EPSON
EXCEED YOUR VISION

OFSCドライバー
ユーザーズマニュアル

概要

OFSCドライバーの動作環境、システム構成について説明します。

仕様

印刷処理方法や、エラーハンドリングについて説明します。

サンプルプログラム

サンプルプログラムの使い方について説明します。

メンテナンス

デバイスの入れ替え方法や、設定ファイルについて説明します。

M00018402
Rev. C

ご注意

- **本書の内容の一部または全部を無断で転載、複写、複製、改ざんすることは固く お断りします。**
- **本書の内容については、予告なしに変更することがあります。最新の情報はお問い合わせく ださい。**
- **本書の内容については、万全を期して作成いたしましたが、万一ご不審な点や誤り、記載も れなど、お気づきの点がありましたらご連絡く ださい。**
- **運用した結果の影響については、上項にかかわらず責任を負いかねま すのでご了承く ださい。**
- **本製品がお客様により不適切に使用さ れたり、本書の内容に従わずに取り扱われたり、またはエプソ ンおよびエプソ ン指定の者以外の第三者により修理・変更さ れたこと などに起因して生じた損害などにつきましては、責任を負いかねま すのでご了承く ださい。**
- **エプソ ン純正品およびエプソ ン品質認定品以外のオプショ ンまたは消耗品を装着してト ラブルが発生した場合には、責任を負いかねま すのでご了承く ださい。**

商標について

EPSON® および ESC/POS® はセイコーエプソン株式会社の登録商標です。

Microsoft®、Windows®、Windows Vista®、Windows Server®、Visual Basic®、Visual C++® は、
米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。

その他、記載されている会社名、製品名は、各社の商標、または登録商標です。

安全のために

記号の意味

本書では以下の記号が使われています。それぞれの記号の意味をよく理解してから製品を取り扱ってください。

⚠ **警告**	この表示を無視して、誤った取り扱いをすると、人が死亡または重傷を負う可能性が想定される内容を示しています。
⚠ **注意**	この表示を無視して、誤った取り扱いをすると、次のような被害が想定される内容を示しています。 • 人が傷害を負う可能性 • 物的損害を起こ す可能性 • データ などの情報損失を起こ す可能性
注意	ご使用上、必ずお守りいただきたいことを記載しています。この表示を無視して誤った取り扱いをすると、製品の故障や動作不良の原因になる可能性があります。
参考	補足説明や知っておいていただきたいことを記載しています。

使用制限

本製品を航空機・列車・船舶・自動車などの運行に直接関わる装置・防災防犯装置・各種安全装置など機能・精度などにおいて高い信頼性・安全性が必要とされる用途に使用される場合は、これらのシステム全体の信頼性および安全維持のためにフェールセーフ設計や冗長設計の措置を講じるなど、システム全体の安全設計にご配慮いただいた上で当社製品をご使用いただくようお願いいたします。

本製品は、航空宇宙機器、幹線通信機器、原子力制御機器、医療機器など、きわめて高い信頼性・安全性が必要とされる用途への使用を意図しておりませんので、これらの用途には本製品の適合性をお客様において十分ご確認のうえ、ご判断ください。

本書について

本書の目的

本書は、OFSC ドライバーを使ったシステムの構築、設計またはプリンターアプリケーションの開発、設計に必要なすべての情報を開発技術者に提供することを、その目的としています。

本書の構成

本書は次のように構成されています。

目次

概要

本章では、OFSC ドライバーの動作環境、システム構成について説明しています。

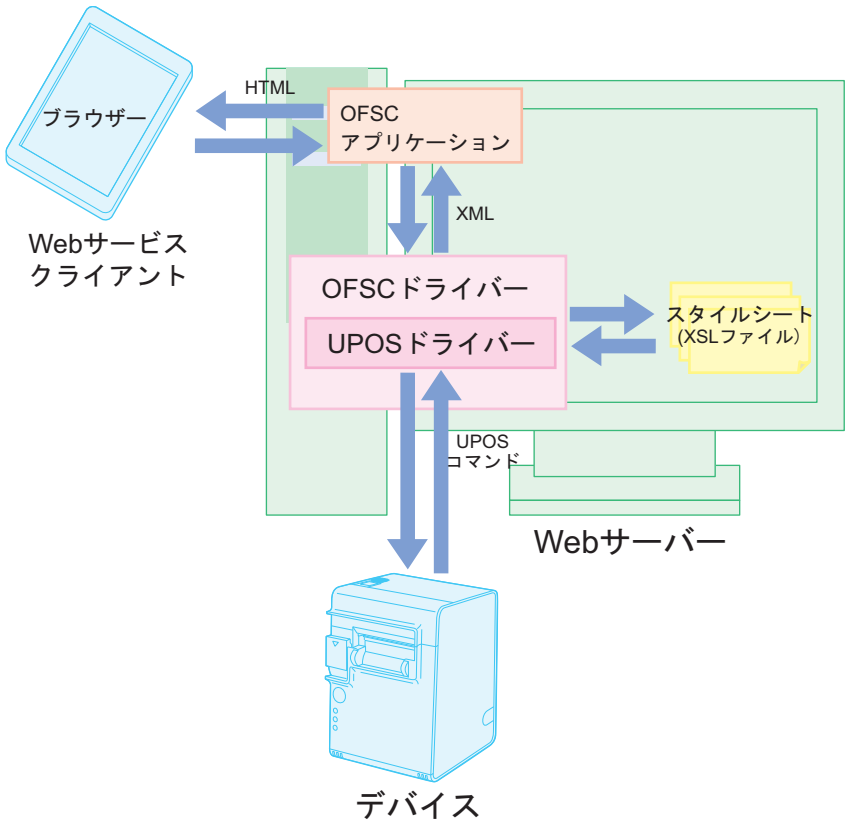
OFSC ドライバーの概要

OFSC ドライバーとは、飲食業界向けの POS / Order Entry System で使用されることを想定した XML ベースのドライバーです。Web サービスの環境で動作します。また、OFSC（Open Foodservice System Consortium）機器標準接続規格に準拠してデバイスを制御します。OFSC 機器標準接続規格については、下記 URL を参照してください。

http://www.ofsc.jp/

OFSC ドライバーを使ったアプリケーションは、Web Service としての OFSC ドライバーからのレスポンスを待たないで処理することもできます。

OFSC ドライバーを使ったアプリケーションは、以下の構造で動作します。



動作環境

ハードウェア（CPU / マシンスペック）

オペレーションシステム、Web Server の推奨スペックに従ってください。

言語

日本語

オペレーション システム

Java 環境

Windows：

- Microsoft Windows 7 SP1 (32 bit)
- Microsoft Windows Vista SP2 (32 bit)
- Microsoft Windows XP SP3 (32 bit)
- Microsoft Windows 2008 Server (32 bit)

Linux：

- Redhat 9 （パラレル非サポート）
- Redhat Enterprise Linux 5 Desktop (32 bit) （パラレル /USB 非サポート）
- Suse Linux Enterprise 10 SP1 (32 bit) （パラレル非サポート）

.NET 環境

Windows：

- Microsoft Windows 7 SP1 (32 bit)
- Microsoft Windows Vista SP2 (32 bit)
- Microsoft Windows XP SP3 (32 bit)
- Microsoft Windows 2008 Server (32 bit)
- Microsoft Windows Embedded POSReady 7(32bit)
- Microsoft Windows Embedded POS Ready 2009

動作条件

> **参考**　外部モジュールの詳細な情報は、インストールマニュアルを参照してください。

Java 環境

- Java SE Runtime Environment (JRE)
- Java Advanced Imaging
- Tomcat
- Axis2
- Xerces2

.NET 環境

- .NET Framework 2.0 SP1 以上 (Windows XP)
- .NET Framework 3.0 以上 (Windows Vista)
- .NET Framework 3.5 SP1 以上 (Windows 7)
- Microsoft Internet Information Services
- POS for .NET v1.11

提供物

弊社では、OFSC 環境用のドライバーパッケージ、デバイス、マニュアル、サンプルプログラムを提供しています。

OFSC ドライバーパッケージ

OFSC ドライバーパッケージを環境ごとに用意しています。

■ Windows / Java 環境
パッケージ名 : EPSON_OFSC_Driver_Java_Vx.xx.exe

■ Windows / .NET 環境
パッケージ名 : EPSON_OFSC_Driver_NET_Vx.xx.exe

■ Linux / Java 環境
パッケージ名 : EPSON_OFSC_Driver_Java_Vx.xx.bin

デバイス

デバイス	インターフェ イス	用途
TM-T90KP	Ethernet	レシート印刷 伝票印刷 (キッチン、フロア)
TM-T88V	シリアル パラレル [*1] USB[*2] Ethernet	レシート印刷
TM-T70		
キャッシュドロアー	DK コネクター [*3]	会計 (レシート印刷時)

[*1] .NET 環境でサポートしています。Java 環境ではサポートしていません。

[*2] Linux 環境の Redhat Enterprise Linux 5 Desktop は、サポートしていません。

[*3] TM-T88V/TM-T70 に接続して使います。

> **参考** 使用 するキャッシュドロ アーの詳細は、販売店まで お問い合わせく ださい。

マニュアル

環境ごとにマニュアルを提供しています。

■ Windows / Java 環境
パッケージ名 : EPSON_OFSC_Driver_Java_Manual_Vx.xx.zip

■ Windows / .NET 環境
パッケージ名 : EPSON_OFSC_Driver_NET_Manual_Vx.xx.zip

■ Linux / Java 環境
パッケージ名 : EPSON_OFSC_Driver_Java_Sample_Vx.xx.tar.gz

インストールマニュアル

OFSC ドライバーを使用する前に必要な環境構築の手順から、インストール / アンインストール方法、
OFSC ドライバーを自動的にインストールするサイレントインストール機能の使い方までを説明しています。

ユーザーズマニュアル

OFSC ドライバーを使用する方法や、サンプルプログラムの使用方法を説明しています。

サンプルスタイルシート リファレンスマニュアル

サンプルスタイルシートを使って、印刷レイアウトを編集する方法を説明しています。

サンプルプログラム

OFSC ドライバーを使ったサンプルプログラムを、環境ごと提供しています。

■ Windows / Java 環境
パッケージ名 : EPSON_OFSC_Driver_Java_Sample_Vx.xx.zip

■ Windows / .NET 環境
パッケージ名 : EPSON_OFSC_Driver_NET_Sample_Vx.xx.zip

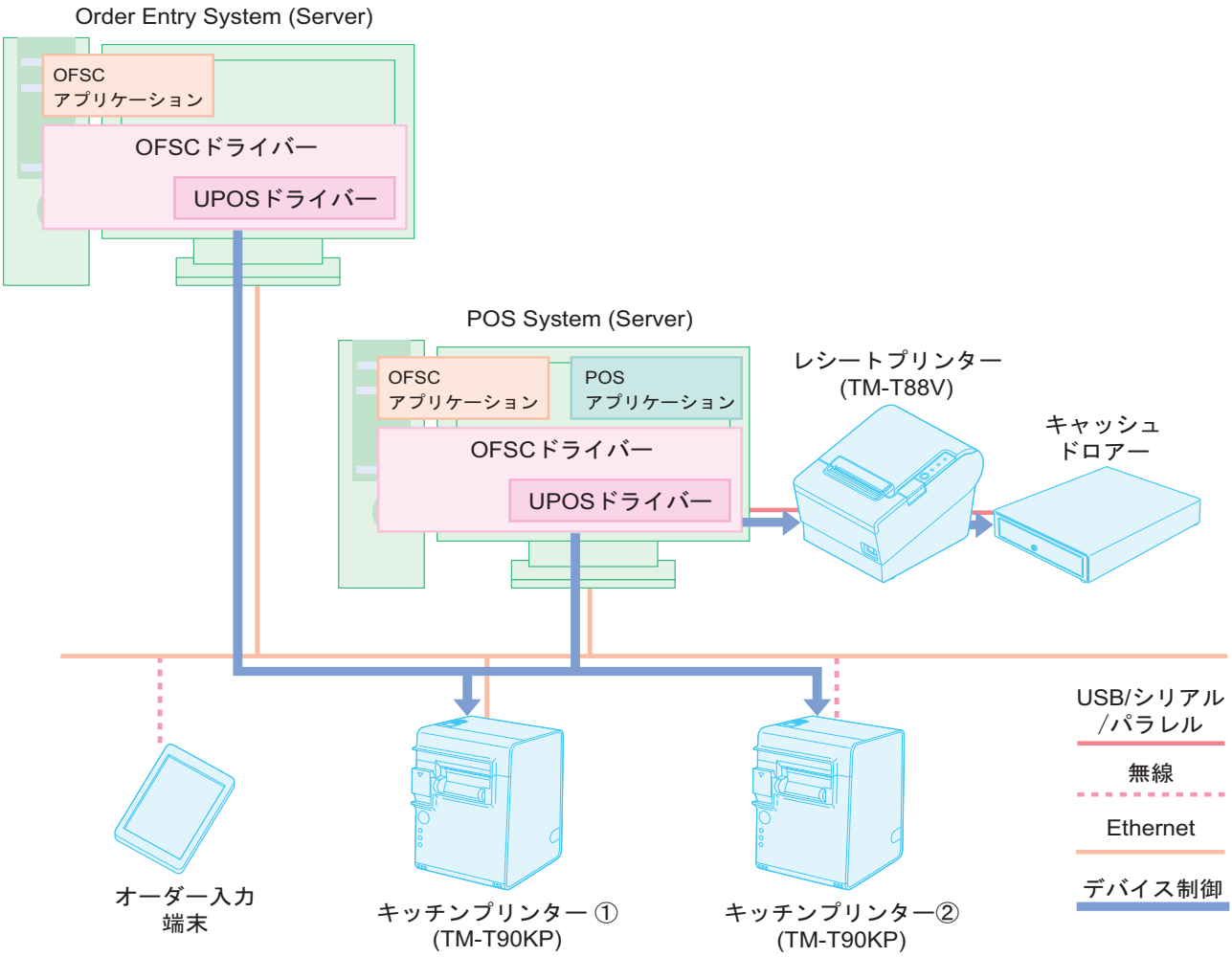
■ Linux / Java 環境
パッケージ名 : EPSON_OFSC_Driver_Java_Sample_Vx.xx.tar.gz

サンプルプログラムには、以下が含まれています。詳細は、25 ページ「サンプルプログラム」を参照してください。

- サンプルアプリケーション
- サンプルスタイルシート

システム構築例

OFSC ドライバーを使ったシステムでは、飲食業界向けの Order Entry System を構築できます。
以下は Order Entry System と POS System を併用したシステム構築例です。



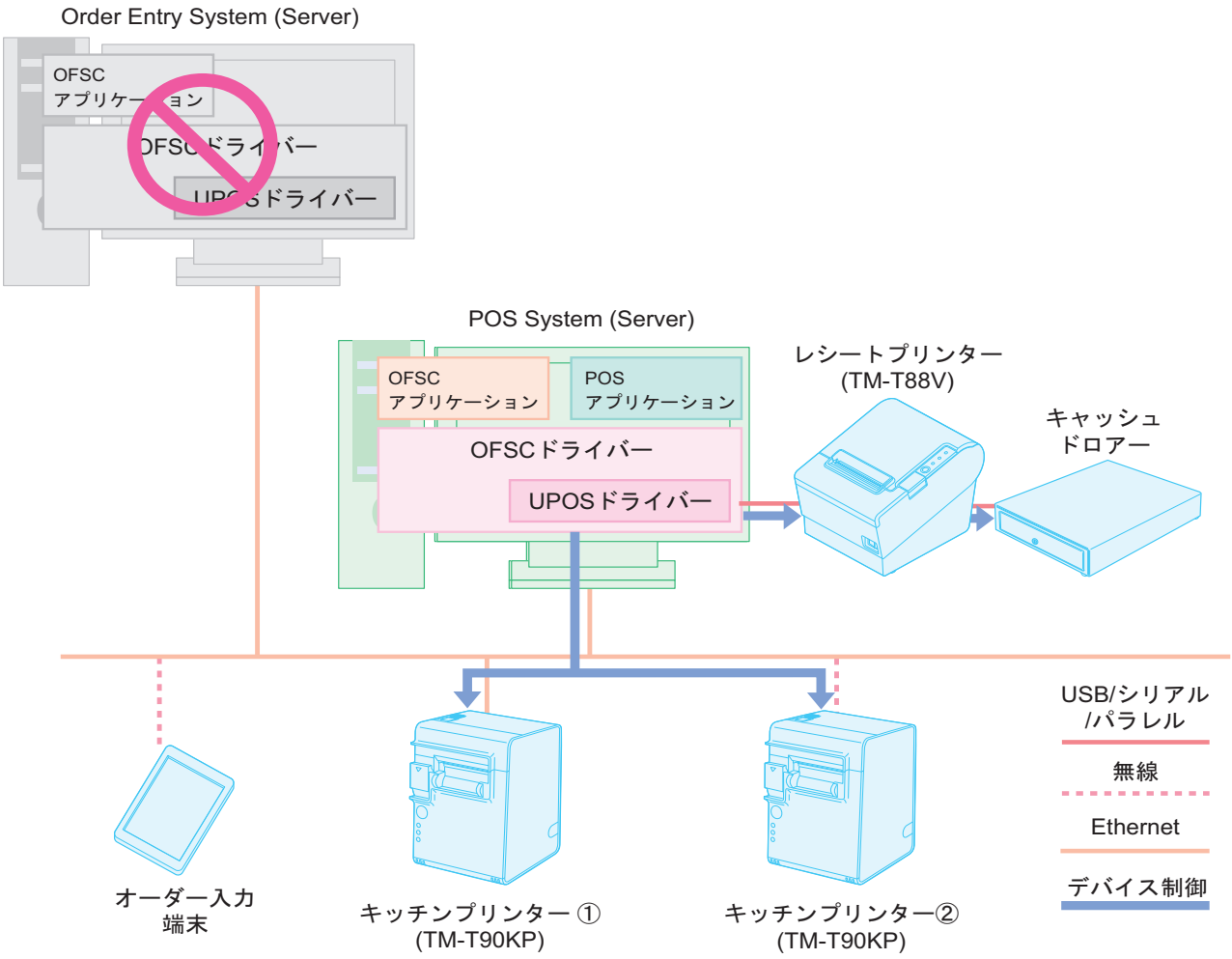
- Order Entry System
  アプリケーション：OFSC アプリケーション
  デバイス制御：　OFSC ドライバーを使って伝票をキッチンプリンターに印刷
  （キッチンプリンターは、複数台配置されることを想定し、Ethernet（有線/無線 LAN）接続）
- POS System
  アプリケーション：・POS アプリケーション
  ・OFSC アプリケーション
  デバイス制御：　UPOS ドライバーを使ってレシートをレシートプリンターに印刷し、キャッシュドロアーをオープン

以下のようなトラブルが発生した場合でも、代用手段を使ってシステムを運用できます。

- Order Entry System がダウンした場合（13 ページ）
- POS System がダウンした場合（14 ページ）
- キッチンプリンターがすべて故障した場合（15 ページ）
- レシートプリンターが故障した場合（16 ページ）

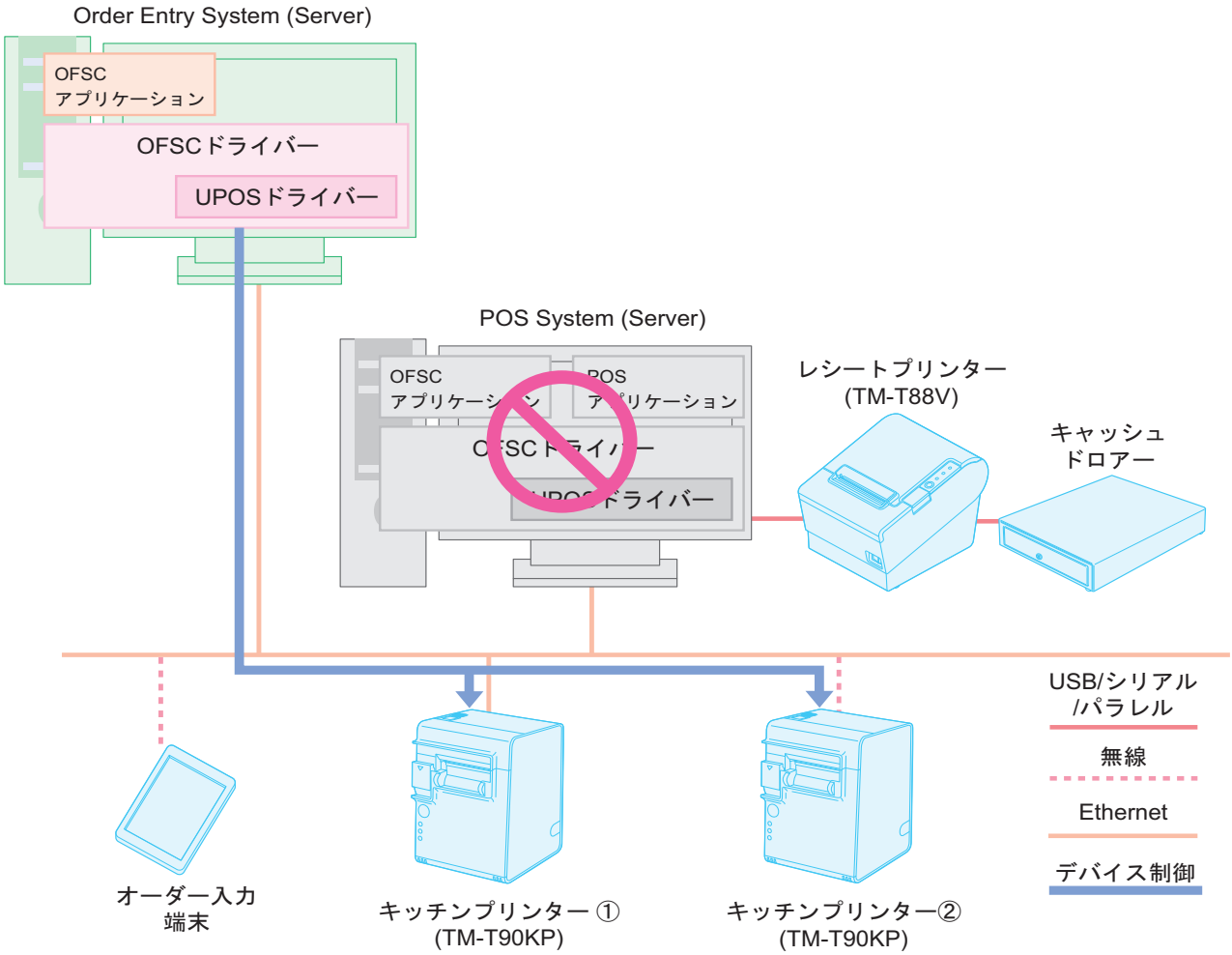
Case1 : Order Entry System がダウンした場合

Order Entry System の代わりに、POS System の OFSC ドライバーを使ってキッチンプリンターに伝票を印刷します。



Case2 : POS System がダウンした場合

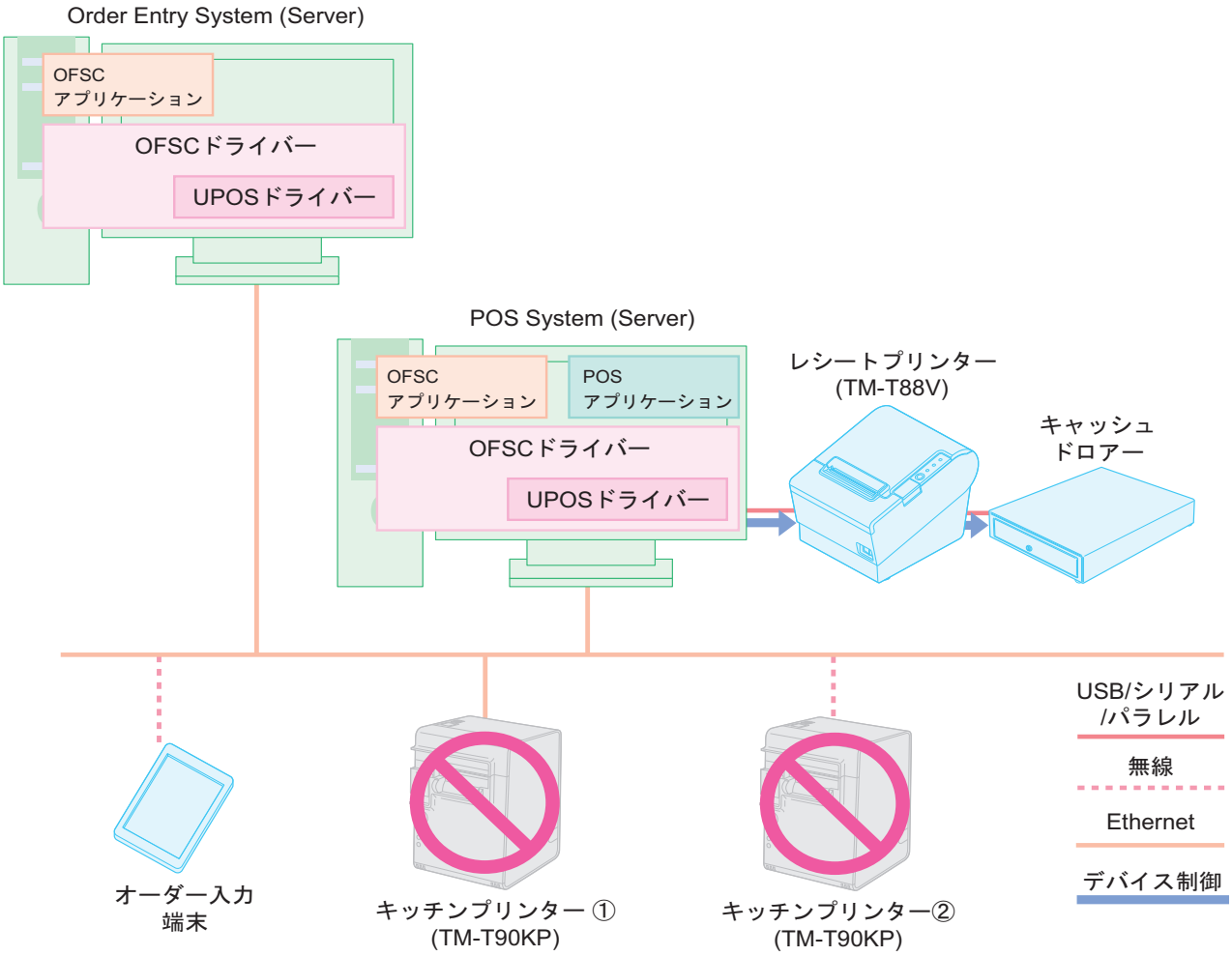
POS System の代わりに、Order Entry System の OFSC ドライバーを使って、レシートプリンターにレシートを印刷します。



Case3 : キッチンプリンターがすべて故障した場合

Order Entry System の OFSC アプリケーションは、POS System の OFSC ドライバー使ってレシートプリンターに伝票を印刷します。

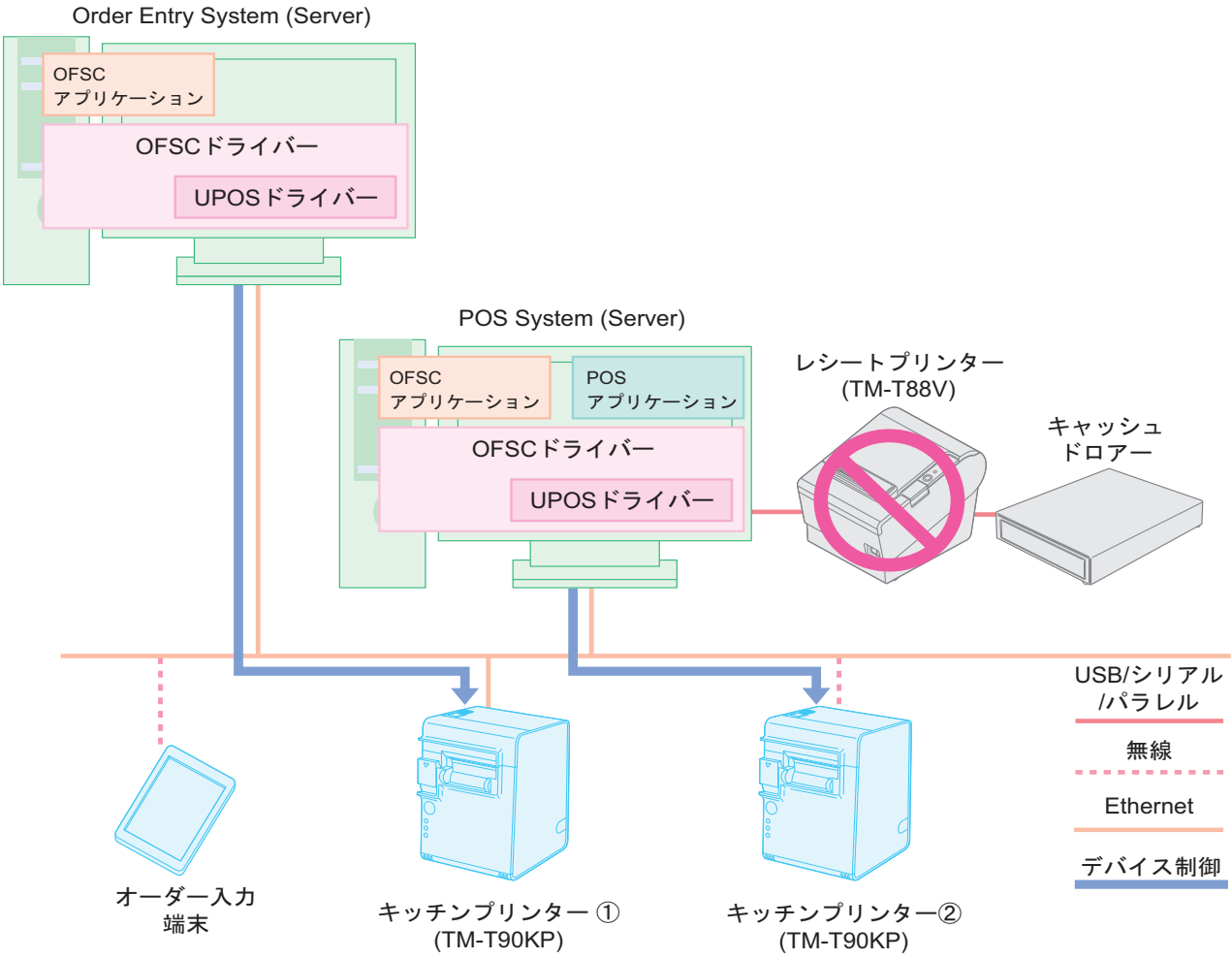
> **参考**　OFSC ドライバーを利用するアプリケーションは、OFSC ドライバーの URL を変更することによって、制御するデバイスの切り替えができます。



Case4 : レシート プリンターが故障した場合

POS System の POS アプリケーションは、UPOS ドライバーを使ってキッチンプリンターにレシートを印刷します。

以下の図は、キッチンプリンター①を Order Entry System の OFSC アプリケーションで制御し、キッチンプリンター②を POS System の POS アプリケーションで制御します。



制限事項

- IIS のサービスを開始して 1 回目の印刷では、印刷が開始されるまでに 15 秒程度掛かります。
  これは、WebService の初期化が行われているためです。2 回目以降の印刷では、即時印刷実行されます。
- 印字中にデバイスの電源を再投入したり、カバーを開けた場合、正常に印字されないことがあります。
- デバイスの電源をOFF/ONする場合、電源をOFFしてから必ず5秒以上の間隔をあけて電源をONしてください。
- 長時間かかる、迂回・逐次処理が定義されたメッセージを送信した場合、ASP.NET のタイムアウトにより、処理が中断されてしまう場合があります。ASP.NET の初期値では、タイムアウトは 110 秒に設定されています。メッセージは、これより短い時間で完了するようにしてください。
- Windows 7/ Vista/ XP/ Embedded POSReady7/ Embedded POSReady 2009 の場合、OFSC ドライバーに同時に 10 以上のメッセージを送信しないでください。IIS の機能制限により、10 以上のコネクションが確立できないようになっています。
  （IIS のコネクションについては、インストールマニュアル（.NET）1 章「IIS のコネクションについて」を参照してください。）
- デバイスを接続する場合、スイッチングハブを使用してください。
  スイッチングハブを使用しない場合、トラフィックの状況によってデバイスが ONLINE 状態でも OFFLINE と判断される可能性が高くなります。
- プリンター動作中にコンピュータが、スタンバイ / 休止状態に移行すると、ドライバの動作が不安定になる場合があります。システムスタンバイおよび休止状態を無効に設定してください。

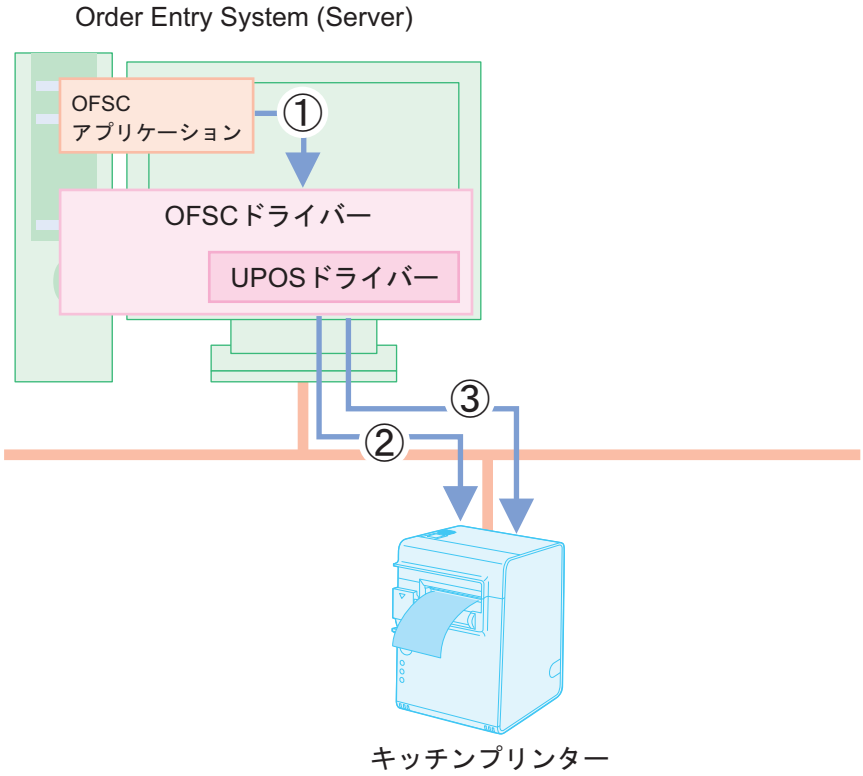
仕様

本章では、OFSC ドライバーの印刷処理方法や、エラーハンドリングについて説明しています。

印刷処理

逐次処理

逐次処理とは、制御対象のデバイスが正常に完了した場合、続けてデバイスを制御する処理です。



動作例：

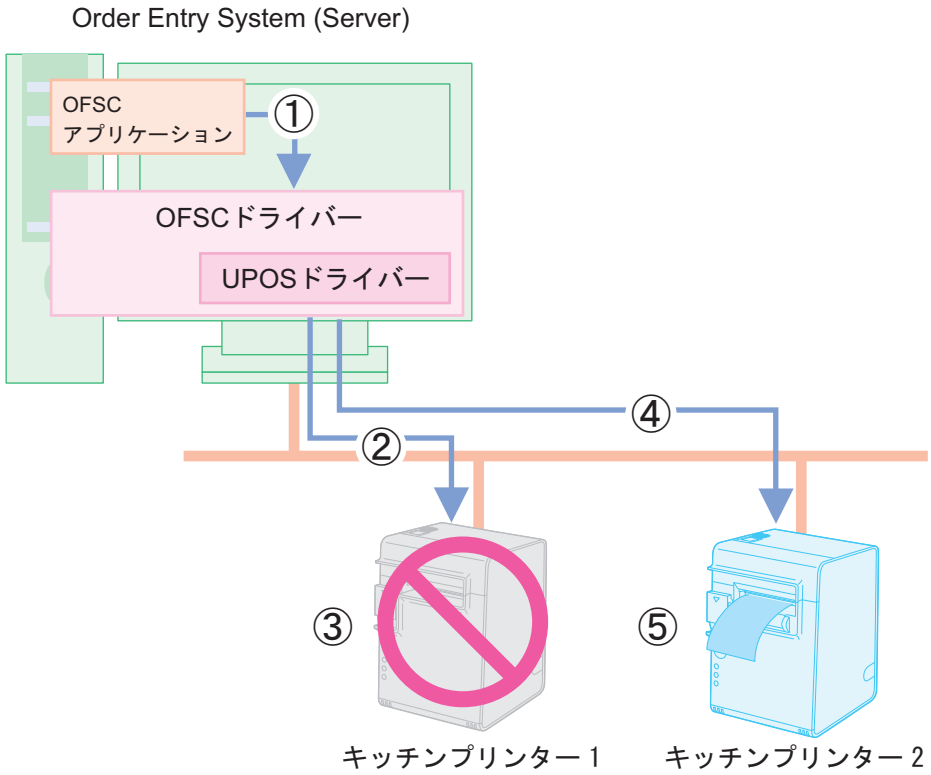
① OFSC アプリケーションから、調理指示一枚伝票とゲスト伝票の印刷を実行します。

② OFSC ドライバーは、キッチンプリンターに調理指示一枚伝票を印刷する命令を出します。

③調理指示一枚伝票の印刷が成功した場合のみ、OFSC ドライバーはゲスト伝票を印刷する命令を出します。

迂回処理

迂回処理とは、制御対象のデバイスが正常に動作しない場合、別のデバイスに切り替えて命令を実行する処理です。



動作例：

① OFSC アプリケーションから、伝票印刷を実行します。

② OFSC ドライバーは、キッチンプリンター１に伝票を印刷する命令を出します。

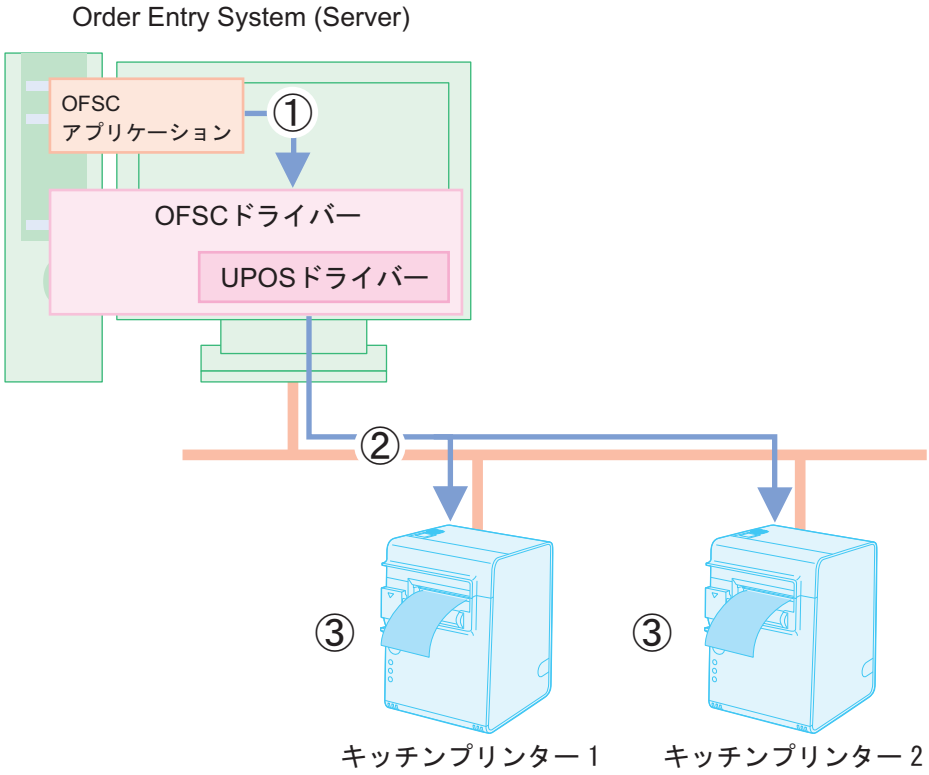
③キッチンプリンター１は故障していて印刷できません。

④ OFSC ドライバーは、キッチンプリンター２に伝票を印刷する命令を出します。

⑤キッチンプリンター２で伝票が印刷されます。

並列処理

並列処理とは、複数のデバイスを同時に制御する処理です。



動作例：

① OFSC アプリケーションから、２枚の伝票印刷を実行します。

② OFSC ドライバーは、キッチンプリンター１とキッチンプリンター２に伝票を印刷する命令をします。

③キッチンプリンター１とキッチンプリンター２から同じ伝票が印刷されます。

レスポンス

お客様のアプリケーションは OFSC ドライバーに情報を送り、返ってきたレスポンスにより、OFSC ドライバーやデバイスの状態を知ることができます。

> **参考** 要求メッセージの必須項目の詳細は、OFSC 接続機器標準規格を参照してください。
> （7 ページの「OFSC ドライバーの概要」を参照）

転送失敗エラー

アプリケーションと OFSC ドライバーが通信できなかった場合に発生します。発生するエラーは以下のとおりです。

OFSC Fault

faultcode	faultstring	対処方法
SchemaError	要求メッセージがスキーマに適合していません。	要求メッセージがスキーマに適合しているか確認してください。
Unknown	予期せぬエラーです。	OFSC ドライバーに不具合が発生している可能性があります。OFSC ドライバーを再度インストールしてください。
その他	要求メッセージがメッセージが OFSC ドライバーに届いていません。	以下を確認してください。 • IIS/Tomcat が正常に稼動しているか確認 • 通信可能な状態か確認 • URL が正しいか確認 • 要求メッセージの内容がスキーマに適合しているか確認

レスポンスエラー

アプリケーションと OFSC ドライバーが通信できたものの、デバイスやスタイルシートに問題がある場合に発生します。発生するエラーは以下のとおりです。

OFSC 定義のエラーコード

エラーコード	OFSC の仕様	例
OrderInvalid	オーダー情報が不正	要求メッセージの必須項目が存在しません。 ARTSHeader が存在しません。 POSLog が存在しません。
FileNotFound	スタイルシートが存在しない	Stylesheet の名前が指定されていません。 Stylesheet にアクセスできません。
DeviceNotFound	デバイスが存在しない	出力先のデバイスの稼動状況制御に問題があり、アクセスできません。
DeviceTimeout	デバイスがタイムアウト	出力先のデバイスの稼動状況に問題が発生し、制御に失敗した場合で、迂回処理が定義されていません。

エラーコード	OFSC の仕様	例
DeviceAlternative	デバイスが迂回処理実行	出力先のデバイスの稼動状況に問題が発生し、制御途中で失敗した場合で、迂回処理で成功しました。
DeviceNoAction	デバイスが動作しない	要求メッセージに、デバイス制御が存在しません。

OFSC 定義外の拡張エラー

ResponseCode	エラーの内容	例
TransformFailed	Stylesheet の適用エラー	Stylesheet、POSLog、 Stylesheet 適用後のデータ形式が不正です。
FaultConfig	設定が不正	デバイスの登録情報が不正です。 対象の出力クラスの定義が不正です。

サンプルプログラム

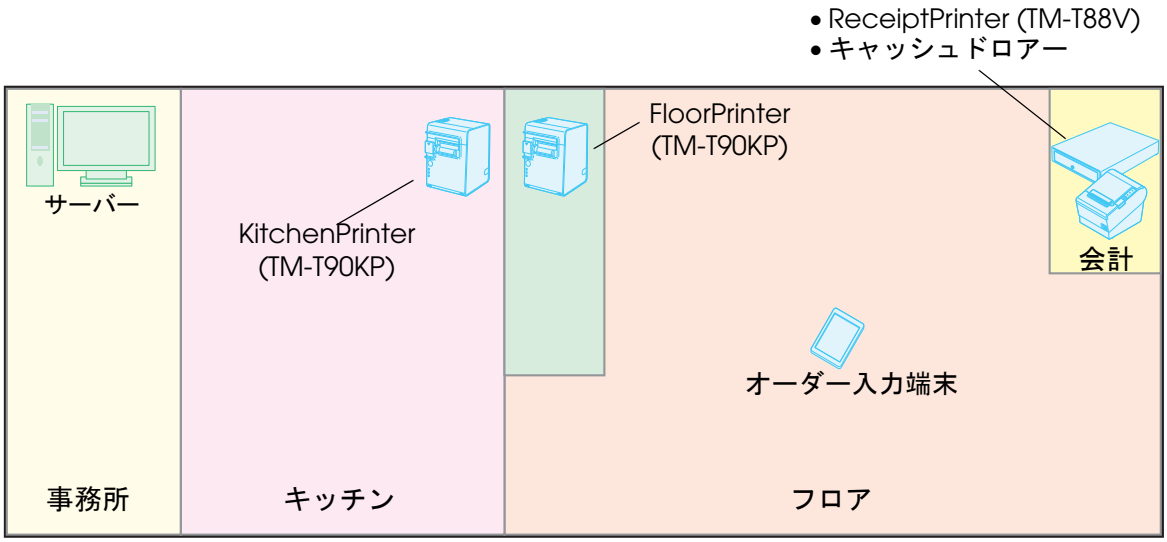
本章では、サンプルプログラムの使い方について説明しています。

サンプルプログラムのシステム概要

特徴

■ 伝票（用意されているスタイルシート）ごと印刷
■ 各印刷処理（迂回処理、並列処理、逐次処理）の実行
■ すべての印刷処理の併用
■ レシート印刷後にドロアーをオープンする POS システムの使用
■ 接続されているデバイスの状態のチェック

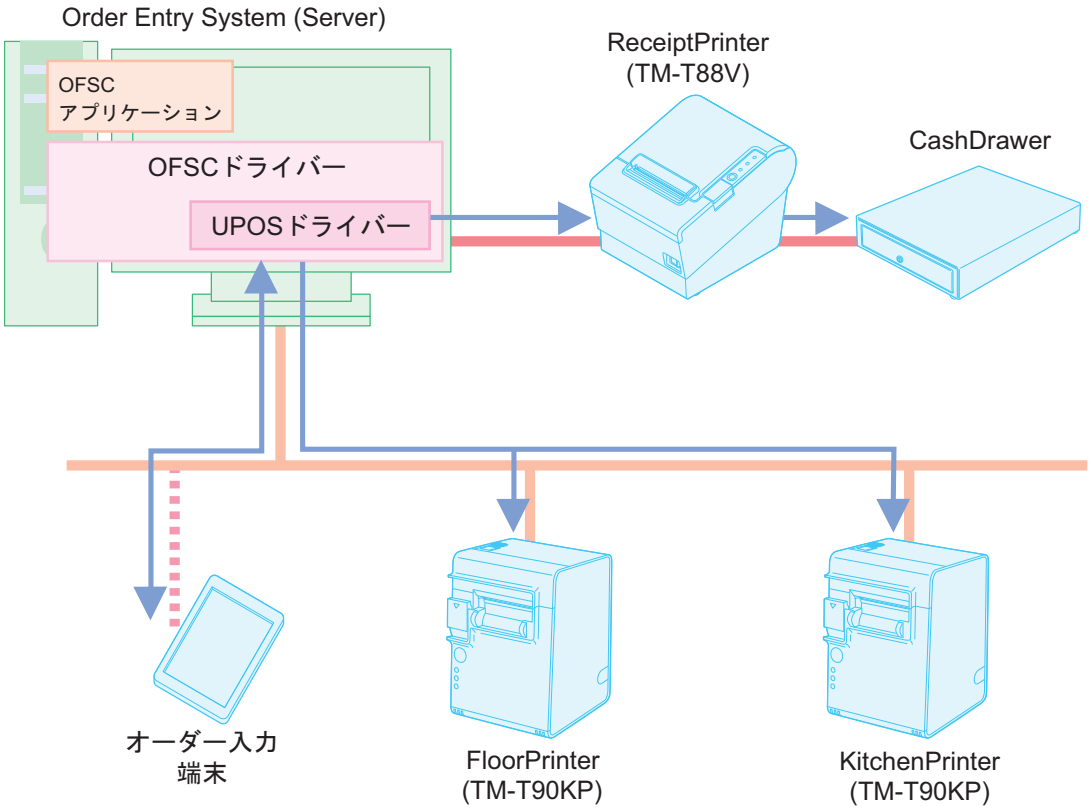
想定店舗



- KitchenPrinter：　キッチンスタッフへ調理指示単品伝票を印刷
- FloorPrinter：　フロアスタッフへ調理指示一枚伝票とお客様に渡すゲスト伝票を印刷
- ReceiptPrinter：　レシートを印刷し、キャッシュドロアーをオープン
- オーダー入力端末：注文を入力するブラウザー（クライアントコンピューターやブラウザー搭載の端末）

> **参考**　本章では、ReceiptPrinter を TM-T88V で説明しています。その他のプリンターをご使用の場合は、読み換えてお使いください。

想定システム



動作環境

サンプルプログラムは以下の環境で動作確認しています。

■ OFSC ドライバーが動作する環境
詳細は、インストールマニュアルを参照してください。

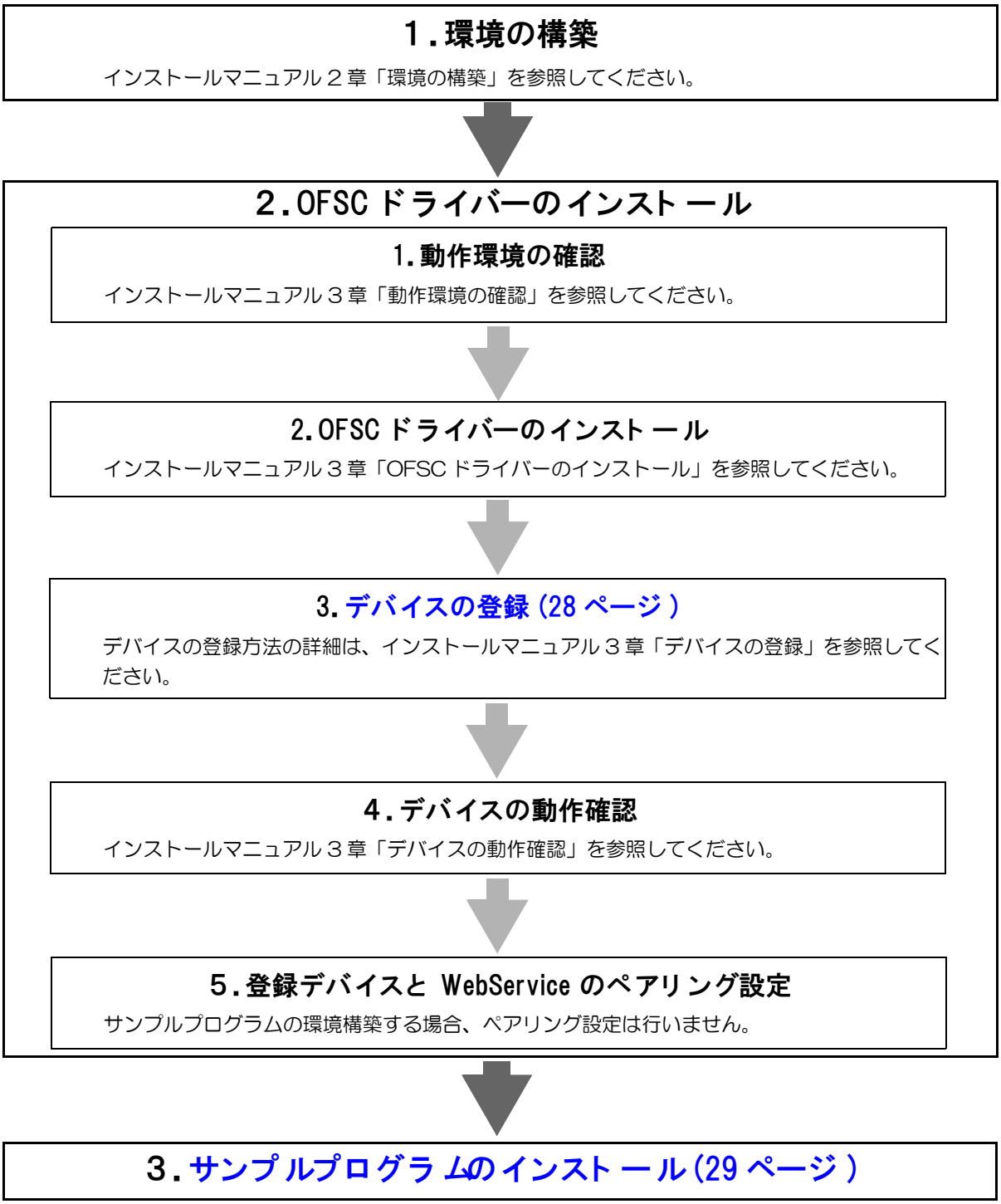
■ ブラウザーの環境
ブラウザーは、サンプルプログラムにアクセスするために必要です。

- Java 環境
  - ＊ Internet Explorer 6 SP1
  - ＊ Internet Explorer 7
  - ＊ Internet Explorer 8
  - ＊ Firefox（Redhat 9 は Ver.2.0.0.6、Redhat 9 以外は OS に付属しているバージョン）
- .NET 環境
  - ＊ Internet Explorer 6 SP1
  - ＊ Internet Explorer 7
  - ＊ Internet Explorer 8

> **参考**
> .NET 環境のサンプルプログラムは、VB Script で開発されています。
> そのため、Internet Explorer のブラウザーのみサポートしています。

サンプルプログラムの環境構築

サンプルプログラムの環境構築のフローを以下に示します。



デバイスの登録

サンプルプログラムでは以下のデバイスを使用します。

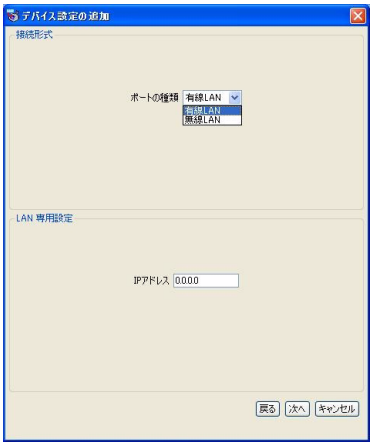
- TM-T90KP (2 台)
- TM-T88V/TM-T70 (1 台)
- キャッシュドロアー (1 台)

1 TMNetWinConfig を使って、プリンターに IP アドレスの設定をします。

IP アドレスをデバイスに設定する方法は、各デバイスの詳細取扱説明書を参照してください。
TMNet WinConfig は弊社のホームページより入手してください。
http://www.epson.jp/

2 SetupPOS を使ってデバイスを登録します。

デバイスの登録方法の詳細は、インストールマニュアル３章「デバイスの登録」を参照してください。



デバイスは以下の表の設定値で登録してください。

デバイス	設定			接続形式	共通設定
	論理デバイス名	デバイス類	デバイス名	ポート の種類	ー
TM-T90KP (1 台目)	KitchenPrinter	POSPrinter	TM-T90KP	有線 LAN	初期値
TM-T90KP (2 台目)	FloorPrinter	POSPrinter	TM-T90KP	有線 LAN	初期値
TM-T88V(TM-T70)	ReceiptPrinter	POSPrinter	TM-T88V (TM-T70)	有線 LAN	初期値
キャッシュドロアー	CashDrawer	CashDrawer	OfscStandard	DK コネクター	初期値

サンプルプログラムのインストール

以下の手順でサンプルプログラムをインストールします。

1 サンプルプログラムのパッケージを以下に解凍します。

解凍後のフォルダー構成は、30 ページの「サンプルプログラムのフォルダー構成」を参照してください。

- Java 環境の場合：
  CATALINA_HOME¥webapps¥ROOT
- .NET 環境の場合：
  システムドライブ:¥Inetpub¥wwwroot

> **参考**
> CATALINA_HOME とは、Tomcat のインストール先フォルダーです。

2 ブラウザーを起動させて、以下の URL を入力します。

- Java 環境の場合：
  http://localhost:8080/epsonsample/epsonsample.html
- .NET 環境の場合：
  http://localhost/epsonsample/epsonsample.html

> **参考**
> クライアントコンピューターやオーダー入力端末から動作する場合、
> URL をサーバーコンピューターの IP アドレスに書き換えてください。
> - Java 環境の場合：
>   http://XXX.XXX.XXX.XXX:8080/epsonsample/epsonsample.html
> - .NET 環境の場合：
>   http://XXX.XXX.XXX.XXX/epsonsample/epsonsample.html

サンプルプログラムのフォルダー構成

サンプルプログラムのフォルダー構成は以下のとおりです。青字のフォルダー、ファイルがサンプルプログラムです。

```
ROOT *1
├ epsonsample ..... サンプルプログラム
│  ├ epsonsample.html
│  ├ driverControl.js *2
│  ├ formControl.js *3
│  ├ epsonsample.css
│  ├ logo1.bmp
│  ├ logo2.bmp
│  ├ image
│  └ stylesheet
│     ├ Check_Drawer.xsl
│     ├ Check_Kitchen.xsl
│     ├ Check_Floor.xsl
│     ├ Check_Receiptxsl
│     ├ Cooking.xsl
│     ├ Guest.xsl
│     ├ Receipt.xsl
│     ├ Serving.xsl
│     ├ Re_Cooking.xsl
│     ├ Re_Serving.xsl
│     ├ DrawerResponse.xsl
│     └ PrinterResponse.xsl
└ SampleStyleSheets. ..... サンプルスタイルシート：
                           詳細は「OFSC ドライバー サンプルスタイルシート
                           リファレンスマニュアル」を参照してください。
```

*1 .NET 環境の場合、wwwroot です。

*2 .NET 環境の場合、driverControl.vbs です。

*3 .NET 環境の場合、formControl.vbs です。

ファイル概要

サンプルプログラムの各ファイルの概要は以下のとおりです。

ファイル		概要
epsonsample.html		サンプル HTML ファイル
• Java 環境 driverControl.js • .NET 環境 driverControl.vbs		• OFSC 機器標準機器規格に準拠した XML ファイルの作成 • OFSC ドライバーと接続するスクリプト • OFSC ドライバーからのレスポンスを解析するスクリプト
• Java 環境 formControl.js • NET 環境 formControl.vbs		• サンプル HTML の画面のコントロール • サンプル HTML の商品イラスト配置を作成するスクリプト
epsonsample.css		サンプル HTML の商品イラスト配置用のスタイルシート
logo1.bmp		レシート印刷で使用するロゴイメージ
logo2.bmp		ゲスト伝票で使用するロゴイメージ
image		サンプル HTML の商品イラスト格納用フォルダー
stylesheet		スタイルシート格納用フォルダー
	Check_Drawer.xsl	キャッシュドロアーのデバイスチェック
	Check_Kitchen.xsl	KitchenPrinter のデバイスチェック
	Check_Floor.xsl	FloorPrinter のデバイスチェック
	Check_Receipt.xsl	ReceiptPrinter のデバイスチェック
	Drawer.xsl	ドロアーオープン
	Cooking.xsl	調理指示単品伝票印刷
	Guest.xsl	ゲスト伝票印刷
	Receipt.xsl	レシート印刷
	Serving.xsl	調理指示一枚伝票印刷
	Re_Cooking.xsl	調理指示単品伝票の再印刷
	Re_Serving.xsl	調理指示一枚伝票の再印刷
	DrawerResponse.xsl	キャッシュドロアーのレスポンスを解析
	PrinterResponse.xsl	POS Printer のレスポンスを解析

画面構成

画面構成

サンプルプログラムは、以下の画面構成になっています。

項目		説明	初期状態
①メニュー		イメージから商品を選択	商品のイメージを表示
②オーダー入力端末番号		使用しているオーダー入力端末の番号を入力	HT10
③担当者		担当者名を入力	山田
④人数		お客様の人数 (1 ～ 10) を選択	1
⑤テーブル		テーブル番号を入力	A12
⑥オーダー表	ID	メニューで選択された商品の ID が表示	空欄
	名称	メニューで選択された商品の名称が表示	空欄
	単価	メニューで選択された商品の単価が表示	空欄
	数量	メニューで選択された商品の数量 (1 ～ 10) を指定	空欄
	金額	メニューで選択された商品の「単価×数量」が表示	空欄
	合計数量	オーダーした合計数量が表示	0
	合計金額	オーダーした合計金額が表示	0
⑦一行取消ボタン		オーダー表の最後尾の行を削除	使用可
⑧全行取消ボタン		オーダー表を初期状態に戻す	使用可
⑨注文ボタン		KitchenPrinter に調理指示単品伝票、FloorPrinter に調理指示一枚伝票、ゲスト伝票を印刷	使用可
⑩会計ボタン		ReceiptPrinter にレシート印刷し、ドロアーオープン	使用可
⑪デバイス診断ボタン		接続されているデバイスが、正常に動作するかチェック	使用可
⑫印刷スタイル	調理指示単品伝票	KitchenPrinter にオーダー表の内容を調理指示単品伝票で印刷	使用可
	調理指示一枚伝票	FloorPrinter にオーダー表の内容を調理指示一枚伝票で印刷	使用可
	ゲスト伝票	FloorPrinter にオーダー表の内容をゲスト伝票で印刷	使用可
	レシート	ReceiptPrinter にオーダー表の内容をレシートで印刷	使用可
⑬単機能処理	迂回	迂回処理で、オーダー表の内容を伝票に印刷	使用可
	並列	並列処理で、オーダー表の内容を伝票に印刷	使用可
	逐次	逐次処理で、オーダー表の内容を伝票に印刷	使用可
⑭処理結果		以下の実行結果を表示 • 注文ボタン • 会計ボタン • デバイス診断ボタン • 印刷スタイル • 単機能処理	空欄

伝票イメージ

サンプルプログラムは、以下の伝票を印刷します。

調理指示単品伝票

調理指示一枚伝票

ゲスト伝票

レシート

機能

入力

メニュー

メニューを選択すると、以下の画面が表示されます。[OK] ボタンを押すと、オーダー表にオーダー情報 (ID、名称、単価、数量、金額、合計数量、合計金額) が表示されます。また、合計数量と合計金額を再計算し、表示されます。

> **参考**　すでにオーダー表にあるメニューを選択した場合、数量が追加されます。またメニューを選択することによって、数量が 10 を超えた場合、新しい行に同じ商品が追加されます。

数量

選択したメニューの数量をオーダー表から変更できます。変更された数量で、金額が再計算されます。また、合計数量と合計金額も再計算し、表示されます。

一行取消

オーダー表の最後尾の注文を削除します。オーダー表に注文がない場合、以下の警告メッセージが表示されます。

全行取消

オーダー表を初期表示の状態にします。

注文

KitchenPrinter に調理指示単品伝票、FloorPrinter に調理指示一枚伝票、お客様に渡すゲスト伝票が印刷されます。
調理指示単品伝票の印刷に KitchenPrinter が失敗した場合、迂回処理で FloorPrinter に印刷されます。

会計

ReceiptPrinter にレシートが印刷されます。印刷が完了すると、ドロアーをオープンし、お会計をします。
印刷に失敗した場合、再印刷できます。

デバイス診断

接続されているデバイス (KitchenPrinter、FloorPrinter、ReceiptPrinter、CashDrawer) の状態を調べ、処理結果に表示します。

印刷 – 迂回処理

KitchenPrinter で調理指示単品伝票が印刷できなかった場合、FloorPrinter で調理指示単品伝票を印刷します。

想定される使用例

KitchenPrinter が故障したり、用紙が切れた場合にアプリケーションは、FloorPrinter に調理指示単品伝票を印刷します。フロアスタッフは、FloorPrinter に印刷された調理指示単品伝票をキッチンスタッフに渡します。

印刷 – 並列処理

同時に KitchenPrinter と FloorPrinter へ調理指示一枚伝票を印刷します。

想定される使用例

注文をキッチンスタッフが調理指示、フロアスタッフが配膳指示として把握するためにアプリケーションは、調理指示一枚伝票を KitchenPrinter と FloorPrinter へ同時に印刷します。

印刷 – 逐次処理

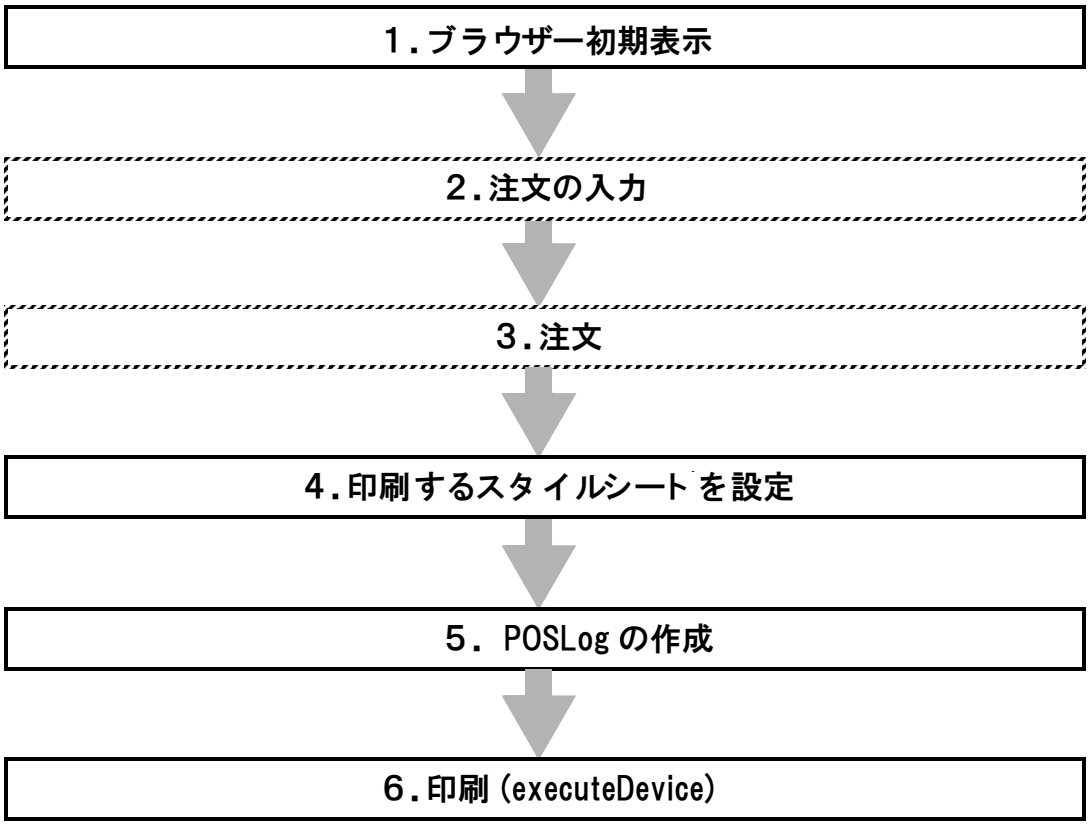
KitchenPrinter で調理指示単品伝票 1 枚と調理指示一枚伝票 1 枚を印刷します。

想定される使用例

FloorPrinter が故障した場合にアプリケーションは、KitchenPrinter で調理指示単品伝票の印刷に成功したのを確認し、調理指示一枚伝票をフロアスタッフ用に、KitchenPrinter で印刷します。
キッチンスタッフは、KitchenPrinter に印刷された調理指示一枚伝票をフロアスタッフに渡します。

プログラムの流れ

サンプルプログラムの初期表示から印刷完了まで以下の流れで動作します。



参考　executeDevice とは、OFSC ドライバーで印刷処理を実行するための API です。

使用している POSLog

サンプルプログラムでは、以下の POSLog を使用しています。

> **参考** POSLog の詳細は、以下を参照してください。
> http://www.nrf-arts.org/arts_xml_main.htm

POSLog/Transaction/RetailStoreID（店名）

固定値：〝長野県松本店〟

POSLog/Transaction/WorkstationID（端末番号）

画面の端末番号テキストフィールドから取得します。12 桁の制限があります。

POSLog/Transaction/SequenceNumber（伝票番号）

伝票を 0000 から印刷するたびにインクリメントします。アプリケーションがロードされるとリセットされます。

POSLog/Transaction/POSLogDateTime（年月日時間）

システム日付から取得します。

POSLog/Transaction/OperatorID（担当者 ID）

画面の担当テキストフィールドから取得します。5 桁の制限があります。

POSLog/Transaction/CustomerOrderTransaction

/ItemCount（合計数量）

画面のオーダー表の合計数量から取得します。

/LineItem/Sale/POSIdentity/POSItemID（商品 ID）

画面のオーダー表の ID から取得します。

/LineItem/Sale/Description（名称）

画面のオーダー表の名称から取得します。10 桁の制限があります。

/LineItem/Sale/RegularSalesUnitPrice（単価）

画面のオーダー表の単価から取得します。

/LineItem/Sale/ExtendedAmount（金額）

画面のオーダー表の金額から取得します。6 桁の制限があります。

/LineItem/Sale/Quantity（商品の数量）

画面のオーダー表の数量から取得します。2 桁の制限があります。

/Total/@TotalType="TransactionGrandAmount"（税込合計）

画面のオーダー表の合計金額から取得します。6 桁の制限があります。

/Total/@TotalType="TransactionNetAmount"（税抜き合計）

画面のオーダー表の「合計金額 - 内消費税」で計算し取得します。6 桁の制限があります。

/Total/@TotalType="TransactionTaxAmount"（内消費税）

画面のオーダー表の「合計金額× (5 / 105)」で計算し、取得します。

/Foodservice/TableID（テーブル ID）

画面のテーブルテキストフィールドから取得します。5 桁の制限があります。

/Foodservice/PartySize（人数）

画面の人数テキストフィールドから取得します。5 桁の制限があります。

レスポンスの解析

サンプルプログラムでは、OFSC ドライバーとデバイスのレスポンスを解析し、メッセージを画面の「処理結果」に表示します。(OFSC ドライバーで発生するエラーの詳細は、22 ページ「レスポンス」を参照してください。)

メッセージ

処理結果	メッセージ
成功	KitchenPrinter での印刷が成功しました。
失敗	KitchenPrinter での印刷が失敗しました。 (エラー内容も表示されます。詳細は、40 ページ「転送失敗エラー」、41 ページ「レスポンスエラー」を参照してください。)
迂回処理	KitchenPrinter から FloorPrinter へ迂回処理をしました。
並列処理	KitchenPrinter と FloorPrinter に並列処理をしました。
逐次処理	KitchenPrinter と KitchenPrinter で逐次処理をしました。

転送失敗エラー

転送失敗エラーとは、サンプルプログラムと OFSC ドライバーが通信できなかった場合に発生するエラーです。

Code	メッセージ	原因
SchemaError	要求メッセージがスキーマに適合していません。	要求メッセージが不正
その他	WebService のレスポンスメッセージ。	• 要求メッセージが不正 • OFSC ドライバーの設定が不正

レスポンスエラー

レスポンスエラーとは、OFSC ドライバーで発生するエラーと、デバイスで発生するエラーです。

OFSC ドライバーで発生するエラー

Code	メッセージ	原因
SchemaError	メッセージがスキーマに適合していません。	要求メッセージが不正
NotSupport	メッセージのデバイスカテゴリはサポートしていません。	Stylesheet が不正 (UPOS の指定方法が不正)
ConfigError	対象の制御デバイスが定義されていません。	• 要求メッセージが不正 • OFSC ドライバーの設定が不正
NotFound	対象のドライバーが存在しません。	ドライバーが正常にインストールされていない
OrderInvalid	オーダー情報が不正です。	• 要求メッセージの必須項目が存在しない • ARTSHeader が存在しない
FileNotFound	スタイルシートが存在しません。	• Stylesheet の名前が指定されていない • Stylesheet にアクセスできない
DeviceNotFound	デバイスが存在しません。	出力先のデバイスの稼動状況制御に問題があり、アクセスできない
DeviceNoAction	デバイスの動作はありませんでした。	要求メッセージにデバイス制御が存在しない
TransformFailed	Stylesheet の適用でエラーが発生しました。	• Stylesheet が不正、POSLog が不正 • Stylesheet 適用後のデータ形式が不正
FaultConfig	対象の制御デバイスが定義されていません。	• 要求メッセージが不正 • OFSC ドライバーの設定が不正

デバイスで発生するエラー

Code	メッセージ	原因
EPTR_AUTOMATICAL	自動復帰が可能なエラーが発生しました。	高密度印字の連続印刷
EPTR_COVER_OPEN	プリンターのカバーが開いています。	カバーが開いている
EPTR_CUTTER	デバイスで問題が発生しています。	カッターに異物がある
EPTR_MECHANICAL	デバイスで問題が発生しています。	メカ駆動中のエラーなど
EPTR_INITIALIZING	プリンターが初期化されています。	ドライバーがデバイスの初期化中
EPTR_REC_EMPTY	用紙がありません。	用紙が無い
EPTR_RESTART	デバイスの電源が OFF、または通信できません。	デバイスの電源が OFF、または別要因で通信できない
EPTR_TOOBIG	ビットマップの幅が広すぎるか、あるいは大きすぎます。	Stylesheet に問題がある
EPTR_UNRECOVERABLE	復帰不能なエラーが発生しました。	低電圧
ESTATS_ERROR	スタティスティクスがリセットできませんでした。	Stylesheet に問題がある
EPTR_BADFORMAT	ファイル形式が不正です。	Stylesheet に問題がある。
EX_BADPORT	ポートが不正か、デバイスが接続されていません。	• 登録情報不正 • デバイスの電源が OFF • デバイスが接続されていない • システム上に他のデバイス制御ソフトウェアが存在している
EX_INVALID_VALUE	パラメータが不正です。	Stylesheet に問題がある
EX_NOTSUPPORTED	指定されたステーションはマウントされていません。	Stylesheet に問題がある
EPTR_POWER_OFF	デバイスの電源が入っていません。	デバイスの電源が OFF
EX_TIMEOUT	タイムアウトしました。	• 時間内に処理が完了できなかった • 1 メソッドで大量のデータを送った • Stylesheet に問題がある

メンテナンス

本章では、OFSC ドライバーを使ったシステムのデバイスの入れ替え方法や、設定ファイルについて説明しています。

> **参考**　本章の説明で使われている CATALINA_HOME とは、Tomcat のインストール先フォルダーです。

デバイスの入れ替え

デバイスの入れ替えは、サンプルプログラムを例に説明しています。

> **参考**　デバイスを置き換える場合、変更する必要はありません。

デバイスを追加

キッチンプリンターを 1 台追加する場合、以下の手順で追加します。

1 SetupPOS でデバイスを登録します。
ここでは、論理デバイス名を〝KitchenPrinter2〟で登録します。デバイスの登録方法は、〝OFSC インストールマニュアル〟のデバイス登録を参照してください。

2 WebService の設定を変更します。
手順 1 で論理デバイス名を〝OFSC インストールマニュアル〟のデバイス登録名リストの名称を使う場合、手順 3 に進んでください。該当しない場合、以下の手順に従って設定を変更してください。

1. OFSCStack WebService の設定を変更します。

   OFSCStack.xml に書かれているデバイス名（青字）を変更します。

   - Java 環境

```
<DeviceInfo name="KitchenPrinter2"> <!--SetupPOS の登録名 -->
    <Class>jp.co.epson.ofsc.xmlposio.OFSCDeviceOutputXMLPOS</Class>
    <Port>http://localhost:8080/axis2/services/XMLPOSWebService1</Port>
    <Interval>1000</Interval>
</DeviceInfo>
```

   - .NET 環境

```
<DeviceInfo name="KitchenPrinter2"> <!--SetupPOS の登録名 -->
    <Class>jp.co.epson.ofsc.xmlposio.OFSCDeviceOutputXMLPOS</Class>
    <Port>http://localhost/XMLPOS/XMLPOSWebService1.asmx</Port>
    <Interval>1000</Interval>
</DeviceInfo>
```

2. XMLPOSBridge WebService の設定を変更します。

XMLPOSBridgeWebService.xml に書かれているデバイス名（青字）を変更します。

- Java 環境

```
<DeviceInfo name="KitchenPrinter2"><!--SetupPOS の登録名 -->
    <Class>jp.co.epson.xmlpos.webservice.XMLPOSWebService1</Class>
</DeviceInfo>
```

- .NET 環境

```
<DeviceInfo name="KitchenPrinter2"><!--SetupPOS の登録名 -->
    <Class>jp.co.epson.xmlpos.webservice.XMLPOSWebService1</Class>
</DeviceInfo>
```

参考

- OFSCStack.xml のディレクトリーは以下のとおりです。
  * Tomcat の場合 ：
    CATALINA_HOME¥webapps¥axis2¥WEB-INF¥classes¥jp¥co¥epson¥ofsc
  * IIS の場合 ：
    システムドライブ :¥Inetpub¥wwwroot¥OFSCStack
- XMLPOSBridgeWebService.xml のディレクトリーは以下のとおりです。
  * Tomcat の場合 ：
    CATALINA_HOME¥webapps¥axis2¥WEB-INF¥classes¥jp¥co¥epson¥xmlpos
  * IIS の場合 ：
    システムドライブ :¥Inetpub¥wwwroot¥XMLPOS

3 スタイルシートを選択します。

追加するデバイス用のスタイルシートを、SampleStyleSheets フォルダーから選択し、stylesheet フォルダーにファイル名を変更してコピーします。

Tomcat 使用時：

CATALINA_HOME¥webapps¥ROOT¥SampleStyleSheets¥Cooking¥TM-T90¥Cooking.xsl
→ CATALINA_HOME¥webapps¥ROOT¥epsonsample¥stylesheet¥Cooking2.xsl

IIS 使用時：

システムドライブ :¥Inetpub¥wwwroot¥SampleStyleSheets¥Cooking¥TM-T90¥Cooking.xsl
→ システムドライブ :¥Inetpub¥wwwroot¥epsonsample¥stylesheet¥Cooking2.xsl

4 アプリケーションのプログラムを修正します。

driverControl.js (driverControl.vbs) から、追加したデバイスの役割に合わせてソースコードを修正します。
以下は、追加する KitchenPrinter2 を KitchenPrinter の迂回処理に使う場合の修正例です。

> **参考**
>
> driverControl.js (driverControl.vbs) のディレクトリーは以下のとおりです。
> - Tomcat の場合 :
>   CATALINA_HOME¥webapps¥ROOT¥epsonsample
> - IIS の場合 :
>   システムドライブ :¥Inetpub¥wwwroot¥epsonsample

- Java 環境

```
// KitchenPrinter で印刷する調理指示単品伝票を設定
'    <Device>\n' + // 論理デバイス名を設定
'      <Name>KitchenPrinter</Name>\n' +
       // 印刷スタイルのスタイルシートを設定
'      <Stylesheet>./webapps/ROOT/epsonsample/stylesheet/Cooking.xsl</Stylesheet>\n' +
       // レスポンス解析用のスタイルシートを設定
'      <ResponseStylesheet>./webapps/ROOT/epsonsample/stylesheet/PrinterResponse.xsl</ResponseStylesheet>\n' +
       // デバイスのタイムアウト時間を設定 (ms)
'      <Timeout>10000</Timeout>\n' +
       // 調理指示単品伝票の印刷が失敗した場合に迂回処理をするデバイスを設定
'      <DeviceRejected>\n' +
         // 論理デバイス名を設定
'        <Name>KitchenPrinter2</Name>\n' +
         // 印刷スタイルのスタイルシートを設定
'        <Stylesheet>./webapps/ROOT/epsonsample/stylesheet/Cooking2.xsl</Stylesheet>\n' +
         // レスポンス解析用のスタイルシートを設定
'        <ResponseStylesheet>./webapps/ROOT/epsonsample/stylesheet/PrinterResponse.xsl</ResponseStylesheet>\n' +
         // デバイスのタイムアウト時間を設定 (ms)
'        <Timeout>10000</Timeout>\n' +
'      </DeviceRejected>\n' +
'    </Device>\n';
```

- .NET 環境

```
    // KitchenPrinter で印刷する調理指示単品伝票を設定
    ofsc = ofsc & "<Device>" & vbCrLf
        ' 論理デバイス名を設定
        ofsc = ofsc & "<Name>KitchenPrinter</Name>" & vbCrLf
        ' 印刷スタイルのスタイルシートを設定
        ofsc = ofsc & "<Stylesheet>./epsonsample/stylesheet/Cooking.xsl</Stylesheet>" & vbCrLf
        ' レスポンス解析用のスタイルシートを設定
        ofsc = ofsc & "<ResponseStylesheet>./epsonsample/stylesheet/PrinterResponse.xsl</ResponseStylesheet>" & vbCrLf
        ' デバイスのタイムアウト時間を設定 (ms)
        ofsc = ofsc & "<Timeout>10000</Timeout>" & vbCrLf
        ' 調理指示単品伝票の印刷が失敗した場合に迂回処理をするデバイスを設定
        ofsc = ofsc & "<DeviceRejected>" & vbCrLf
            ' 論理デバイス名を設定
        ofsc = ofsc & "<Name>KitchenPrinter2</Name>" & vbCrLf
            ' 印刷スタイルのスタイルシートを設定
        ofsc = ofsc & "<Stylesheet>./epsonsample/stylesheet/Cooking2.xsl</Stylesheet>" & vbCrLf
            ' レスポンス解析用のスタイルシートを設定
        ofsc = ofsc & "<ResponseStylesheet>./epsonsample/stylesheet/PrinterResponse.xsl</ResponseStylesheet>" & vbCrLf
            ' デバイスのタイムアウト時間を設定 (ms)
        ofsc = ofsc & "<Timeout>10000</Timeout>" & vbCrLf
        ofsc = ofsc & "</DeviceRejected>" & vbCrLf
    ofsc = ofsc & "</Device>" & vbCrLf
```

デバイスの変更

TM-T88V をキッチンプリンターとして使用する場合、以下の手順で変更します。
（初期値では TM-T88V はレシートプリンターとして使用しています。）

1 SetupPOS でデバイスを登録します。

ここでは、TM-T88V の論理デバイス名を〝KitchenPrinter2〟で登録します。デバイスの登録方法は、〝OFSC インストールマニュアル〟のデバイス登録を参照してください。
なお、論理デバイス名を〝OFSC インストールマニュアル〟のデバイス登録名リストの名称を使う場合、デバイスの変更は完了です。

2 アプリケーションのプログラムを修正します。

driverControl.js (driverControl.vbs) から、追加したデバイスの役割に合わせてソースコードを修正します。
以下は、KitchenPrinter2 に変更した TM-T88V を、KitchenPrinter の迂回処理に使う場合の修正例です。

参考

driverControl.js (driverControl.vbs) のディレクトリーは以下のとおりです。

- Tomcat の場合：
  CATALINA_HOME¥webapps¥ROOT¥epsonsample
- IIS の場合：
  システムドライブ:¥Inetpub¥wwwroot¥epsonsample

- Java 環境

```
// KitchenPrinter で印刷する調理指示単品伝票を設定
'    <Device>\n' + // 論理デバイス名を設定
'      <Name>KitchenPrinter</Name>\n' +
       // 印刷スタイルのスタイルシートを設定
'      <Stylesheet>./webapps/ROOT/epsonsample/stylesheet/Cooking.xsl</Stylesheet>\n' +
       // レスポンス解析用のスタイルシートを設定
'      <ResponseStylesheet>./webapps/ROOT/epsonsample/stylesheet/PrinterResponse.xsl</ResponseStylesheet>\n' +
       // デバイスのタイムアウト時間を設定 (ms)
'      <Timeout>10000</Timeout>\n' +
       // 調理指示単品伝票の印刷が失敗した場合に迂回処理をするデバイスを設定
'      <DeviceRejected>\n' +
         // 論理デバイス名を設定
'        <Name>KitchenPrinter2</Name>\n' +
         // 印刷スタイルのスタイルシートを設定
'        <Stylesheet>./webapps/ROOT/epsonsample/stylesheet/Cooking2.xsl</Stylesheet>\n' +
         // レスポンス解析用のスタイルシートを設定
'        <ResponseStylesheet>./webapps/ROOT/epsonsample/stylesheet/PrinterResponse.xsl</ResponseStylesheet>\n' +
         // デバイスのタイムアウト時間を設定 (ms)
'        <Timeout>10000</Timeout>\n' +
'      </DeviceRejected>\n' +
'    </Device>\n';
```

- .NET 環境

```
// KitchenPrinter で印刷する調理指示単品伝票を設定
ofsc = ofsc & "<Device>" & vbCrLf
    ' 論理デバイス名を設定
    ofsc = ofsc & "<Name>KitchenPrinter</Name>" & vbCrLf
    ' 印刷スタイルのスタイルシートを設定
    ofsc = ofsc & "<Stylesheet>./epsonsample/stylesheet/Cooking.xsl</Stylesheet>" & vbCrLf
    ' レスポンス解析用のスタイルシートを設定
    ofsc = ofsc & "<ResponseStylesheet>./epsonsample/stylesheet/PrinterResponse.xsl</ResponseStylesheet>" & vbCrLf
    ' デバイスのタイムアウト時間を設定 (ms)
    ofsc = ofsc & "<Timeout>10000</Timeout>" & vbCrLf
    ' 調理指示単品伝票の印刷が失敗した場合に迂回処理をするデバイスを設定
    ofsc = ofsc & "<DeviceRejected>" & vbCrLf
        ' 論理デバイス名を設定
    ofsc = ofsc & "<Name>KitchenPrinter2</Name>" & vbCrLf
        ' 印刷スタイルのスタイルシートを設定
    ofsc = ofsc & "<Stylesheet>./epsonsample/stylesheet/Cooking2.xsl</Stylesheet>" & vbCrLf
        ' レスポンス解析用のスタイルシートを設定
    ofsc = ofsc & "<ResponseStylesheet>./epsonsample/stylesheet/PrinterResponse.xsl</ResponseStylesheet>" & vbCrLf
        ' デバイスのタイムアウト時間を設定 (ms)
    ofsc = ofsc & "<Timeout>10000</Timeout>" & vbCrLf
    ofsc = ofsc & "</DeviceRejected>" & vbCrLf
 ofsc = ofsc & "</Device>" & vbCrLf
```

3 スタイルシートを選択します。

手順 1 で論理デバイス名を〝OFSC インストールマニュアル〟のデバイス登録名リストの名称を使う場合、手順 3 に進んでください。該当しない場合、以下の手順に従って設定を変更してください。

追加するデバイス用のスタイルシートを、SampleStyleSheets フォルダーから選択し、stylesheet フォルダーにファイル名を変更してコピーします。

Tomcat 使用時：

CATALINA_HOME¥webapps¥ROOT¥SampleStyleSheets¥Cooking¥TM-T88V¥Cooking.xsl
→ CATALINA_HOME¥webapps¥ROOT¥epsonsample¥stylesheet¥Cooking2.xsl

IIS 使用時：

システムドライブ:¥Inetpub¥wwwroot¥SampleStyleSheets¥Cooking¥TM-T88V¥Cooking.xsl
→ システムドライブ:¥Inetpub¥wwwroot¥epsonsample¥stylesheet¥Cooking2.xsl

デバイスを削除

フロアプリンターを削除して、そのプリンターの役割をキッチンプリンターにする場合、フロアプリンターを以下の手順で削除します。

1 スタイルシートを選択します。

削除されたデバイスに代用するデバイスのスタイルシートを、SampleStyleSheets フォルダーから選択し、stylesheet フォルダーにファイル名を変更してコピーします。

Tomcat 使用時：

CATALINA_HOME¥webapps¥ROOT¥SampleStyleSheets¥Cooking¥TM-T90¥Cooking.xsl
→ CATALINA_HOME¥webapps¥ROOT¥epsonsample¥stylesheet¥Cooking2.xsl

IIS 使用時：

システムドライブ:¥Inetpub¥wwwroot¥SampleStyleSheets¥Cooking¥TM-T90¥Cooking.xsl
→ システムドライブ:¥Inetpub¥wwwroot¥epsonsample¥stylesheet¥Cooking2.xsl

2 アプリケーションのプログラムを修正します。

driverControl.js (driverControl.vbs) から、削除したデバイスの役割を割り当てたソースコードを修正します。

- ″FloorPrinter″ を ″KitchenPrinter″ に変更
- 並列処理になっているソースコードを、逐次処理に変更
- 迂回処理になっているソースコードは削除

> **参考**
>
> driverControl.js (driverControl.vbs) のディレクトリーは以下のとおりです。
> - Tomcat の場合：
>   CATALINA_HOME¥webapps¥ROOT¥epsonsample
> - IIS の場合：
>   システムドライブ:¥Inetpub¥wwwroot¥epsonsample

設定情報

設定情報は、サンプルプログラムを例に説明しています。

タイムアウト

デバイス制御に失敗した場合、指定したタイムアウト時間が経過するまでは、指定した時間間隔でデバイス制御をリトライします。リトライ中にデバイス制御が成功した場合、レスポンスは成功を返します。

> **参考**
>
> - 以下の場合、タイムアウト処理されます。
> * タイムアウト時間とリトライが重なった場合
> * リトライ後に、タイムアウトまでの残り時間に比べて、リトライ間隔が大きい場合
> - タイムアウト時間を、リトライの時間間隔に満たない時間に指定した場合、リトライ処理は行いません。

タイムアウトの指定方法

以下のファイルを編集します。

- アプリケーションのプログラム
- OFSCStack.xml

アプリケーションのプログラム

タイムアウト時間を設定します。アプリケーションのプログラムの〝DeviceExecuteRequest/Device/Timeout〟(青字)を編集します。

記述例：タイムアウト時間を 10000 ミリ秒に指定する場合

- Java 環境

```
'    <Name>KitchenPrinter</Name>\n' +
'    <Stylesheet>./webapps/ROOT/epsonsample/stylesheet/Cooking.xsl</Stylesheet>\n' +
'    <ResponseStylesheet>./webapps/ROOT/epsonsample/stylesheet/PrinterResponse.xsl</ResponseStylesheet>\n' +
'    <Timeout>10000</Timeout>\n' +
```

- .NET 環境

```
ofsc = ofsc & "<Name>KitchenPrinter</Name>" & vbCrLf
ofsc = ofsc & "<Stylesheet>./epsonsample/stylesheet/Cooking.xsl</Stylesheet>" & vbCrLf
ofsc = ofsc & "<ResponseStylesheet>./epsonsample/stylesheet/PrinterResponse.xsl</ResponseStylesheet>" & vbCrLf
ofsc = ofsc & "<Timeout>10000</Timeout>" & vbCrLf
```

> **参考**
>
> サンプルプログラムでは、スタイルシートを設定するスクリプトファイルの driverControl.js (.NET 環境では、driverControl.vbs)を提供しています。

OFSCStack.xml

リトライの時間間隔を設定します。OFSCStack.xml に書かれている〝DeviceInfo/Interval〟(青字)を編集します。

記述例:リトライの時間間隔を 1000 ミリ秒に指定する場合

- Java 環境

```
<DeviceInfo name="KitchenPrinter">
    <Class>jp.co.epson.ofsc.xmlposio.OFSCDeviceOutputXMLPOS</Class>
    <Port>http://localhost:8080/axis2/services/XMLPOSWebService14</Port>
    <Interval>1000</Interval>
</DeviceInfo>
```

- .NET 環境

```
<DeviceInfo name="KitchenPrinter">
    <Class>jp.co.epson.ofsc.xmlposio.OFSCDeviceOutputXMLPOS</Class>
    <Port>http://localhost/XMLPOS/XMLPOSWebService14.asmx</Port>
    <Interval>1000</Interval>
</DeviceInfo>
```

参考

OFSCStack.xml のディレクトリーは以下のとおりです。

- Tomcat の場合:
  CATALINA_HOME¥webapps¥axis2¥WEB-INF¥classes¥jp¥co¥epson¥ofsc
- IIS の場合:
  システムドライブ:¥Inetpub¥wwwroot¥OFSCStack

スタイルシート

アプリケーションのプログラムに OFSC 機器標準機器規格に準拠した XML ファイルにデバイス情報で、スタイルシート名を記述して指定します。スタイルシート名の指定は、絶対パス、相対パスどちらでも指定できますが、URI 指定はできません。

記述例：

- Java 環境

```
'    <Name>KitchenPrinter</Name>\n' +
'    <Stylesheet>./webapps/ROOT/epsonsample/stylesheet/Cooking.xsl</Stylesheet>\n' +
'    <ResponseStylesheet>./webapps/ROOT/epsonsample/stylesheet/PrinterResponse.xsl</ResponseStylesheet>\n' +
'    <Timeout>10000</Timeout>\n' +
```

- .NET 環境

```
ofsc = ofsc & "<Name>KitchenPrinter</Name>" & vbCrLf
ofsc = ofsc & "<Stylesheet>./epsonsample/stylesheet/Cooking.xsl</Stylesheet>" & vbCrLf
ofsc = ofsc & "<ResponseStylesheet>./epsonsample/stylesheet/PrinterResponse.xsl</ResponseStylesheet>" & vbCrLf
ofsc = ofsc & "<Timeout>10000</Timeout>" & vbCrLf
```

相対パスを使用する場合のホームディレクトリーは以下のとおりです。

- Tomcat の場合：　CATALINA_HOME（Tomcat のインストールディレクトリー）
- IIS の場合：　　システムドライブ:¥Inetpub¥wwwroot

> **参考**　サンプルプログラムでは、スタイルシートを設定するスクリプトファイルの driverControl.js（.NET 環境では、driverControl.vbs）を提供しています。

OFSC Stack ライブラリの設定（OFSCStack.xml）

OFSC Stack ライブラリ（OFSCStack.xml）では、デバイスやログの情報を設定します。
青字部分が、設定を編集する箇所です。

> **参考**
>
> OFSCStack.xml のディレクトリーは以下のとおりです。
> - Tomcat の場合：
>   CATALINA_HOME¥webapps¥axis2¥WEB-INF¥classes¥jp¥co¥epson¥ofsc
> - IIS の場合：
>   システムドライブ:¥Inetpub¥wwwroot¥OFSCStack

DeviceInfo

デバイスの情報を設定します。

/@name

デバイス名を設定します。

```
<DeviceInfo name="KitchenPrinter">
    <Class>jp.co.epson.ofsc.xmlposio.OFSCDeviceOutputXMLPOS</Class>
</DeviceInfo>
```

/Class

出力クラス名を設定します。

```
<DeviceInfo name="KitchenPrinter">
    <Class>jp.co.epson.ofsc.xmlposio.OFSCDeviceOutputXMLPOS</Class>
</DeviceInfo>
```

/Port

デバイスの出力先 URL を設定します。

- Java 環境

```
<DeviceInfo name="KitchenPrinter">
    <Class>jp.co.epson.ofsc.xmlposio.OFSCDeviceOutputXMLPOS</Class>
    <Port>http://localhost:8080/axis2/services/XMLPOSWebService14</Port>
</DeviceInfo>
```

- .NET 環境

```
<DeviceInfo name="KitchenPrinter">
    <Class>jp.co.epson.ofsc.xmlposio.OFSCDeviceOutputXMLPOS</Class>
    <Port>http://localhost/XMLPOS/XMLPOSWebService14.asmx</Port>
</DeviceInfo>
```

/Interval

タイムアウト時間内に行う、リトライ処理の時間間隔（100 ～ 60000 ミリ秒）を設定します。
初期値は 1000 ミリ秒です。また、指定された値が不正な場合、初期値が適用されます。
（タイムアウトの詳細は、49 ページ「タイムアウト」を参照してください。）

- Java 環境

```
<DeviceInfo name="KitchenPrinter">
    <Class>jp.co.epson.ofsc.xmlposio.OFSCDeviceOutputXMLPOS</Class>
    <Port>http://localhost:8080/axis2/services/XMLPOSWebService14</Port>
    <Interval>1000</Interval>
</DeviceInfo>
```

- .NET 環境

```
<DeviceInfo name="KitchenPrinter">
    <Class>jp.co.epson.ofsc.xmlposio.OFSCDeviceOutputXMLPOS</Class>
    <Port>http://localhost/XMLPOS/XMLPOSWebService14.asmx</Port>
    <Interval>1000</Interval>
</DeviceInfo>
```

GlobalInfo

ログの情報を設定します。ログの詳細は、54 ページ「ログの取得方法」を参照してください。

/TraceMode

ログの出力を設定します。初期値は 1 です。

- 0：出力しない
- 1：出力する

```
<GlobalInfo>
    <TraceMode>1</TraceMode>
    <TraceFileName>OFSCStack</TraceFileName>
</GlobalInfo>
```

/TraceFileName

出力するログのファイル名を設定します。出力されたログファイル名は、指定したログファイル名に日付、識別文字列、拡張子が付加されます。

```
<GlobalInfo>
    <TraceMode>1</TraceMode>
    <TraceFileName>OFSCStack</TraceFileName>
</GlobalInfo>
```

注意

- <TraceFileName> が設定されていない場合、ログは出力されません。
- <TraceFileName> に日本語を設定した場合は、UTF-8 で保存してください。

ログの取得方法

ログの取得には以下のファイルを設定します。

- OFSCStack.xml
- OFSCStackWebService.xml
- XMLPOSBridgeWebService.xml

参考

- OFSCStack.xml、OFSCStackWebService.xml のディレクトリーは以下のとおりです。
  * Tomcat の場合 ：
    CATALINA_HOME¥webapps¥axis2¥WEB-INF¥classes¥jp¥co¥epson¥ofsc
  * IIS の場合 ：
    システムドライブ :¥Inetpub¥wwwroot¥OFSCStack
- XMLPOSBridgeWebService.xml のディレクトリーは以下のとおりです。
  * Tomcat の場合 ：
    CATALINA_HOME¥webapps¥axis2¥WEB-INF¥classes¥jp¥co¥epson¥xmlpos
  * IIS の場合 ：
    システムドライブ :¥Inetpub¥wwwroot¥XMLPOS

■ ログ出力の設定

ログ出力の有無は、<TraceMode> で設定します。

- 1： 出力します。
- 0： 出力しません。

■ ログファイル名の設定

出力するログファイル名は、任意で設定できます。<TraceFileName> で設定します。
各設定ファイルの初期値は以下のとおりです。

- OFSCStack.xml ⇨ OFSCStack
- OFSCStackWebService.xml ⇨ OFSCStackWS
- XMLPOSBridgeWebService.xml ⇨ XMLPOSBridge

```
<GlobalInfo>
    <TraceMode>1</TraceMode>
    <TraceFileName>OFSCStack</TraceFileName>
</GlobalInfo>
```

注意

- <TraceFileName> が設定されていない場合、ログは出力されません。
- <TraceFileName> に日本語を設定した場合は、UTF-8 で保存してください。

ログの出力

■ ログの出力先

Tomcat 使用時： CATALINA_HOME¥logs
IIS 使用時： システムドライブ :¥Inetpub¥wwwroot¥OFSCStack¥AppData

■ 出力されたログのファイル名

各設定ファイルで設定したファイル名の末尾に、出力日付が付加されます。
以下は、各設定ファイルのログファイル名が、初期値の場合の出力例です。

- OFSCStack.xml ⇨ OFSCStack_2009_03_11stack.log
- OFSCStackWebService.xml ⇨ OFSCStackWS_2009_03_11stackWS.log
- XMLPOSBridgeWebService.xml ⇨ XMLPOSBridge_2009_03_11brg.log
  XMLPOSBridge_2009_03_11brgWS.log