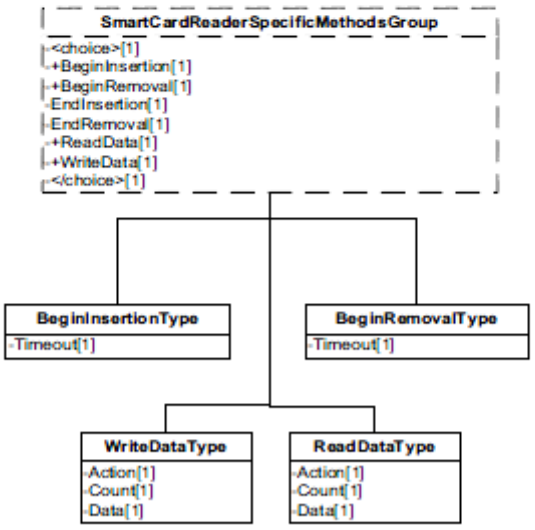
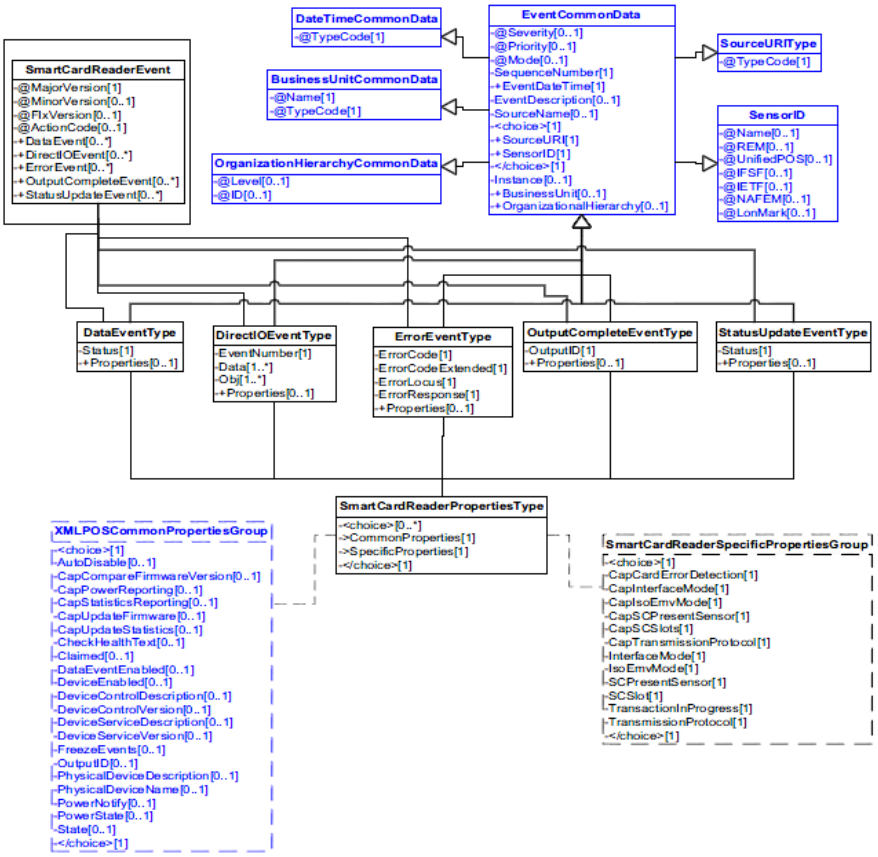


Figure 149: Smart Card Reader Methods Domain View

Smart Card Reader Events



Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
beginInsertion			
	E_BUSY	Warning	This operation cannot be performed because asynchronous output is in progress.
	E_ILLEGAL	Error	The SCR/W does not exist or an invalid timeout parameter was specified.
	E_TIMEOUT	Error	The specified time has elapsed without the smart card being properly inserted.
beginRemoval			
	E_BUSY	Warning	This operation cannot be performed because asynchronous output is in progress.
	E_ILLEGAL	Error	The SCR/W does not exist or an invalid timeout parameter was specified.
	E_TIMEOUT	Warning	The specified time has elapsed without the smart card being properly inserted.
endInsertion			
	E_ILLEGAL	Warning	The SCR/W is not in smart card insertion mode.
	E_FAILURE	Warning	A card is not inserted in the SCR/W.
endRemoval			
	E_ILLEGAL	Warning	The SCR/W is not in smart card removal mode.
	E_FAILURE	Warning	There is a card in the SCR/W.
readData			
	E_CLAIMED	Warning	Cannot read because the smart card present in the SCR/W is claimed by another application.
	E_ILLEGAL	Error	The action is not valid for the type of smart card present in the SCR/W or the count value is not valid for the smart card present in the SCR/W.
writeData			
	E_CLAIMED	Warning	Cannot read because the smart card present in the SCR/W is claimed by another application.

	E_ILLEGAL	Error	The action is not valid for the type of smart card present in the SCR/W or the count value is not valid for the smart card present in the SCR/W.
E_EXTENDED			
	ESC_READ	Error	There was a read error
	ESC_WRITE	Error	There was a write error
	ESC_TORN	Warning	The smart card was prematurely removed from the SCR/W unexpectedly.
	ESC_NO_CARD	Warning	There is no card detected in the SCR/W but a card was expected

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
SC_SUE_NO_CARD	Warning	No card detected in the SCR/W Device.
SC_SUE_CARD_PRESENT	Information	There is a card in the device.

Tone Indicator

Tone Indicator Example

→ set Tone1Frequency

```
<?xml version="1.0" encoding="UTF-8"?>
<ToneIndicator xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/ToneIndicator/
        ToneIndicatorV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        ToneIndicator/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish" ActionCode="Begin">
            <MessageID>1</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="ToneIndicator">Buzzer1</SensorID>
    </ARTSHeader>
    <ToneIndicatorBody>
            <SetProperty>
                <Tone1Pitch>1000</Tone1Pitch>
            </SetProperty>
    </ToneIndicatorBody>
</ToneIndicator>
```

→ Sound(3, 100)

```
<?xml version="1.0" encoding="UTF-8"?>
<ToneIndicator xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.nrf-arts.org/UnifiedPOS/ToneIndicator/
        ToneIndicatorV1.14.1.xsd" xmlns="http://www.nrf-arts.org/UnifiedPOS/
        ToneIndicator/" MajorVersion="1" MinorVersion="14" FixVersion="1">
    <ARTSHeader MessageType="Publish">
            <MessageID>2</MessageID>
            <DateTime>2001-12-17T09:30:47.0Z</DateTime>
            <SensorID UnifiedPOS="ToneIndicator">Buzzer1</SensorID>
    </ARTSHeader>
    <ToneIndicatorBody>
            <Sound>
                <NumberOfCycles>3</NumberOfCycles>
                <InterSoundWait>100</InterSoundWait>
            </Sound>
    </ToneIndicatorBody>
</ToneIndicator>
```

Tone Indicator Domain

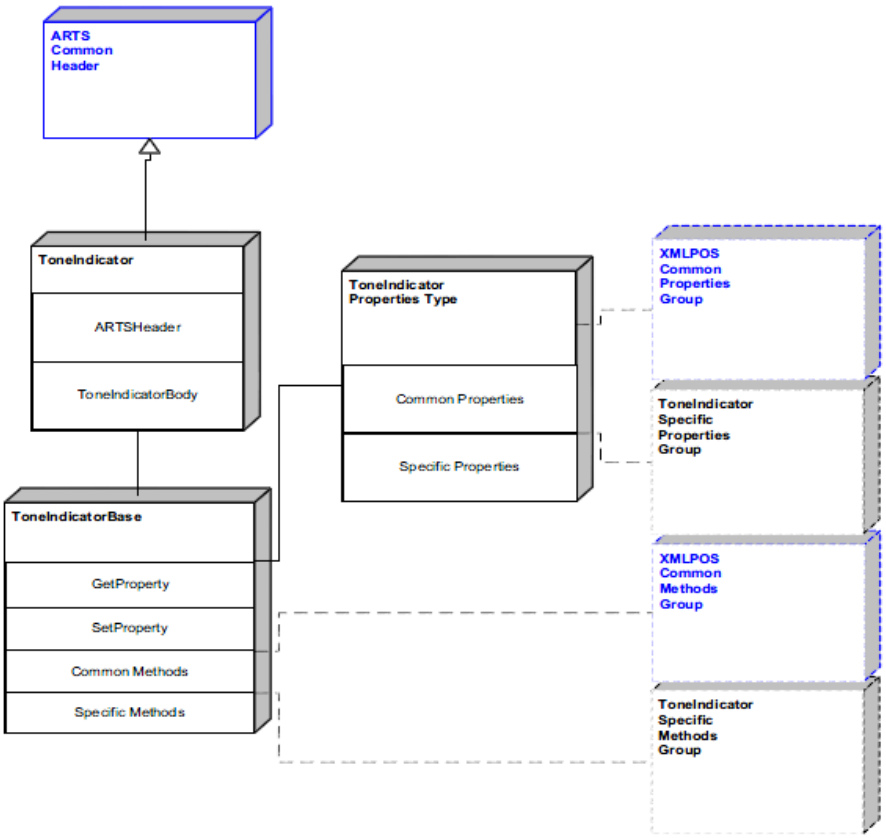


Figure 151: Tone Indicator Domain View

Tone Indicator Properties



Figure 152: Tone Indicator Properties Domain View

Tone Indicator Methods

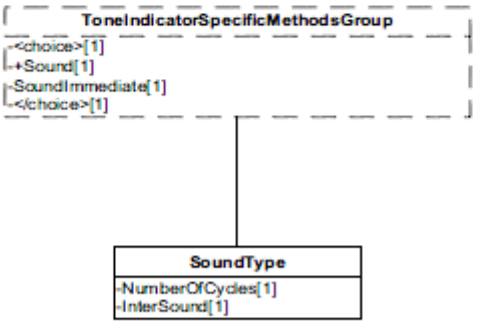


Figure 153: Tone Indicator Methods Domain View

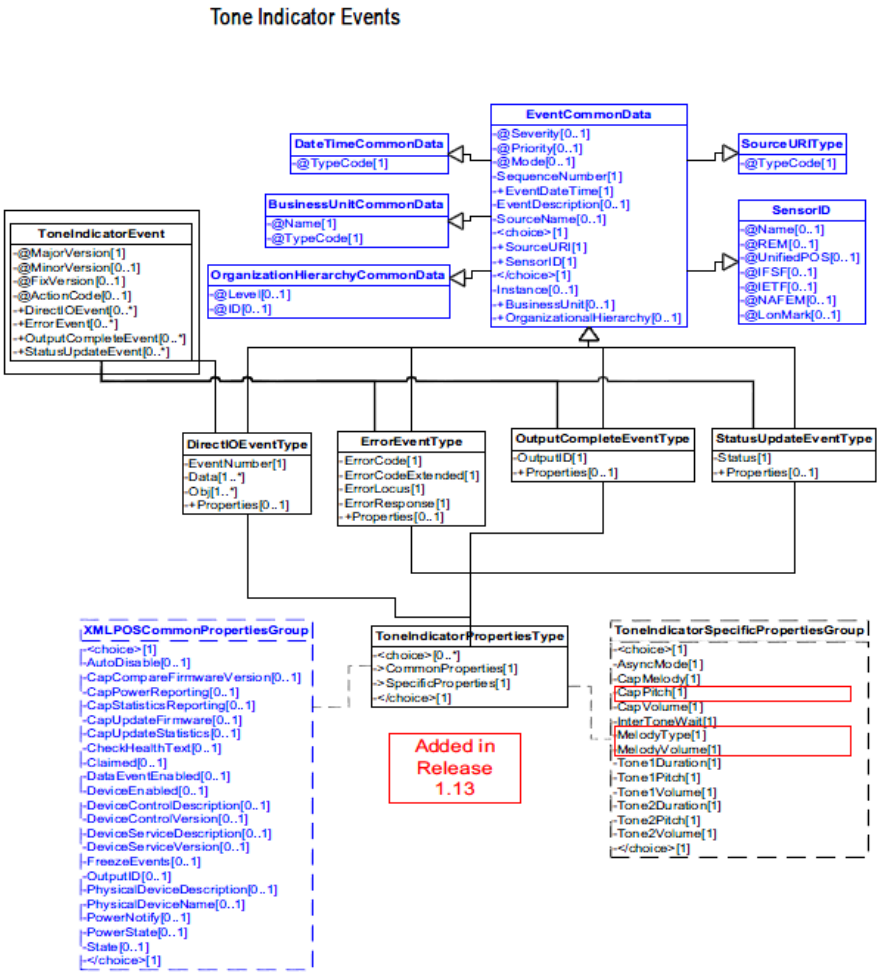


Figure 154: Tone Indicator Events Domain View

Device Error Codes to Message Severity Codes

This table is for device specific error codes. The common errors are defined in Device Error Codes and Message Severity Codes on page 10.

Method	Value	Severity	Meaning
sound			
	E_CLAIMED	Warning	Indicates that another application has claimed the device and has taken over the tone device causing the sound from this method to be interrupted
	E_ILLEGAL	Error	One of the following errors occurred: • *numberOfCycles* is neither a positive, non-zero value nor FOREVER. • *numberOfCycles* is FOREVER when **AsyncMode** is false. • A negative *interSoundWait* was specified • A negative *interToneWait* was specified

Status Codes to Message Severity Codes

This table is for device specific status codes. The common status codes are defined in Standard Status Codes to Severity Codes on page 12.

Device Specific Status Messages

Value	Severity	Meaning
N/A		

NAFEM Protocol

The XMLPOS Common Data components are used in the ProCon interface to National Association of Food Equipment Manufacturers (NAFEM) Hardware.

Administration Enterprise Group

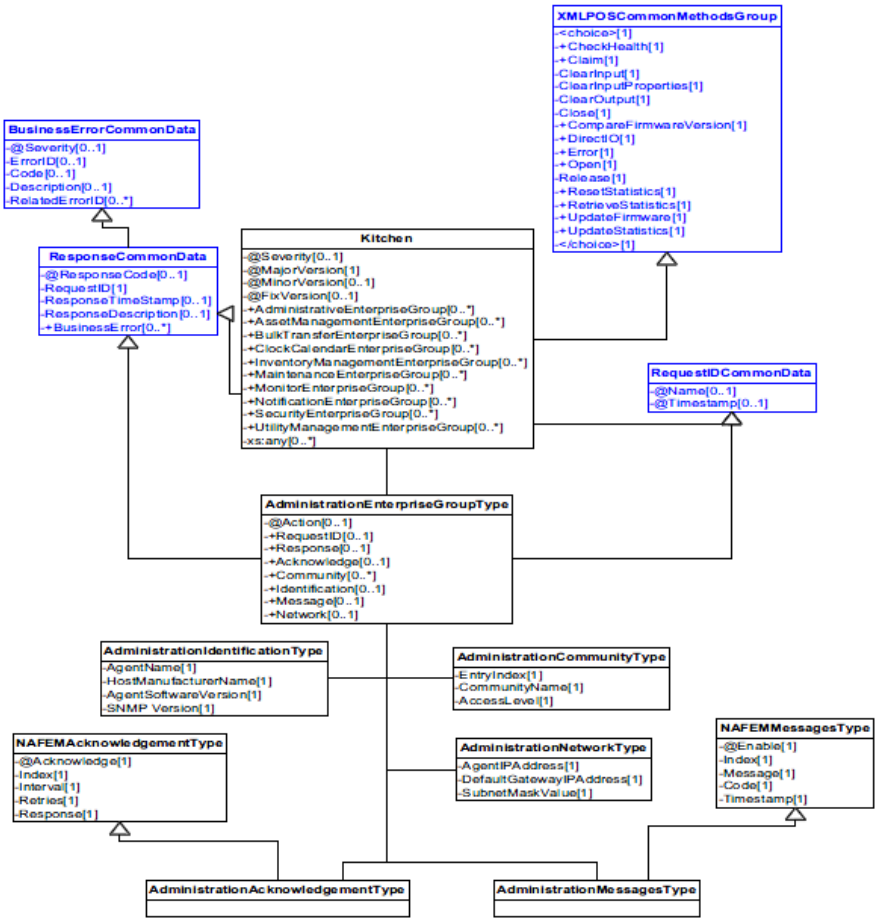


Figure 155: Administrative Enterprise Group Domain View

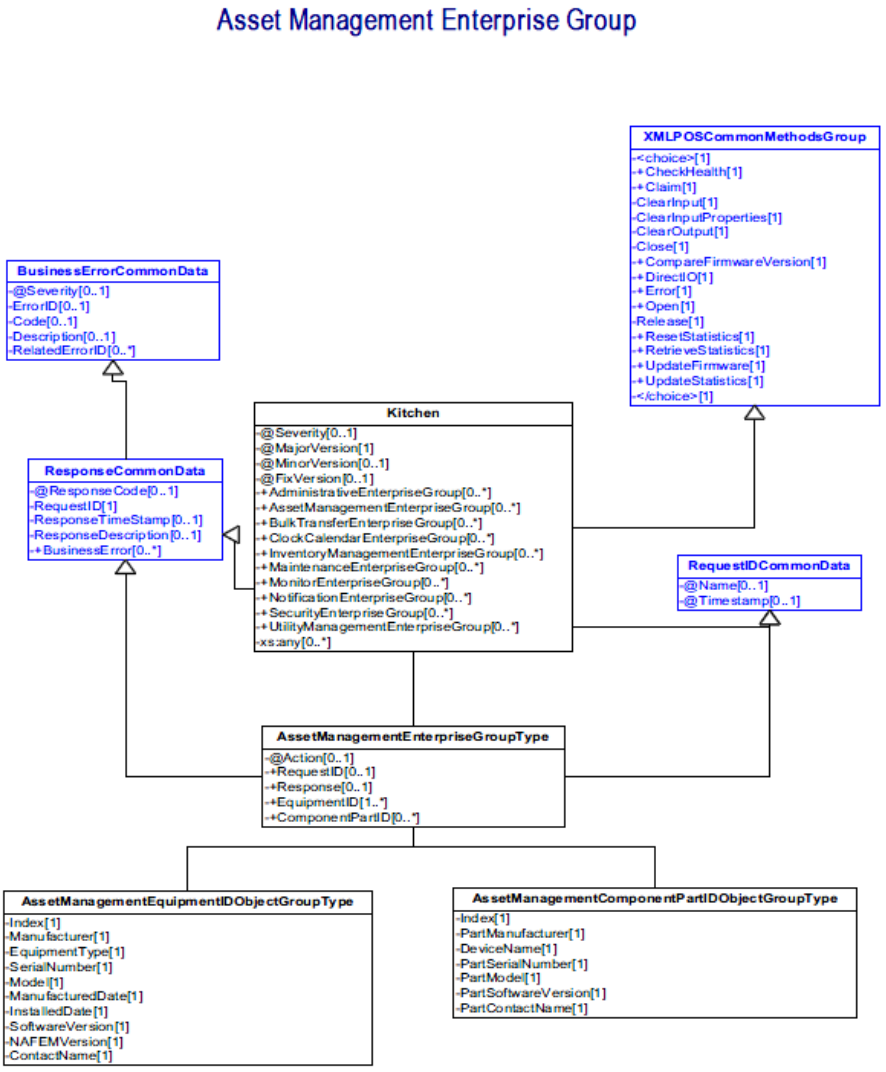


Figure 156: Asset Management Enterprise Group Domain View

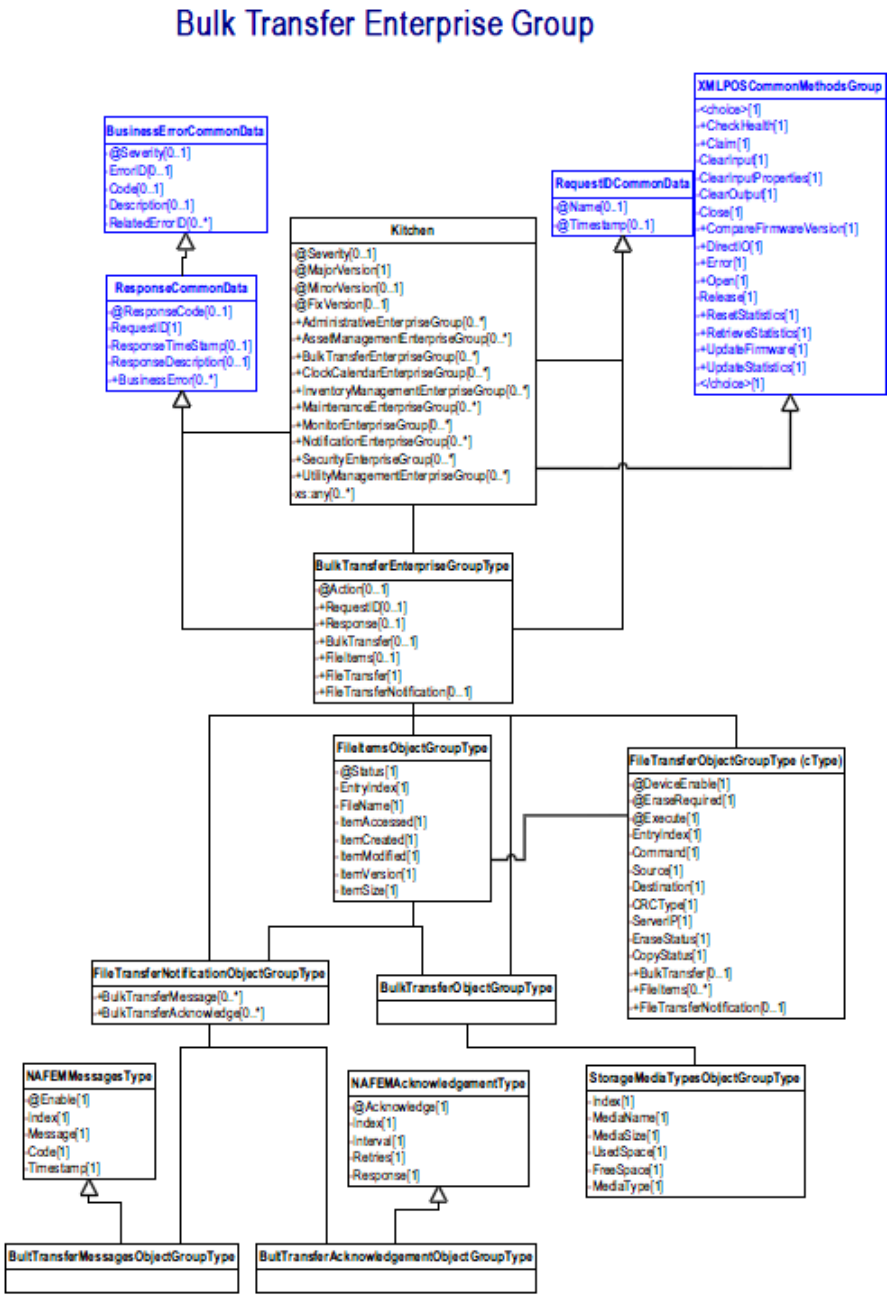
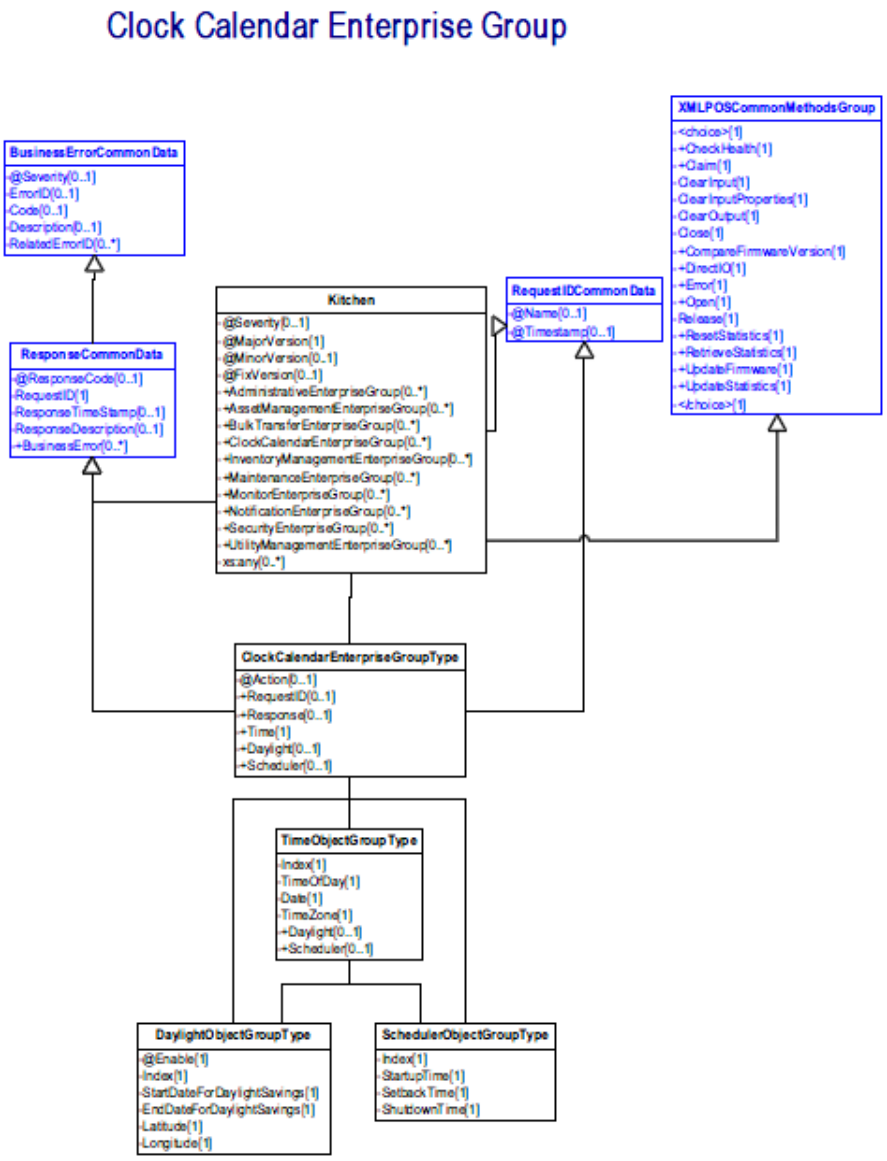


Figure 157: Bulk Transfer Enterprise Group Domain View


Clock Calendar Enterprise Group
BusinessErrorCommonData
-@Severity[0..1]
-ErrorID[0..1]
-Code[0..1]
-Description[0..1]
-RelatedErrorID[0..*]
XMLPOSCommonMethodsGroup
-<choice>[1]
-+CheckHealth[1]
-+Claim[1]
-ClearInput[1]
-ClearInputProperties[1]
-ClearOutput[1]
-Close[1]
-+CompareFirmwareVersion[1]
-+DirectIO[1]
-+Error[1]
-+Open[1]
-Release[1]
-+ResetStatistics[1]
-+RetrieveStatistics[1]
-+UpdateFirmware[1]
-+UpdateStatistics[1]
-</choice>[1]
Kitchen
-@Severity[0..1]
-@MajorVersion[1]
-@MinorVersion[0..1]
-@FixVersion[0..1]
-+AdministrativeEnterpriseGroup[0..*]
-+AssetManagementEnterpriseGroup[0..*]
-+BulkTransferEnterpriseGroup[0..*]
-+ClockCalendarEnterpriseGroup[0..*]
-+InventoryManagementEnterpriseGroup[0..*]
-+MaintenanceEnterpriseGroup[0..*]
-+MonitorEnterpriseGroup[0..*]
-+NotificationEnterpriseGroup[0..*]
-+SecurityEnterpriseGroup[0..*]
-+UtilityManagementEnterpriseGroup[0..*]
-xs:any[0..*]
RequestIDCommonData
-@Name[0..1]
-@Timestamp[0..1]
ResponseCommonData
-@ResponseCode[0..1]
-RequestID[1]
-ResponseTimeStamp[0..1]
-ResponseDescription[0..1]
-+BusinessError[0..*]
ClockCalendarEnterpriseGroupType
-@Action[0..1]
-+RequestID[0..1]
-+Response[0..1]
-+Time[1]
-+Daylight[0..1]
-+Scheduler[0..1]
TimeObjectGroupType
-Index[1]
-TimeOfDay[1]
-Date[1]
-TimeZone[1]
-+Daylight[0..1]
-+Scheduler[0..1]
DaylightObjectGroupType
-@Enable[1]
-Index[1]
-StartDateForDaylightSavings[1]
-EndDateForDaylightSavings[1]
-Latitude[1]
-Longitude[1]
SchedulerObjectGroupType
-Index[1]
-StartupTime[1]
-SetbackTime[1]
-ShutdownTime[1]

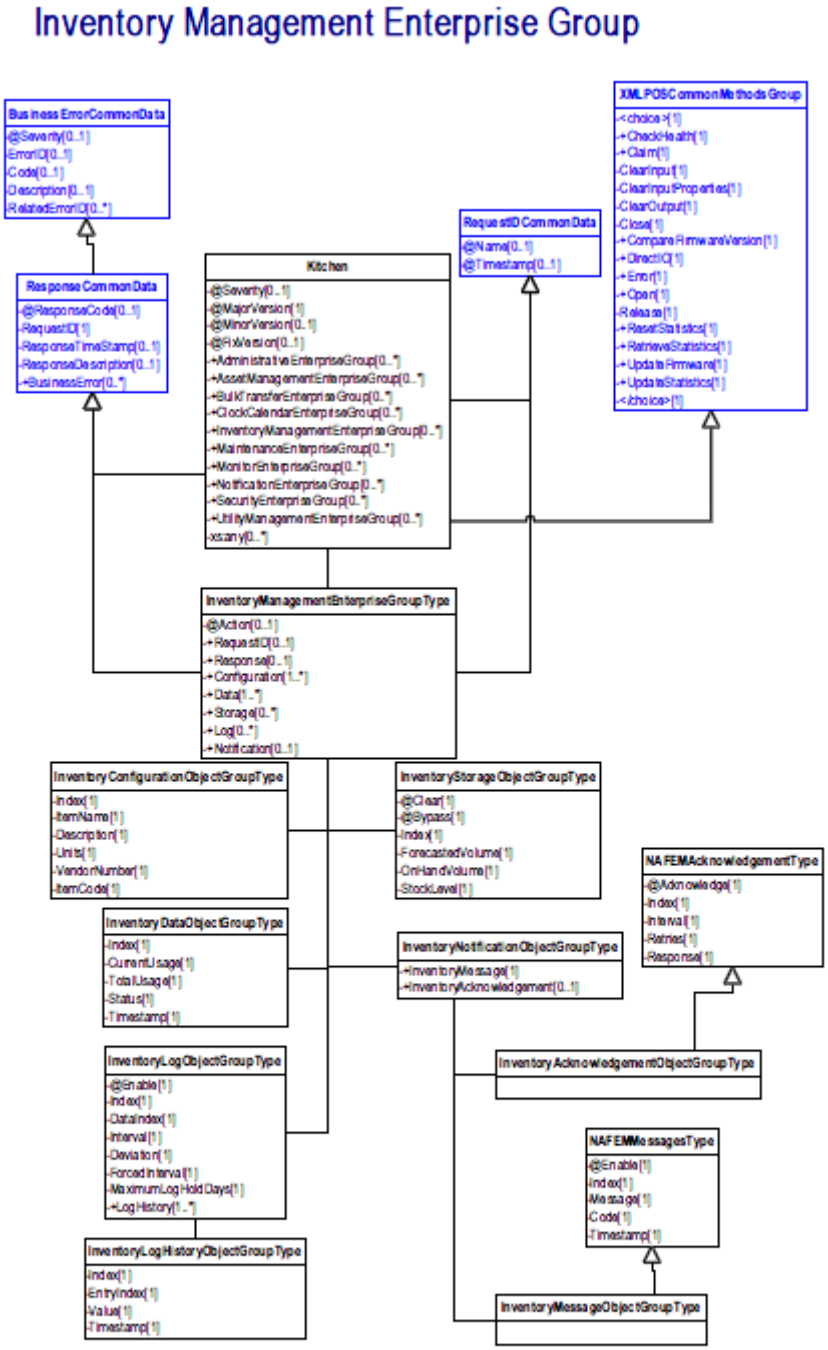


Figure 159: Inventory Management Enterprise Group Domain View

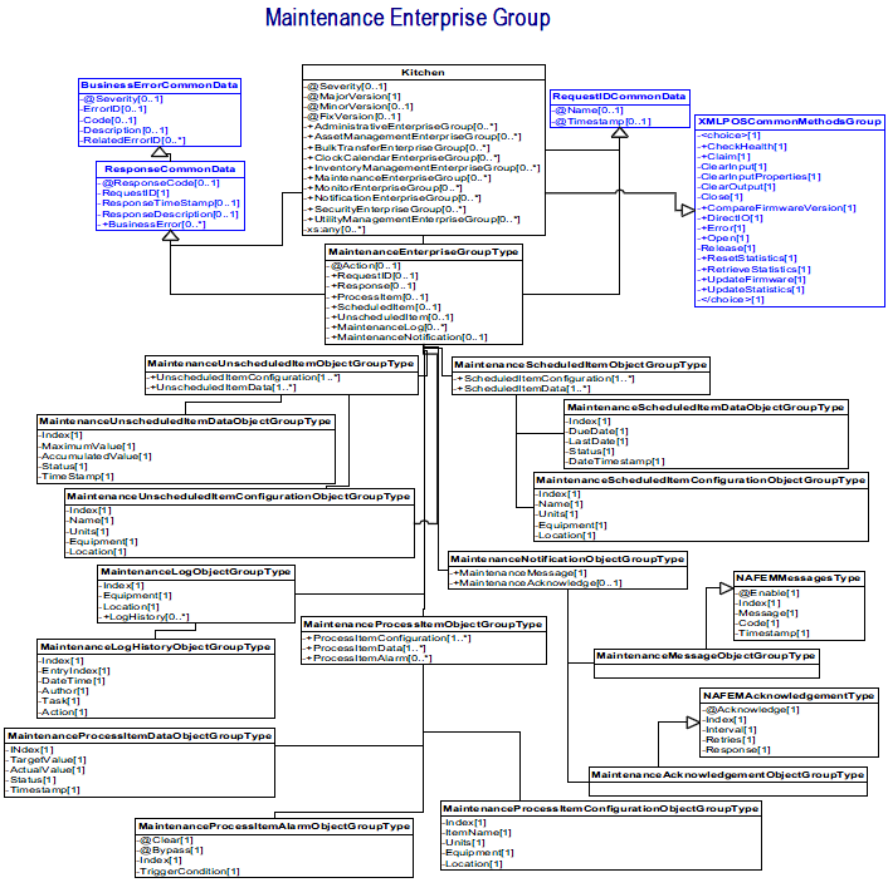


Figure 160: Maintenance Enterprise Group Domain View

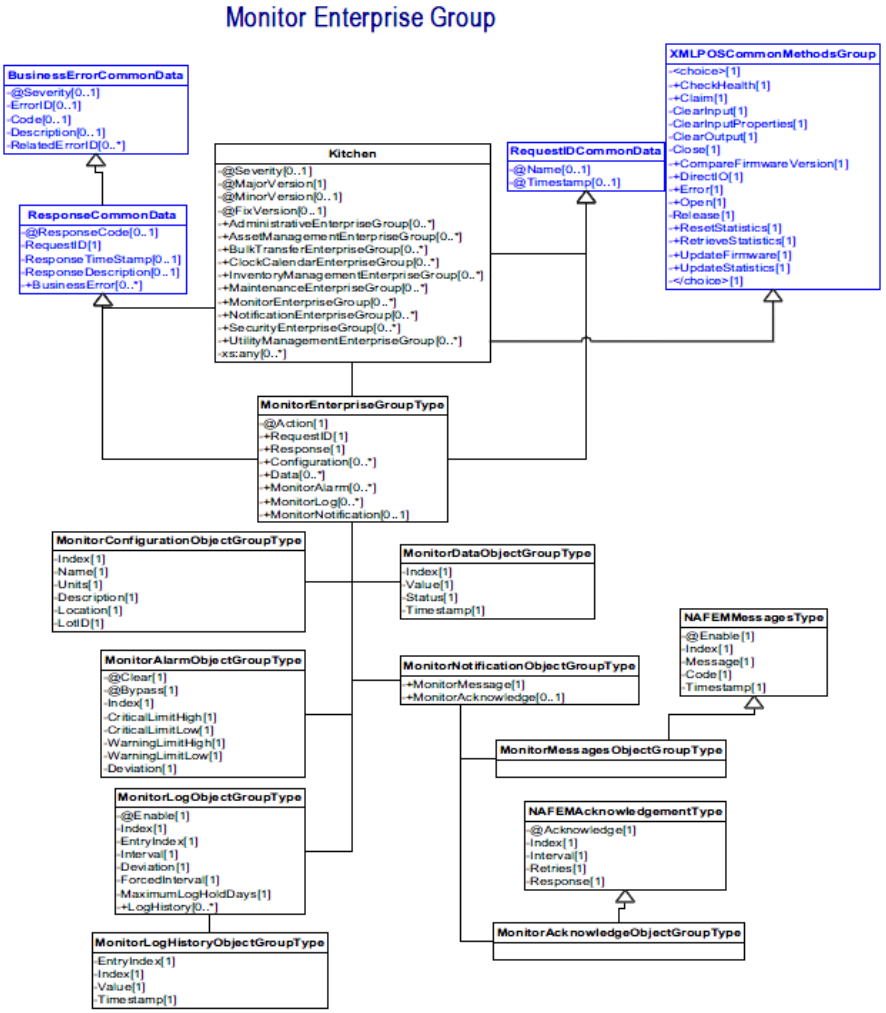


Figure 161: Monitor Enterprise Group Domain View

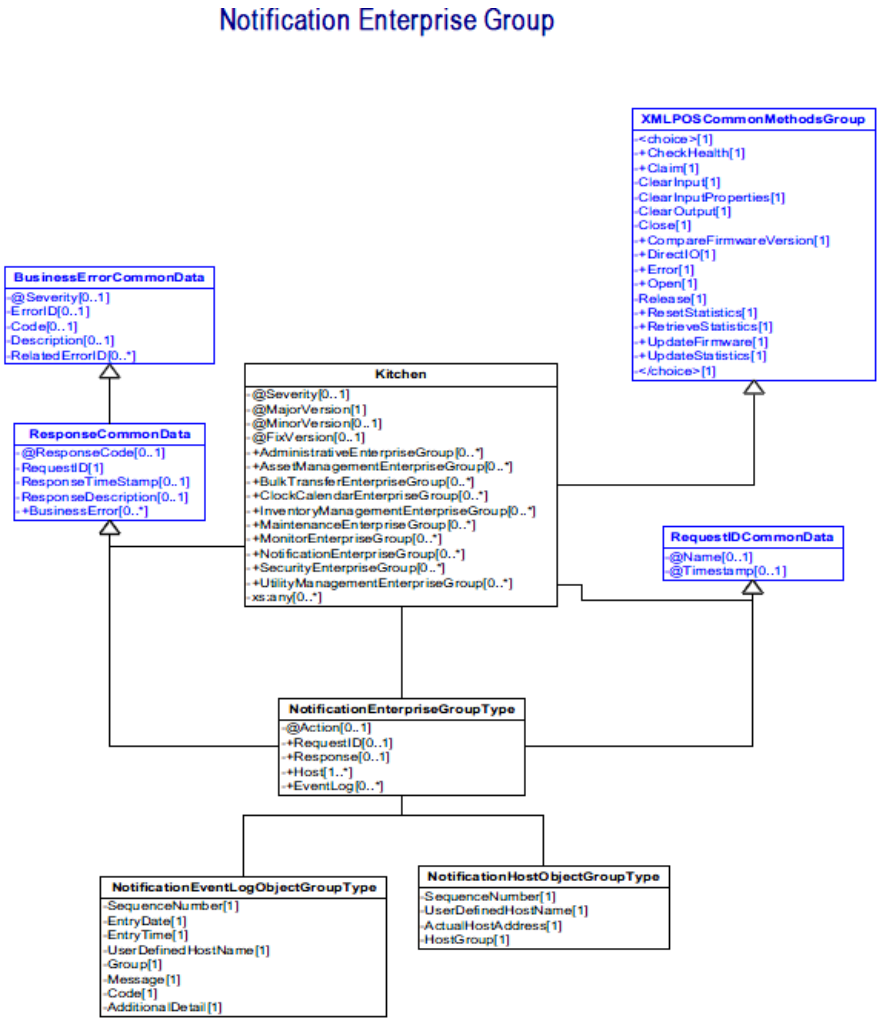


Figure 162: Notification Enterprise Group Domain View

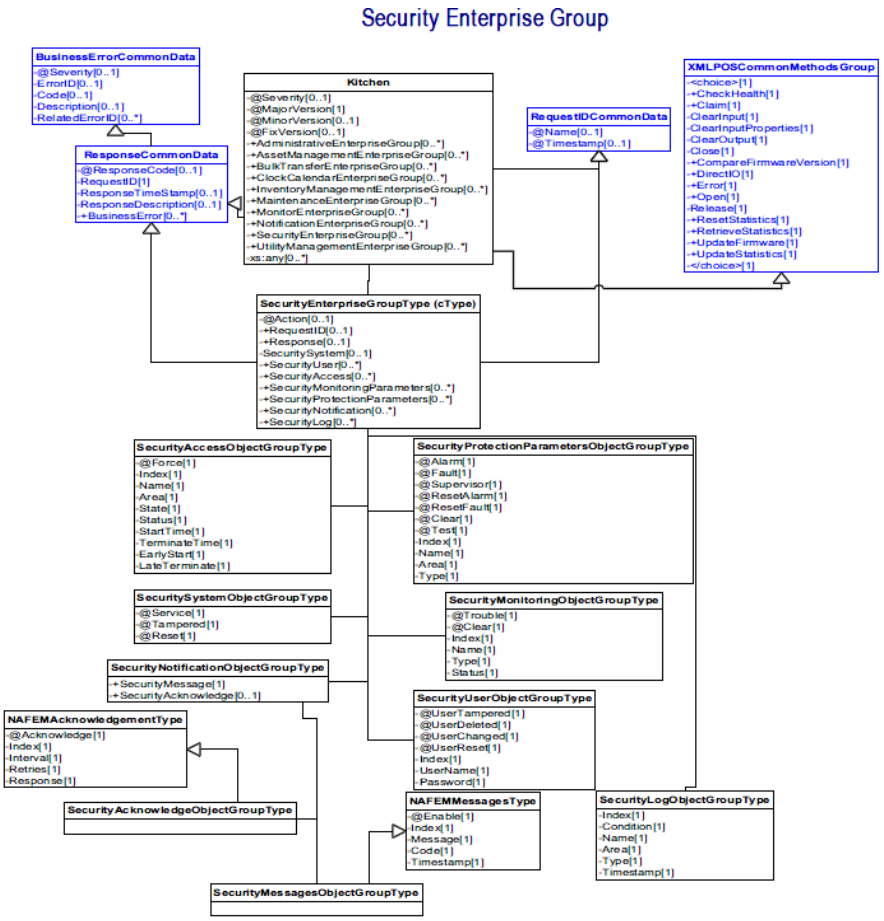


Figure 163: Security Enterprise Group Domain View

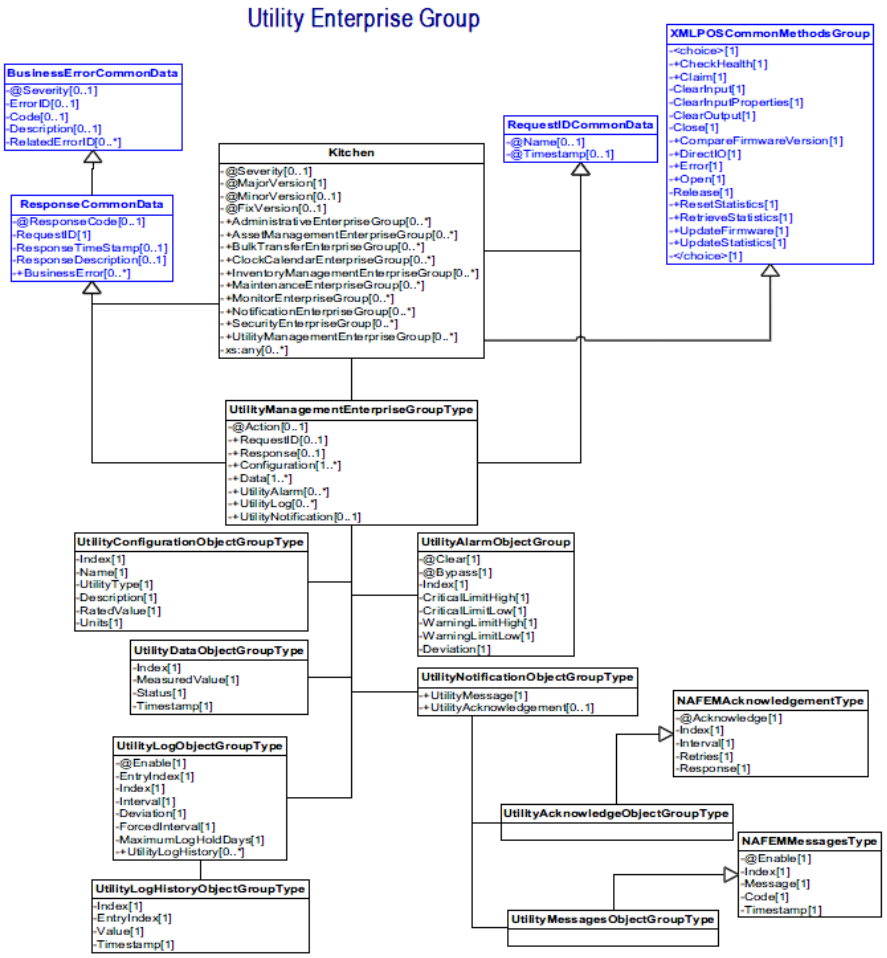


Figure 164: Utility Enterprise Group Domain View

Distributed Files

Distributed FilesThe following is a list of the XSD Schema files that are provided to support the XMLPOS environment.

BeltV1.16.0.xsd	BeltEventV1.16.0.xsd
BillAcceptorV1.16.0.xsd	BillAcceptorEventV1.16.0.xsd
BillDispenserV1.16.0.xsd	BillDispenserEventV1.16.0.xsd
BiometricsV1.16.0.xsd	BiometricsEventV1.16.0.xsd
BumpBarV1.16.0.xsd	BumpBarEventV1.16.0.xsd
CashChangerV1.16.0.xsd	CashChangerEventV1.16.0.xsd
CashDrawerV1.16.0.xsd	CashDrawerEventV1.16.0.xsd
CATV1.16.0.xsd	CATEventV1.16.0.xsd
CheckScannerV1.16.0.xsd	CheckScannerEventV1.16.0.xsd
CoinAcceptorV1.16.0.xsd	CoinAcceptorEventV1.16.0.xsd
CoinDispenserV1.16.0.xsd	CoinDispenserEventV1.16.0.xsd
DeviceMonitorV1.16.0.xsd	DeviceMonitorEventV1.16.0.xsd
ElectronicJournalV1.16.0.xsd	ElectronicJournalEventV1.16.0.xsd
ElectronicValueRWV1.16.0.xsd	ElectronicValueRWEventV1.16.0.xsd
FiscalPrinterV1.16.0.xsd	FiscalPrinterEventV1.16.0.xsd
GateV1.16.0.xsd	GateEventV1.16.0.xsd
GestureControlV1.16.0.xsd	GestureControlEventV1.16.0.xsd
GraphicDisplayV1.16.0.xsd	GraphicDisplayEventV1.16.0.xsd
HardTotalsV1.16.0.xsd	HardTotalsEventV1.16.0.xsd
ImageScannerV1.16.0.xsd	ImageScannerEventV1.16.0.xsd
IndividualRecognitionV1.16.0.xsd	IndividualRecognitionEventV1.16.0.xsd
ItemDispenserV1.16.0.xsd	ItemDispenserEventV1.16.0.xsd
KeylockV1.16.0.xsd	KeylockEventV1.16.0.xsd
LightsV1.16.0.xsd	LightsEventV1.16.0.xsd
LineDisplayV1.16.0.xsd	LineDisplayEventV1.16.0.xsd
MICRV1.16.0.xsd	MICREventV1.16.0.xsd
MotionSensorV1.16.0.xsd	MotionSensorEventV1.16.0.xsd
MSRV1.16.0.xsd	MSREventV1.16.0.xsd
PINPadV1.16.0.xsd	PINPadEventV1.16.0.xsd
PointCardRWV1.16.0.xsd	PointCardRWEventV1.16.0.xsd
POSKeyboardV1.16.0.xsd	POSKeyboardEventV1.16.0.xsd
POSPowerV1.16.0.xsd	POSPowerEventV1.16.0.xsd
POSPrinterV1.16.0.xsd	POSPrinterEventV1.16.0.xsd
RemoteOrderDisplayV1.16.0.xsd	RemoteOrderDisplayEventV1.16.0.xsd

RFIDScannerV1.16.0.xsd	RFIDScannerEventV1.16.0.xsd
ScaleV1.16.0.xsd	ScaleEventV1.16.0.xsd
ScannerV1.16.0.xsd	ScannerEventV1.16.0.xsd
SignatureCaptureV1.16.0.xsd	SignatureCaptureEventV1.16.0.xsd
SmartCardRWV1.16.0.xsd	SmartCardRWEventV1.16.0.xsd
SoundPlayerV1.16.0.xsd	SoundPlayerEventV1.16.0.xsd
SoundRecorderV1.16.0.xsd	SoundRecorderEventV1.16.0.xsd
SpeechSynthesisV1.16.0.xsd	SpeechSynthesisEventV1.16.0.xsd
ToneIndicatorV1.16.0.xsd	ToneIndicatorEventV1.16.0.xsd
VideoCaptureV1.16.0.xsd	VideoCaptureEventV1.16.0.xsd
VoiceRecognitionV1.16.0.xsd	VoiceRecognitionEventV1.16.0.xsd

付 録 G
削除予定履歴

この付録は、第1.11版で追加されました。

仕様書から削除されるプロパティ、メソッド、コンスタントの履歴を記述します。サポート終了が近いリリース状態で終了が通知されているときに、提供されるリリースレベルを示します。

デバイス名	機能名称	リリース終了	サポート終了	その他
FiscalPrinter	CapAmountNotPaid	1.11		
FiscalPrinter	PrintRecVoidItem	1.11		
PosPrinter/ PrintBarCode	PTR_BCS_RSS14 PTR_BCS_RSS_EXPANDED	1.12		
Scanner/ ScanDataType	SCANPTR_BCS_RSS14 SCAN_BCS_RSS_EXPANDED	1.12		

付　録　H

システムマネジメントインフォメーション

この付録は、第1.13 版で追加されました。

詳細はUPOS仕様書Version1.14.1を参照ください。