

# *Browser Mobile*

---

## 開発ガイド

|  |           |
|--|-----------|
| <b>目次1. はじめに</b> .....                   | <b>1</b>  |
| 1.1 表記について.....                          | 1         |
| <b>2. 準備</b> .....                       | <b>2</b>  |
| 2.1 Biz/Browser のインストール.....             | 2         |
| <b>3. Biz/Designer 開発手引き</b> .....       | <b>3</b>  |
| 3.1 目的.....                              | 3         |
| 3.2 サンプルプログラムの概要.....                    | 3         |
| 3.2.1 サンプルプログラムを実行するために必要な環境.....        | 3         |
| 3.2.2 画面レイアウト・機能一覧・システム構成概念図.....        | 4         |
| 3.2.2.1 画面レイアウト.....                     | 4         |
| 3.2.2.2 機能一覧.....                        | 6         |
| 3.2.2.3 システム構成概念図.....                   | 7         |
| 3.2.2.4 サンプルプログラムのファイル構成.....            | 7         |
| 3.3 準備と設定.....                           | 8         |
| 3.4 サンプルプログラムのポイント.....                  | 9         |
| 3.4.1 モバイル用アプリケーションの一般的な注意事項.....        | 9         |
| 3.4.2 データ一覧表示方法.....                     | 11        |
| 3.5 実装方法解説.....                          | 13        |
| 3.5.1 共通.....                            | 13        |
| 3.5.2 ログイン機能.....                        | 14        |
| 3.5.3 メニュー機能.....                        | 16        |
| 3.5.4 販売支援機能.....                        | 17        |
| 3.5.4.1 商品一覧画面.....                      | 17        |
| 3.5.4.1.1 商品一覧機能.....                    | 18        |
| 3.5.4.1.2 ローン計算機能.....                   | 24        |
| 3.5.4.2 商品情報画面.....                      | 26        |
| 3.5.5 売上一覧機能.....                        | 30        |
| 3.5.5.1 店舗プルダウンリスト、商品カテゴリプルダウンリスト表示..... | 31        |
| 3.5.5.2 データ一覧機能.....                     | 32        |
| 3.5.5.2.1 業務ロジック.....                    | 33        |
| 3.5.5.2.2 表示ロジック.....                    | 38        |
| 3.5.6 在庫商品登録機能.....                      | 41        |
| <b>4. 参考資料</b> .....                     | <b>44</b> |
| 4.1 バーコード.....                           | 44        |
| 4.1.1 社員マスタ.....                         | 44        |
| 4.1.2 商品マスタ.....                         | 45        |
| 4.1.3 在庫伝票マスタ.....                       | 49        |
| テーブル定義.....                              | 50        |
| 4.2.....                                 | 50        |
| 4.2.1 ER 図.....                          | 50        |
| 4.2.2 テーブル一覧.....                        | 50        |

## 1. はじめに

Biz/Browser Mobile、Biz/Designer をご利用いただき、誠に有難うございます。  
本ガイドは、Biz/Browser Mobile の基本的な使用方法について理解していただくことを目的として作成致しました。

本製品の導入に向けて本ガイドが、皆様のお役に立つことを心より願っております。

- 本ガイドは Biz/Browser Mobile(以下、Biz/Browser)を使った Web システムの構築方法を具体的な例を示しながら説明しています。
- Biz/Browser(PC 版)の位置付けと Biz/Designer の基本的な使用方法につきましては、「導入ガイド」をご参照ください。
- 詳細情報は「Biz/Designer オンラインマニュアル」の「Biz/Designer オペレーションガイド」、「Biz/Browser CRS マニュアル」、および「追補マニュアル」をご覧ください。
- 本ガイド および、付属のソフトウェアに関する所有権、知的財産権、その他全ての権利 および、権限は、アクシソフト株式会社が所有しています。
- 本ガイド および、付属のソフトウェアは使用許諾契約書に記載されている内容に反して使用または複製することはできません。
- 本ガイドに記載されている内容 および、付属のソフトウェアの仕様は、将来予告なく変更することがあります。
- 付属のソフトウェアを運用した結果の影響について、一切責任を負いかねますのでご了承ください。
- Windows は米国マイクロソフト社の登録商標です。
- その他の製品名は各社の商標または登録商標です。

### 1.1 表記について

- 画面について
  - 操作中表示されている画面例は、必要な部分のみ載せています。
- ヒントについて
  - 操作方法や、設定内容に関する参考情報は、ヒントマークで表示します。
- 注意について
  - 特に注意が必要な情報については注意マークで表示します。



## 2. 準備

### 2.1 Biz/Browser のインストール

端末にプログラムのセットアップを行います。



インストールの詳細は「追補マニュアル」の「セットアップ」をご覧ください。

## 3. Biz/Designer 開発手引き

### 3.1 目的

サンプルプログラムの理解、実行を通して、モバイル端末を対象としたプログラム開発のポイントを習得します。

### 3.2 サンプルプログラムの概要

「家電量販店システム」です。家電量販店の業務で想定した、主に以下の3つの機能からなります。  
([]内は機能を実現している画面名です。)

#### 販売支援機能

店頭で販売員が顧客に商品の説明を行い、注文を受け付ける機能。  
商品を選択し[商品情報画面]、それらをバスケットに入れて精算します[商品一覧画面]。  
在庫の確認、値引き限度額の確認が行えます。また、ローン計算機能があります。

#### 売上管理機能

本部バイヤーが各店舗を周り、製品の販売状況を確認することができる機能。  
店舗名、商品カテゴリを選択し、対象商品の平均販売数、その店舗の販売数、最も販売数の多い店舗名を一覧で表示します[売上一覧画面]。

#### 入在庫管理機能

倉庫で入庫担当者が入庫伝票の通り商品が入庫されたかどうかを確認する機能  
入庫伝票を検索し、検索された伝票の情報(入庫日、取引先、商品一覧)を表示します。  
商品とその数量を確認後、入庫完了登録を行います。[入庫商品登録画面]

その他、ログイン機能[ログイン画面]、メニュー機能[メニュー画面]があります。

#### 3.2.1 サンプルプログラムを実行するために必要な環境

本書では、IIS を Web サーバとし、ホームディレクトリにファイル共有設定を行います。このホームディレクトリにサンプルプログラムファイルをコピーします。

Biz/Designer をインストールした PC から、このファイルのプロジェクトを開きます。

モバイルクライアントからは無線 LAN 等を用いて、http で Web サーバに接続します。

#### Web サーバ、開発用 PC のハードウェア

- Windows NT、Windows 2000、Windows XP Professional のいずれかがインストールされたコンピュータ

#### ソフトウェア

- |                    |                            |
|--------------------|----------------------------|
| ・ Biz/Designer     | プログラム閲覧用                   |
| ・ Biz/Browser      | プログラム実行用                   |
| ・ IIS              | Web サーバ用 (サーバアプリは ASP を使用) |
| ・ Microsoft Access | データベース操作用                  |

※本書では、IIS のホームディレクトリを C:\¥Inetpub¥wwwroot で設定したものと説明しております。

### 3.2.2 画面レイアウト・機能一覧・システム構成概念図

#### 3.2.2.1 画面レイアウト

・ログイン画面

ログイン 終了

社員番号

パスワード

・メニュー画面

業務メニュー 戻る

社員名: 池東 太郎

・商品一覧画面

販売支援 - バasket 戻る

1 / 2

| Del | 商品名         | 数    | 価格      |
|-----|-------------|------|---------|
| Del | CAMEC720    | 2    | 100,400 |
| Del | Cyber-T1    | 1    | 57,300  |
| Del | EXILEP600   | 5    | 366,000 |
| Del | FinePooF700 | 3    | 135,000 |
| Del | HOTPIX5500  | 1    | 85,700  |
| 合計  | 788,700     | 修正合計 | 788,700 |

ローン計算 頭金  再

分割数  X 月払金額

・商品情報画面

販売支援 - 商品情報 戻る

商品コード

商品名

在庫数

単価

値引き限度

---

価格

数量

金額

メニュー画面

・売上一覧画面

メニュー  
画面

店舗別・カテゴリ別 売上順位 [戻る](#)

池袋東

| 商品名           | 平均 数▼ | Top店    |
|---------------|-------|---------|
| Cyber-T2A     | 11.8  | 18 池袋東店 |
| HOTPD5500     | 4.3   | 17 池袋東店 |
| CAMEC720      | 12.5  | 16 新宿店  |
| FinePooF700   | 3.5   | 14 池袋東店 |
| DXZDIGITAL500 | 11.5  | 14 新宿店  |
| CAMEC760      | 3.3   | 13 池袋東店 |
| EXILEP600     | 3.3   | 13 池袋東店 |
| DXZDIGITAL600 | 10.5  | 13 渋谷西店 |

・入庫商品登録画面

メニュー  
画面

入庫出庫管理 - 登録 [戻る](#)

伝票No

取引先

入荷日

金額  訂正

| No. | 商品コード   | 商品名       | 数量  |
|-----|---------|-----------|-----|
| 1   | DSC-U40 | CyberU    | 15  |
| 2   | EK-S20  | EXILE-S20 | 400 |
| 3   | E320-KK | m-30DIGIT | 10  |
| 4   | DXIDL   | DXZDIGITA | 500 |
| 5   | DSC-P43 | CyberP43  | 30  |

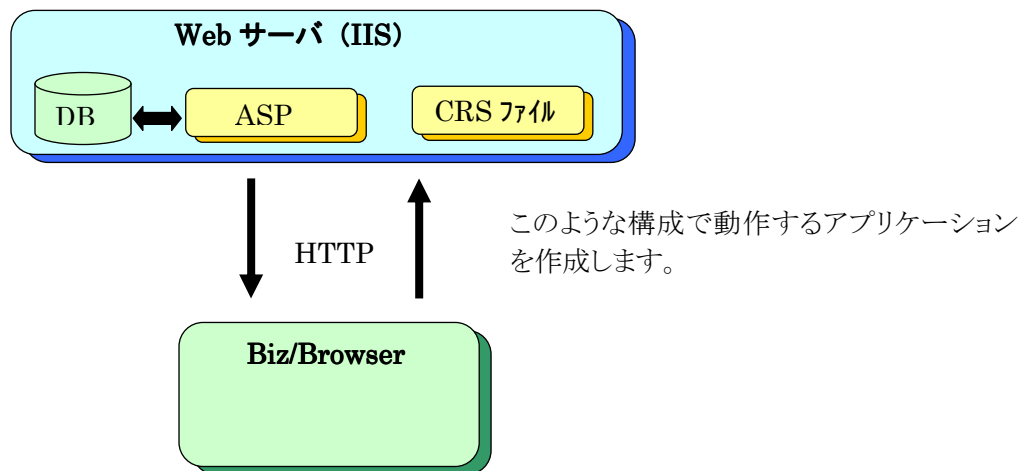
1 / 3

3.2.2.2 機能一覧

| 主機能   | 画面名    | 機能名                          | 機能内容   |
|-------|--------|------------------------------|--|
| ログイン  | ログイン   | ログイン                         | ログインIDとパスワードで認証する。認証された場合のみメニュー画面に遷移する。  |
| メニュー  | メニュー   | ボタンの表示制御                     | ログインユーザによって、使用可能な機能を制限する(ボタンを押しなくする。<br><br>○ユーザの権限と表示するボタン<br>販売員・・・販売支援機能のみ<br>スーパーバイザ・・・全機能<br>倉庫担当・・・入出庫管理機能   |
| 販売支援  | 商品一覧   | 商品追加                         | 商品情報画面に遷移する。   |
|       |        | 商品一覧、ローン計算                   | 追加された商品の情報の一覧などが表示される。<br>表示項目は削除ボタン、商品名、数量、金額。<br>削除ボタンを押すと、確認Msg表示後、対象の1行が削除される。<br>商品名をクリックすると、商品情報画面に遷移する(ここで商品の変更が可能)。<br>数量は変更可能。変更結果が金額と合計額に反映される。<br>・合計<br>金額の合計が表示される。<br>・修正後合計<br>初期値は合計額。修正可能。<br>・頭金、分割数、月払金額<br>ローン計算で設定した値が表示される。初期値は分割数は10。他は0。 |
|       |        | 前の5件を表示<br>後の5件を表示<br>ページ数表示 | 一覧に表示する情報の内、現在表示されているものより前、もしくは後の5件の情報を表示する。<br>また、その際表示している部分が全ページ中の何ページに当たるのかを表示する。  |
|       |        | 分割数、月払金額                     | 分割数、月払金額を変更すると、頭金が修正合計を元に自動的に計算される。  |
|       |        | 清算                           | 確認メッセージ表示後、サーバに購入情報を送ったメッセージを表示する。(実際はサーバに情報を送らない)。商品一覧等の項目は初期化され、メニューに戻る。   |
|       |        | 戻る                           | メニュー画面に遷移する。   |
|       | 商品情報   | 商品(B)                        | 入力された商品コードから、商品の情報を検索する<br>商品名に、商品コードの商品名と、同じカテゴリの商品名を表示<br>在庫数、単価、値引き限度を表示<br>価格に単価、数量に1、金額に価格×数量を表示  |
|       |        | 価格、数量                        | 金額、数量を変更すると、金額が自動的に計算される。  |
|       |        | 追加                           | 購入商品一覧画面に遷移する。その際選択した商品が一覧に追加される。  |
|       |        | 戻る                           | 購入商品一覧画面に遷移する。   |
| 売上管理  | 売上一覧   |                              | 店舗選択プルダウンリスト、カテゴリ選択プルダウンリスト、売上一覧が表示される。  |
|       |        | 店舗選択、カテゴリ選択                  | 選択したカテゴリに含まれる商品の、販売数平均、選択した店舗の販売数、1位の店舗の一覧を表示する。<br>※一覧に表示される内容は、独立した情報。販売支援で清算した情報は反映されない。<br>-----<br>商品名・平均・数・Top店を一行づつ表示   |
|       |        | ソート順変更                       | 表の"数"をクリックすると、販売数でソートして一覧を再表示する。クリックするたびに昇順/降順が交互に変わる。   |
|       |        | 戻る                           | メニュー画面に遷移する。   |
| 入出庫管理 | 入庫商品登録 | 入庫伝票No(B)                    | 入力された入庫伝票Noから伝票情報を取り込む<br>取引先名、入荷日・入荷金額を表示<br>入荷修正金額には、合計金額と同じ値段を表示<br>-----<br>連番・商品コード・商品名・数量を一行づつ明細表示   |
|       |        | 前の5件を表示<br>後の5件を表示<br>ページ数表示 | 一覧に表示する情報の内、現在表示されているものより前、もしくは後の5件の情報を表示する。<br>また、その際表示している部分が全ページ中の何ページに当たるのかを表示する。  |
|       |        | 登録                           | 情報を登録する(実際はサーバに情報を送らない)。メニュー画面に遷移する。   |
|       |        | 戻る                           | メニュー画面に遷移する。   |



## 3.2.2.3 システム構成概念図



## 3.2.2.4 サンプルプログラムのファイル構成

|                     |                   |
|---------------------|-------------------|
| CRS ファイル            |                   |
| G_const.crs         | 定数定義データ領域定義       |
| G_staff.crs         | ログイン社員データ受け渡し領域定義 |
| G_basket.crs        | 商品一覧データ受け渡し領域定義   |
| kadenlogin.crs      | ログイン画面            |
| menu.crs            | メニュー画面            |
| basket.crs          | 販売支援 - 商品一覧画面     |
| product.crs         | 販売支援 - 商品情報画面     |
| salesmanagement.crs | 売上一覧画面            |
| warehousing.crs     | 入庫商品登録画面          |

|            |        |
|------------|--------|
| ASP        |        |
| select.asp | 汎用検索機能 |
| sales.asp  | 売上検索機能 |

|               |        |
|---------------|--------|
| DB            |        |
| kadenshop.mdb | データベース |

※ASP から ADO で接続しています。

※参考資料として、4.2 テーブル定義書 (E/R 図) があります。

### 3.3 準備と設定

#### ■プログラムの配置

プログラムを実行するために、以下の設定を行います。

・IIS のホームディレクトリ直下に「kadenshop」ディレクトリを作成し、拡張子が「crs」、「asp」のファイルをコピーします。

※IIS のホームディレクトリが C:\¥Inetpub¥wwwroot で無い場合は、asp ファイル内に記述されているパスを修正して下さい。

・C:\¥kadenshop ディレクトリを作成し、そこに kadenshop.mdb ファイルをコピーします。

※別のディレクトリにコピーした場合は、asp ファイル内に記述されているパスを修正して下さい。

#### ■プロジェクト作成

Biz/Designer からプログラムを参照、実行するために、以下の設定を行います。

- ①メニューバーの[ファイル]－[新規プロジェクト]を選択します。
- ②[空のプロジェクト]を選択し、[OK]ボタンをクリックします。
- ③表示された[空のプロジェクト作成]ダイアログで、プロジェクト名を設定します。  
ここでは[**kadenshop**]と入力します。
- ④作成フォルダは共有している Web サーバの IIS のルートディレクトリのパスを設定して、[OK]ボタンをクリックします。
- ⑤メニューバーの[プロジェクト]－[プロジェクトへ追加]を選択します。
- ⑥表示される CRS ファイルを全て選択して、「開く」をクリックします。



概要は「導入ガイド」をご覧ください。詳細情報は「Biz/Designer オンラインマニュアル」の「Biz/Designer オペレーションガイド」および「Biz/Browser CRS マニュアル」をご覧ください。

#### ■基本設定

メニューバーの [プロジェクト] － [設定] で [プロジェクトの設定] ダイアログが表示されますので [基本設定] タブを選択します。

- ①プロジェクトにマッピングする URL  
プロジェクトにマッピングする URL を指定します。  
要求ファイルがプロジェクト内に存在しない場合、ここで指定した URL にあるファイルを探しにいきます。  
"http://[Web サーバの完全修飾ドメイン名 or IP アドレス]/kadenshop" と設定します。
- ②スタートアップ CRS ファイル  
アプリケーションの起点を指定します。"kadenlogin.crs" と指定します。
- ③スタートアップ CRS ファイルから実行する  
②の設定を有効にします。チェックを付けます。

## 3.4 サンプルプログラムのポイント

### 3.4.1 モバイル用アプリケーションの一般的な注意事項

#### ■ CPU、メモリの性能を考慮したプログラミング

PC に搭載されている CPU、メモリなどの性能と比較して、モバイル端末の環境はかなり劣ります。このため、PC では動作スピードで問題にならないコードでも、モバイル端末では性能が出ないことがあります。従って、パワーがないことを考慮したプログラミングを行う必要があります。例えば、ループ処理のプログラミングでメモリ取得をループ内で行わずに、ループ処理の前に行うなど、一般的な効率を考慮したプログラムのテクニックを使用するという配慮です。

#### ■ フォームの表示サイズ

モバイル端末の場合、PC と比べて QVGA と画面サイズが小さいため、大きなフォームを作成するのは好ましくありません。

QVGA の画面サイズから、Form の Width, Height の最大値は以下の式によって計算できます。

Form.Width

= 端末の画面の幅 - 2 \* 2(エッジの幅 \* 2)

Form.Height

= 端末の画面の高さ  
 - 2 \* 2(エッジの幅 \* 2)  
 - 20(OS のタスクバーの高さ)                   ... (\*)  
 - 19(ステータスバーの高さ)  
 - 41(Biz/Browser のメニューバー:WindowsCE 機のみ)

(\*) OS の設定を”自動的に隠す”にすれば 0 になります。

例えば、画面の幅が 240、高さが 320 の場合、上の式により Form の Width, Height はそれぞれ 236, 240 となります。(サンプルプログラムはこのサイズで作成されています)。

また ShowMenu メソッドによりメニューやステータスバーを非表示にする場合サイズの変化を考慮する必要があります。

#### ■ 画面構成

モバイル端末は、画面表示速度が PC より遅いため、多くのオブジェクトを表示する画面を設計する場合は、あらかじめ実機で速度を確認することをおすすめします。

#### ■ バーコード対応

端末にバーコードリーダーが付属しており、キーボードをエミュレートできるソフトが付属している場合があります。その設定で読み取り文字の最後に改行コードを付けることが可能な場合、読み取り後の処理を自動化できます。

入力欄の OnTouch イベント内に読み取り後の処理を記述します。

### ※例

商品情報画面の商品コード入力欄

```
TextBox:String PRODUCT_CODE {  
    TabIndex = 0;  
    Function OnTouch(e) {  
        /* バーコード読み取りモジュールが最後に改行を送った場合、  
        指定した商品コードの商品を取得、表示 */  
        ^.getProducts(^.PRODUCT_CODE.Value);  
    }  
}
```

### ■ スクロールバーの幅、TextBox の文字の幅

Form 等のスクロールバーや TextBox 等は、デバッグ時に PC 版 BizBrowser で表示される幅と、モバイル版で表示される幅が異なることがあります。

緻密な画面設計を行う場合は、実機で確認することをおすすめします。

### 3.4.2 データ一覧表示方法

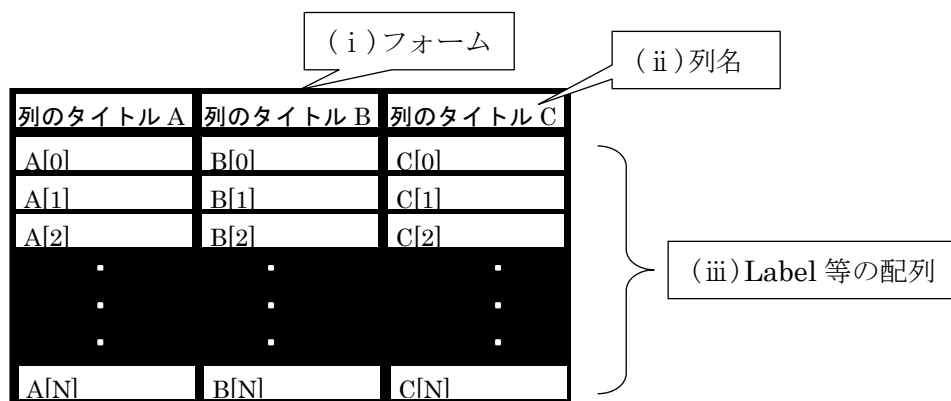
データの一覧を表示したい場合、PC 版ではスプレッドオブジェクトを利用するのが便利です。しかしモバイル版には存在しません。

そのため、サンプルプログラムではその代わりの機能を2種類の方法で実装しています。ここでは、その設計概要を示します。

- ①Label 等のオブジェクトの配列を横に並べる方法  
 フォーム上に列のタイトルを水平方向に配置します。  
 データを表示するラベルの配列を垂直方向に並べます。

#### 特徴

次頁の②の方法と比較した場合、表示が早い



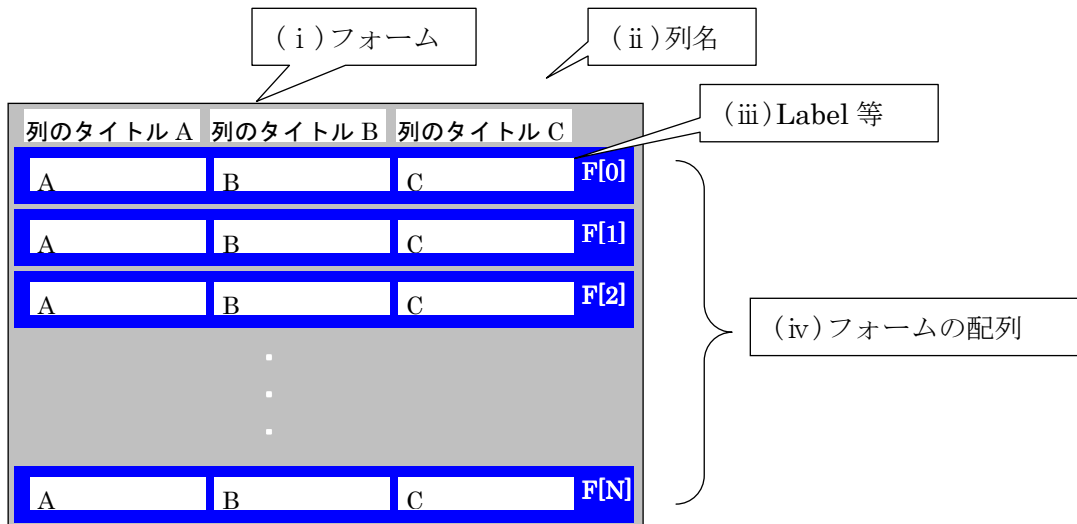
※商品一覧画面、入庫商品登録画面で採用しています。

※サンプルプログラムではより表示を早くするために、表示するデータの件数を制限しています。

- ②1フォーム上にいくつかのオブジェクトを作成し、そのフォームの配列を縦に並べる方法  
フォーム上に列のタイトルを水平方向に配置します。  
その上にフォームの配列を垂直方向に並べます。  
データを表示するラベルをそのフォーム上に水平方向に配置します。

特徴

- スプレッドオブジェクトの構造 (Spread, SpreadRow, SpreadColumn) に近いため、実装方法を理解しやすい



※売上一覧画面で採用しています。

## 3.5 実装方法解説

機能毎にポイントとなる部分を説明します。

### 3.5.1 共通

#### ■ 共通定数

|       |             |
|-------|-------------|
| 画面名   | なし          |
| ファイル名 | G_const.crs |

サンプルプログラムでは、画面、オブジェクトの背景色を統一しています。これらの背景色のデータ定義は、このファイルのグローバル名前空間のオブジェクトに設定しています。

他に売上一覧で使用する店舗情報(shop\_data)、商品カテゴリ情報(category\_data)を、CSVDocument オブジェクトで定義しています。またこれらに値をセットする関数(setShop\_category())も定義しています。

このファイルは他の全ファイルで import しています。  
その他の共通定数も追加した方がよいでしょう。

#### ■ 社員情報保持オブジェクト

|       |             |
|-------|-------------|
| 画面名   | なし          |
| ファイル名 | G_staff.crs |

ログイン後のいくつかの機能で参照する、ログインした社員の権限、店舗等を保持するオブジェクトです。

#### ■ 共通データ取得用サーバプログラム

|       |            |
|-------|------------|
| 画面名   | なし         |
| ファイル名 | select.asp |

売上一覧機能以外で、データベースからデータを取得する際に用いる ASP です。

SQL の select 文を受け取り、結果を CSV 形式で返します。

その CSV を CSVDocument オブジェクトの get メソッドで取得する方法で、本サンプルプログラムは統一しています。



通常、SQL などの業務ロジックはサーバ側に記述することが多いと思われませんが、この ASP を用いる場合は、crs ファイルに SQL を記述しています。

3.5.5.2.1 章で説明する sales.asp では、サーバ側に SQL を記述しています。

※内部の処理については、sales.asp の説明をご参照下さい。

### 3.5.2 ログイン機能

|       |                |
|-------|----------------|
| 画面名   | ログイン画面         |
| ファイル名 | kadenlogin.crs |

初期表示画面です。

本システムを利用する社員が、社員番号とパスワードを入力し、認証します。

#### 主な機能

ログインボタンが押下されたとき、主に以下の処理行っています。

- ・ 社員番号、パスワードが一致する社員がいるかどうか検索する SQL 作成、実行(\*)
- ・ 結果、データが無かった場合はログイン失敗のメッセージを表示
- ・ データが存在した場合は、以下の情報をグローバル名前空間のオブジェクトに設定し、メニュー画面に遷移
  - 社員の情報
  - 社員の権限(販売支援機能、売上一覧機能、入在庫管理機能が利用可能かどうか)



詳細はソースプログラムをご覧ください。  
以降、ソースの主要な部分の抜粋を示します。一般的なプロパティも省略する場合があります。

#### ログインボタンを押下したときの主な処理

```

CSVDocument csvobj;

Button login {
OnTouch ( e ) {
  /* sql 文生成 */
  var sql_txt = new String;
  sql_txt = "selectSQL=";

  sql_txt += "SELECT";
  sql_txt += " SP.SHOP_CODE,";
  sql_txt += " SP.SHOP_NM,";
  sql_txt += " ST.STAFF_NO,";
  sql_txt += " ST.STAFF_NM,";
  sql_txt += " ST.AUTH_SALES,";
  sql_txt += " ST.AUTH_CHECK,";
  sql_txt += " ST.AUTH_STORE_IO";
  sql_txt += " FROM";
  sql_txt += " STAFF ST, SHOP SP ";
  sql_txt += " WHERE";
  sql_txt += " ST.SHOP_CODE = SP.SHOP_CODE";
  sql_txt += " AND ST.STAFF_NO = " + kadenlogin.id.value + " AND ST.PASSWD=" + kadenlogin.pass.value + """;

  /* 社員番号とパスワードで検索 */
  ^.csvobj.get("select.asp", sql_txt);

```

<次ページへ続く>





社員テーブルを検索して、結果を CSVDocument オブジェクトに格納します。



4.2 テーブル定義書もご覧下さい。



”^”は特別に、this の親オブジェクトを指す名前です。すなわち相対パスでオブジェクトを指定することができます。

詳細情報は「Biz/Designer オンラインマニュアル」の Biz/Browser CRS マニュアル-「オブジェクトの操作」の”オブジェクト名の検索”をご覧ください。

〈続き〉

```

/* 社員番号とパスワードで検索 */
if (^.csvobj.rows == 0) {
    MessageBox("該当する社員情報はありません","ログインエラー",$Ok + $IconExclamation);
} else {
    /* 社員情報をグローバル変数を設定 */
    G_staff.shop_code = ^.csvobj.getCell(0,0);
    G_staff.shop_nm = ^.csvobj.getCell(0,1);
    G_staff.staff_no = ^.csvobj.getCell(0,2);
    G_staff.staff_nm = ^.csvobj.getCell(0,3);
    /* MSAccess の Yes/No 型の戻り値は"True"/"False" */
    G_staff.auth_sales = ^.csvobj.getCell(0,4) == "False" ? False : True;
    G_staff.auth_check = ^.csvobj.getCell(0,5) == "False" ? False : True;
    G_staff.auth_store_io = ^.csvobj.getCell(0,6) == "False" ? False : True;
    /* 売上一覧画面の店舗、商品カテゴリプルダウンリストの要素をここで取得しておく */
    if (G_staff.auth_check) G_const.setShop_category();
    /* 権限を取得できたら、メイン画面に遷移 */
    kadenlogin.Delete();
    //.Get("menu.crs");
}
}
}

```

売上一覧機能を使用可能の社員の場合は、売上一覧画面の店舗カテゴリプルダウンリスト、商品カテゴリプルダウンリストの要素をここで取得しておきます。

結果は G\_const に格納します。



売上一覧画面を表示するたびに同じ処理が実行されるのを避けるために、ここで1回実行しています。

### 3.5.3 メニュー機能

|       |          |
|-------|----------|
| 画面名   | メニュー画面   |
| ファイル名 | menu.crs |

ログイン画面から遷移します。

社員名を表示します。

各機能に遷移するボタンを表示します。

G\_staff に保持されたログインした社員の権限により、各機能に遷移するボタンの活性(押下できるかどうか)を制御します。

ソースプログラムで特記すべき項目はありません

### 3.5.4 販売支援機能

#### 3.5.4.1 商品一覧画面

|       |            |
|-------|------------|
| 画面名   | 商品一覧画面     |
| ファイル名 | basket.crs |

メニュー画面から遷移します。

主な機能

データ一覧機能

- ・商品一覧データの表示(\*)
- ・商品の削除、変更(\*)
- ・前のデータ、次のデータへの表示領域の移動、ページ数表示(\*)
- ・数量変更による価格の再計算
- ・商品の追加画面への遷移
- ・追加済全商品の合計額、及びそれを修正する入力欄(修正合計)の表示
- ・ローン計算機能(\*)  
修正合計金額から、分割数を指定した時の月払金額、頭金の計算
- ・お買上(※現状、データベースへの登録は行わず、追加済の商品データを全て削除してメニュー画面に戻る。)

以下、(\*)の機能について解説します。

### 3.5.4.1.1 商品一覧機能

まず、商品一覧関連のオブジェクトからご説明します。

商品一覧、商品グローバル変数定義(G\_basket.crs)

```
Record G_basket {
  /*商品一覧*/
  CSVDocument basket_data;

  /* 追加商品 */
  Record product{
    String PRODUCT_CODE; /* 商品コード */
    String PRODUCT_NM; /* 商品名 */
    Number UNIT; /* 価格 */
    Number COUNT; /* 数量 */
    Number basket_index; /* 修正時に用いる、元の商品データの basket_data の index */
  }
}
```

CSVDocument オブジェクトの商品一覧(以下、商品一覧グローバル変数)、Record オブジェクトの追加商品(以下、商品グローバル変数)を定義しています。

商品グローバル変数は、商品一覧画面と商品情報画面との間で1件の商品の情報をやりとりするために用います。

その他、商品グローバル変数を空にする関数、商品グローバル変数を商品一覧グローバル変数に追加する関数があります。

※同一商品が既に商品一覧グローバル変数に存在する場合、価格は上書きされ、数量は追加されません。

## 商品一覧画面(basket.crs)

```

import G_const;
import G_basket;
Form: String basket {
  /* 現在表示しているページ数 */
  Number page = 1;
  /* 最大ページ数 */
  Number maxpage = 1;
  /* 代替スプレッドの定数 */
  Number spX = 5; /* 始点の X 座標 */
  Number spY = 45; /* 始点の Y 座標 */
  Number spW = 220; /* 全体の幅 */
  Number spH = 90; /* 全体の高さ */
  Number spLen = 5; /* 要素数 */
  Number spBtnY = 22; /* スプレッドの上端から前、次ボタンの上端までの距離 */
}

```

商品一覧、商品グローバル変数定義(G\_basket.crs)をインポートします。  
 データの一部を表示するので、現在表示しているページ数、最大ページ数を保持します。  
 また、一覧表示のサイズ等の変数を宣言します。これらは一覧内のオブジェクトから使用します。  
 本画面では、一覧表機能、ローン計算機能を解説します。

## 商品一覧表

商品一覧表のオブジェクトは以下の様に実装しています。

```

/* 代替スプレッドシート */
Form spBase {

  /* タイトル行表示 */
  Label: String title[4] {
    Layout = $HORIZONTAL;
    /* 幅を指定 */
    title[0].Width = 26;
    title[1].Width = 110;
    title[2].Width = 25;
    title[3].Width = ^.^ .spW - title[0].Width - title[1].Width - title[2].Width;

    /* タイトルを指定 */
    title[0].Value = "Del";
    title[1].Value = "商品名";
    title[2].Value = "数";
    title[3].Value = "価格";
  }
}

```

spBase フォームが、商品一覧表示方法①の“(i)フォーム”に相当します。  
 title ラベルが、同“(ii)列名”に相当します。

<次ページへ続く>



```
/* 削除ボタン */
Button:String btnDelete[basket.spLen] {
    Title = "Del";
    X = ^.title[0].X;
    Y = ^.title[0].Y + 15 + 1;
    Width = ^.title[0].Width;
    Height = int(^.spH / ^.spLen);
    VerticalAlign = $CENTER;
    LayoutMargin = 1;
}

/* 商品コード */
String PRODUCT_CODE[basket.spLen];

/* 商品名 */
Button:String btnProduct[basket.spLen]

/* 価格 */
Number numUnitPrice[basket.spLen];

/* 数量 */
NumberEdit neNum[basket.spLen] {
    IMEMode = $ALPHA;
    UseChange = True;
    Format = "90";
}

/* 価格 (小計) */
Label:Number lblPrice[basket.spLen] {
    Format = "9,999,999";
}
}
```

上記オブジェクトが、商品一覧表示方法①の“(iii)Label 等の配列”に相当します。



削除ボタンの Value Type (Button:String の”String”) は設定しなくても問題ありません。

```
/* 現在のページ数 */
Label:String page_disp {
    /* 現在のページ数/最大ページ数を設定 */
    function setValue() {
        Value = str(^.page) + "/" + str(^.maxpage);
    }
}

/* 前へボタン */
Button:String previous {
    Title = "←";
    Function OnTouch(e) {
        /* 前のデータを表示 */
        if(^.page > 1) {
            ^.page--;
            /* 前のデータを設定（1つ目の引数は、そのページの先頭の行番号） */
            ^.setdata(G_basket.basket_data, ^.page, ^.spLen);
        }
    }
}

/* 次へボタン */
Button:String next {
    Title = "→";
    Function OnTouch(e) {
        /* 次のデータを表示 */
        if(^.page < ^.maxpage) {
            ^.page++;
            /* 次のデータを設定（1つ目の引数は、そのページの先頭の行番号） */
            ^.setdata(G_basket.basket_data, ^.page, ^.spLen);
        }
    }
}
```

一覧表の全ページ中の現在のページ、前へボタン、次へボタンオブジェクトです。  
前へボタン、次へボタンを押下することによってページを遷移する処理を記述しています。

## 一覧表のデータ表示関数

```
/* CSVDocumentObject、表示するページ数、1ページ当たりの行数を与えてデータを表示
表示されない商品コードも設定 */
function setdata(csvdataobj, page, splen) {
  for (var i = 0; i < splen; i++) {
    if ((splen * (page - 1) + i + 1) <= csvdataobj.Rows) {
      spBase.PRODUCT_CODE[i].Value = csvdataobj.getCell(splen * (page - 1) + i, 0);
      .
      .
      .
    } else {
      /* 表示すべきデータが無い部分は表示しない */
      spBase.PRODUCT_CODE[i].Value = "";
      .
      .
      .
    }
  }
}

/* 前、次ボタン表示制御 */
next.Active = True;
previous.Active = True;
if (page == 1) {
  previous.Active = False;
}
if (page == maxpage) {
  next.Active = False;
}

/* ページ数表示を更新 */
page_disp.setValue();
}
```

一覧表に表示するデータは商品一覧グローバル変数から取得します。表示するページ数、1ページ当たりの行数から表示するデータを取得します。

データが存在しない部分は空にします。

前へボタン、次へボタンの活性、ページ数を更新します。



## 商品の削除、変更

```

/* 配列 obj のイベントハンドラは、親の Form に1つだけ設ける(でないと、配列要素数分だけ出来てしまう) */
Function OnTouch(e) {
  /* クリックをしたオブジェクトを、Name で判定 */
  switch (e.From.Name) {
  case "btnDelete":
    /* 削除処理 */
    if (MessageBox(spBase.btnProduct[e.From.index].Title + " を削除してもよろしいですか?","削除確認",
    $YesNo + $IconQuestion) == YesSelected) {
      /* 削除 */
      G_basket.basket_data.deleteRow((page - 1) * spLen + e.From.index);
      /* 最大ページ数更新 */
      setMaxpage(G_basket.basket_data, spLen);
      /* 表示 */
      setdata(G_basket.basket_data, page, spLen);
      /* 合計額更新 */
      setSum(G_basket.basket_data);
    }
    break;
  case "btnProduct":
    /* 商品情報(変更)画面に遷移する */
    /* 修正する商品の商品コード、数量を設定 */
    G_basket.product.init();
    G_basket.product.PRODUCT_CODE = spBase.PRODUCT_CODE[e.From.index];
    G_basket.product.COUNT = spBase.neNum[e.From.index];
    G_basket.product.basket_index = (page - 1) * spLen + e.From.index;

    /* 商品情報画面に遷移 */
    basket.Delete();
    //Get("product.crs");
    break;
  }
}

```

※メモリ効率を下げないようにするため、イベントハンドラはボタン毎、つまり配列の要素数分だけ定義するのではなく、配列オブジェクトの上位に1つだけ定義します。



詳細情報は「Biz/Designer オンラインマニュアル」の Biz/Browser CRS マニュアル」-「配列オブジェクト」-「配列オブジェクトの表示」の「配列オブジェクトのイベントハンドラ」をご覧ください。

- ・削除機能

削除確認後、商品一覧グローバル変数から該当行を削除し、データを再表示します。

- ・変更機能

修正する商品の商品コード、数量、商品一覧グローバル変数での要素の位置を商品グローバル変数に設定します。その後、商品情報画面に遷移します。

## 3.5.4.1.2 ローン計算機能

```
/* 修正合計 */
NumberEdit sumchg {
    Function OnChange( e ) {
        /* ローンシミュレート (頭金を自動計算) */
        ^.loan_simulation( False );
    }
}

/* 分割数 */
NumberEdit INSTALLMENTS_NUM {
    Value = 10; /* 初期値は 10 分割 */
    Function OnChange( e ) {
        /* 値が 1 以上 99 以下かどうかチェック、修正 */
        if( Value <= 0 ) {
            Value = 1;
        } else if( Value > 99 ) {
            Value = 99;
        }
        /* ローンシミュレート (月払金額、頭金を自動計算) */
        ^.loan_simulation( False );
    }
}

/* 月払金額 */
NumberEdit INSTALLMENTS {
    Function OnChange( e ) {
        /* 値が 1 以上修正合計以下かどうかチェック、修正 */
        if( Value <= 0 ) {
            Value = 1;
        } else if( ^.sumchg.Value != 0 && Value > ^.sumchg.Value ) {
            Value = ^.sumchg.Value;
        }
        /* ローンシミュレート (分割数、頭金を自動計算) */
        ^.loan_simulation( True );
    }
}
```

対象となる項目の値が変更されたとき、まず、値のチェックを行います。  
その後、次に解説するローン計算関数を実行します。

<次ページへ続く>



## ローン計算関数

```
/* ローンシミュレート（自動計算） */  
function loan_simulation(at_chg_installments_flg) {  
    /* 修正合計が 0 の時は、処理を行わない */  
    if (sumchg.Value != 0) {  
        /* 金額が変更されたときのみ変更 */  
        if (DEPOSIT.Value + INSTALLMENTS.Value * INSTALLMENTS_NUM.Value != sumchg.Value) {  
            /* 月払金額が修正されたとき以外、月払金額自体を修正する */  
            if (!at_chg_installments_flg) {  
                INSTALLMENTS.Value = int(sumchg.Value / INSTALLMENTS_NUM.Value);  
            } else {  
                INSTALLMENTS_NUM.Value = int(sumchg.Value / INSTALLMENTS.Value);  
            }  
            DEPOSIT.Value = sumchg.Value - INSTALLMENTS.Value * INSTALLMENTS_NUM.Value;  
        }  
    }  
}
```

以下の式で月払金額、頭金を計算します。

月払金額 = (修正合計 / 分割数) の小数点以下切り捨て

頭金 = 修正合計 - 月払金額 \* 分割数

ただし月払金額を修正したときは、分割数、頭金を以下の式で計算します。

分割数 = (修正合計 / 月払金額) の小数点以下切り捨て

### 3.5.4.2 商品情報画面

|       |             |
|-------|-------------|
| 画面名   | 商品情報画面      |
| ファイル名 | product.crs |

商品一覧画面から遷移します。  
追加ボタンで遷移した場合は、商品の追加を行います。修正ボタン(商品一覧画面の商品名が Title のボタン)で遷移した場合は、商品の修正を行います。

#### 主な機能

- ・入力した商品コードからデータベースを検索し、商品情報を表示(\*)
- ・その商品と同一カテゴリの商品名を、商品名プルダウンリストに表示
- ・商品名プルダウンリストから商品を選択すると、選択した商品の商品情報を表示
- ・価格、数量変更による金額の再計算
- ・追加(変更)ボタン押下により、商品を商品一覧グローバル変数に追加し、商品一覧画面に遷移(\*)

以下、(\*)の機能について解説します。

#### 共通

```
import G_const;
import G_staff;
import G_basket;

Form product {
  /* 指定した商品と同じカテゴリの商品の一覧を格納する */
  /* 順に、商品コード、商品名、在庫数、単価、値引き限度額 */
  CSVDocument products;
```

商品一覧画面と商品情報を受け渡しするために、商品一覧、商品グローバル変数定義(G\_basket.crs)をimportします。  
指定した商品と同じカテゴリの商品の一覧を格納する CSVDocument の変数を宣言します。

## 商品情報検索関数

指定した商品コードの商品と同じカテゴリの商品を検索します。  
商品コードを入力したとき、及び商品一覧画面から変更ボタンを押下して遷移してきたときに実行されます。

```
/* 指定した商品コードの商品情報をサーバから取得、表示する */
function getProducts(pcode) {
    if (pcode == "") {
        MessageBox ("商品コードを入力して下さい");
    } else {
        /* sql 文生成 */
        var sql_txt = new String;
        var csvbuffer = new CSVDocument;
        sql_txt = "selectSQL=";
        sql_txt += "SELECT";
        sql_txt += " p.PRODUCT_CODE,";
        sql_txt += " p.PRODUCT_NM,";
        sql_txt += " s.STOCK_COUNT,";
        sql_txt += " p.UNIT,";
        sql_txt += " p.UNIT_LIMIT";
        sql_txt += " FROM";
        sql_txt += " PRODUCT_CATEGORY c,";
        sql_txt += " PRODUCT p,";
        sql_txt += " STOCK s";
        sql_txt += " WHERE";
        sql_txt += " c.PRODUCT_CATEGORY_CODE = p.PRODUCT_CATEGORY_CODE";
        sql_txt += " AND s.PRODUCT_CATEGORY_CODE = p.PRODUCT_CATEGORY_CODE";
        sql_txt += " AND p.PRODUCT_CATEGORY_CODE = ";
        sql_txt += " (SELECT PRODUCT_CATEGORY_CODE FROM PRODUCT";
        sql_txt += " WHERE (INSTRB(1, PRODUCT_CODE, "" + pcode + "", 0));";
        sql_txt += " AND LENB(PRODUCT_CODE) = LENB("" + pcode + ""))";
        sql_txt += " AND s.PRODUCT_CODE = p.PRODUCT_CODE";
        sql_txt += " AND s.SHOP_CODE = "" + G_staff.shop_code + """;
        sql_txt += " ORDER BY";
        sql_txt += " p.PRODUCT_NM";
        csvbuffer.get("select.asp", sql_txt);
        /* 指定した商品コードの商品が存在しない場合、その旨表示する */
        if (csvbuffer.rows == 0) {
            MessageBox ("指定した商品コードの商品は存在しません");
        } else {
            products.truncate();
            products << csvbuffer;
            dispProduct(pcode);
        }
    }
}
```

商品、商品カテゴリ、在庫テーブルを結合して情報を取得しています。  
在庫はログインした社員の所属している店舗のものです。また在庫数を登録していない商品は検索されません。(在庫数が0個の商品は検索されます。)

入力した商品コードが存在した場合のみ、CSVDocument に格納し、別の関数でそのデータを各オブジェクトに設定しています。

※Where 句の1行目と2行目の処理は、Access が大文字と小文字(他にはひらがなとカタカナ、半角と全角)を区別しないための対応です。

## 追加(変更)機能

```

Button ADD {
  Title = "追加";

  Function OnTouch(e) {
    /* 追加する商品があるかどうかチェック */
    if (^products.find(0, "0==" + ^PRODUCT_CODE.Value) < 0) {
      MessageBox(Title + "する商品がありません");
      ^PRODUCT_CODE.SetFocus();
    } else {
      /* 追加する商品がある場合、追加 */
      /* グローバル変数に商品の各値をセット */
      G_basket.product.PRODUCT_CODE = ^PRODUCT_CODE.Value;
      G_basket.product.PRODUCT_NM = ^PRODUCT_NM.PRODUCT_NMItem[^PRODUCT_NM.Value].Title;
      G_basket.product.UNIT      = ^CHGUNIT.Value;
      G_basket.product.COUNT     = ^COUNT.Value;

      /* 購入商品に追加商品を追加する */
      G_basket.addProduct();

      /* グローバル変数の商品の各値を削除 */
      G_basket.truncateproduct();

      /* バスケットに戻る */
      product.Delete();
      //.Get("basket.crs");
    }
  }
}

```

- ①商品コードの商品が、存在するかどうかチェックを行う。
- ②商品グローバル変数に登録／修正する商品の情報をセットする。
- ③商品グローバル変数の値を商品一覧グローバル変数に追加する。
- ④商品グローバル変数の値を削除する。
- ⑤商品一覧画面に遷移する。

以上が、販売支援機能の解説です。

### 3.5.5 売上一覧機能

|           |                     |
|-----------|---------------------|
| 画面名       | 売上一覧画面              |
| ファイル名     | salesmanagement.crs |
| asp ファイル名 | sales.asp           |

メニュー画面から遷移します。

#### 主な機能

店舗プルダウンリスト、商品カテゴリプルダウンリスト表示(\*)

売上一覧機能(\*)

- ・表示対象: 指定した店舗で販売された、指定した商品カテゴリに含まれる商品
- ・表示内容: 商品名、その商品の平均販売数、指定した店舗での販売数(ソート可能)、その商品の販売数が最も多い店舗名

以下、(\*)の機能について解説します。



## 3.5.5.1 店舗プルダウンリスト、商品カテゴリプルダウンリスト表示

```

/* 全店舗情報をサーバーから取得、店舗選択プルダウンリストを表示する */
function getShop() {
  if (G_const.shop_data.rows == 0) {
    print ("店舗が 1 件も存在しません¥n");
  } else {
    SHOP_NM.SHOP_NMItem.truncate();
    SHOP_NM.SHOP_NMItem.insert(G_const.shop_data.rows + 1);
    /* 先頭行に固定値を設定 */
    SHOP_NM.SHOP_NMItem[0].Title = "店舗";
    SHOP_NM.SHOP_NMItem[0].Value = "";
    for (var i = 1; i <= G_const.shop_data.rows; i++) {
      SHOP_NM.SHOP_NMItem[i].Title = G_const.shop_data.getCell(i - 1, 1); /* 店舗名を表示 */
      SHOP_NM.SHOP_NMItem[i].Value = G_const.shop_data.getCell(i - 1, 0); /* 値は店舗コード */
    }
    /* 先頭が初期値 */
    SHOP_NM.Value = 0;
  }
}

/* 全商品カテゴリ情報をサーバーから取得、商品カテゴリ選択コンボボックスを表示する */
function getProduct_category() {
  if (G_const.category_data.rows == 0) {
    print ("商品カテゴリが 1 件も存在しません¥n");
  } else {
    PRODUCT_CATEGORY_NM.PRODUCT_CATEGORY_NMItem.insert(G_const.category_data.rows + 1);
    /* 先頭行に固定値を設定 */
    PRODUCT_CATEGORY_NM.PRODUCT_CATEGORY_NMItem[0].Title = "商品カテゴリ";
    PRODUCT_CATEGORY_NM.PRODUCT_CATEGORY_NMItem[0].Value = "";
    for (var i = 1; i <= G_const.category_data.rows; i++) {
      PRODUCT_CATEGORY_NM.PRODUCT_CATEGORY_NMItem[i].Title = G_const.category_data.getCell(i - 1, 1); /* 商
品カテゴリ名を表示 */
      PRODUCT_CATEGORY_NM.PRODUCT_CATEGORY_NMItem[i].Value = G_const.category_data.getCell(i - 1, 0); /*
値は商品カテゴリコード */
    }
    /* 先頭が初期値 */
    PRODUCT_CATEGORY_NM.Value = 0;
  }
}

```

画面作成時に、ログイン画面で G\_const に格納しておいた店舗テーブル、商品カテゴリテーブルのデータを取得し、それぞれ店舗プルダウンリスト、商品カテゴリプルダウンリストを作成します。プルダウンリストの先頭は固定値です。



print 関数を利用して、デバッグ用のメッセージを「アウトプットビュー」の「実行結果」タブに出力することができます。

## 3.5.5.2 データ一覧機能

&lt;salesmanagement.crs&gt;

```

/* 指定した商品カテゴリコードの販売実績情報をサーバから取得、表示する */
function getProducts() {
    if (SHOP_NM.Value > 0 && PRODUCT_CATEGORY_NM.Value > 0) {
        /* リストの先頭以外が選択されたとき */
        /* sql 文条件生成 */
        var sql_txt = new String;
        sql_txt = "whereSQL=";
        sql_txt += "SHOP_CODE = " + SHOP_NM.SHOP_NMItem[SHOP_NM.Value].Value + "";
        sql_txt += " AND PRODUCT_CATEGORY_CODE = "
        sql_txt += " + PRODUCT_CATEGORY_NM.PRODUCT_CATEGORY_NMItem[PRODUCT_CATEGORY_NM] + """;

        /* 売上一覧検索・表示 */
        csvdata.truncate();
        csvdata.get("sales.asp", sql_txt);

        fsp.setData(csvdata);
    }
}

```

店舗プルダウンリスト、商品カテゴリプルダウンリストから、店舗、商品カテゴリが選択されたときに、販売実績情報をサーバから取得し表示します。

売上一覧でのみ用いる asp (sales.asp) があります。引数に店舗コード、商品カテゴリコードを含む SQL を渡しています。

本 asp について解説します。

売上一覧に表示するデータを3回に分けて取得しています。これらは一旦配列に格納し、最後に CSV 形式でクライアントに返します。

※配列の列は、順に以下のデータを意味します。

商品コード,商品名,平均販売数,販売数,Top 店

## 3.5.5.2.1 業務ロジック

商品コード、商品名、販売数の取得

<sales.asp>

```

baseSQL = "SELECT" & _
" PRODUCT_CODE," & _
" PRODUCT_NM," & _
" Sum(COUNT)" & _
" FROM" & _
" SALES_DETAIL" & _
" WHERE 0=0 And " & _
whereSQL & _
" GROUP BY" & _
" PRODUCT_CODE," & _
" PRODUCT_NM" & _
" ORDER BY" & _
" PRODUCT_CODE"

### SQL の実行
dbRS.Open dbCOM.CommandText,dbCN

'対象が 0 件の時は、空を返す
If dbRS.EOF Then
    Response.Write ""
    Response.End
End If
Do Until dbRS.EOF
    '商品コード,商品名,(平均販売数),販売数,(1 位の店舗)を設定
    output(i, 0) = dbRS.Fields(0)
    output(i, 1) = dbRS.Fields(1)
    output(i, 3) = dbRS.Fields(2)

    i = i + 1
    dbRS.MoveNext          '次のデータを読む
Loop

```

まず、商品コード、商品名、販売数を取得します。  
クライアントから送られてきた店舗コード、商品カテゴリコード(変数名:whereSQL)の条件で、その店舗かつその商品カテゴリに含まれる販売実績を、販売実績明細テーブルから取得します。

販売数は商品毎にグループ化した数量の合計です。  
結果は配列に格納します。

該当する販売実績が 0 件の時は空を返します。

※エラーチェック等の記述は省略して表示しています。

<次ページへ続く>



## 平均販売数の取得

```
### 平均販売数取得 SQL 作成
baseSQL = "SELECT" & _
" PRODUCT_CODE," & _
" SUM(COUNT)/(SELECT count(*) FROM shop) AS ave" & _
" FROM" & _
" SALES_DETAIL" & _
" WHERE 0 = 0 And PRODUCT_CODE in ("
For i = 0 To rows
  If output(i, 0) <> "" Then
    If i = rows Then
      baseSQL = baseSQL & "" & output(i, 0) & ""
    Else
      baseSQL = baseSQL & "" & output(i, 0) & ","
    End If
  End If
Next
baseSQL = Left(baseSQL, Len(baseSQL) - 1) 'カンマ削除
baseSQL = baseSQL & ")"

baseSQL = baseSQL & _
" GROUP BY" & _
" PRODUCT_CODE," & _
" PRODUCT_NM" & _
" ORDER BY" & _
" PRODUCT_CODE"

### SQL の実行
dbRS.Open dbCOM.CommandText,dbCN

Do Until dbRS.EOF
  '(商品コード),(商品名),平均販売数,(販売数),(1位の店舗)を設定
  '商品コードが一致しない場合はエラー
  If output(i, 0) = dbRS.Fields(0) Then
    output(i, 2) = dbRS.Fields(1)
  End If
  i = i + 1
  dbRS.MoveNext '次のデータを読む
Loop
```

先に取得した商品コードの商品の平均販売数を取得します。  
商品毎にグループ化した数量の合計値を店舗数で割って算出します。

結果は配列に格納します。

<次ページへ続く>



## Top 店の取得

```
### No.1 店舗取得 SQL 作成
baseSQL = "SELECT" & _
" PRODUCT_CODE," & _
" SHOP_NM," & _
" SUM(COUNT)" & _
" FROM" & _
" SALES_DETAIL" & _
" WHERE 0 = 0 And PRODUCT_CODE in ("

For i = 0 To rows
  If output(i, 0) <> "" Then
    If i = rows Then
      baseSQL = baseSQL & "" & output(i, 0) & ""
    Else
      baseSQL = baseSQL & "" & output(i, 0) & ","
    End If
  End If
Next
baseSQL = Left(baseSQL, Len(baseSQL) - 1) 'カンマ削除
baseSQL = baseSQL & ")" & _
" GROUP BY" & _
" PRODUCT_CODE," & _
" SHOP_NM" & _
" ORDER BY" & _
" PRODUCT_CODE"

### SQL の実行
dbRS.Open dbCOM.CommandText,dbCN
```

先に取得した商品コードの商品の数量が、最も多い店舗を取得します。  
まず、商品毎、店舗毎にグループ化した商品コード、店舗名、数量の合計値を取得します。

```
#### レコードセットからループでデータを取り出す
'1行目はテンポラリにデータを保存
i = 0
product_code_buf = dbRS.Fields(0)
no1shop_nm_buf   = dbRS.Fields(1)
no1shop_count_buf = dbRS.Fields(2)
output(i, 4) = no1shop_nm_buf
dbRS.MoveNext

Do Until dbRS.EOF
'商品コード),(商品名),(平均販売数),(販売数),1位の店舗を設定
'商品コードでソートされていることが条件
'1つ前の行と商品コードが同じ時は、売上数を比較し、大きい方に設定
If product_code_buf = dbRS.Fields(0) Then
  If no1shop_count_buf < dbRS.Fields(2) Then
    no1shop_nm_buf   = dbRS.Fields(1)
    no1shop_count_buf = dbRS.Fields(2)
    output(i, 4) = no1shop_nm_buf
  End If
Else
  '商品コードが変わったときは、新たにテンポラリに保存
  product_code_buf = dbRS.Fields(0)
  no1shop_nm_buf   = dbRS.Fields(1)
  no1shop_count_buf = dbRS.Fields(2)
  i = i + 1
  output(i, 4) = no1shop_nm_buf
End If

dbRS.MoveNext      '次のデータを読む
Loop
```

結果から、商品毎に各店舗の販売数を比較し、最大の店舗名を配列に格納します。

```

### CSV データにして返す
For i = 0 To rows
  If output(i, 0) <> "" Then
    For j = 1 To 4
      outputCSV = outputCSV & "" & output(i, j) & ""
      outputCSV = outputCSV & ","
    Next
    ### 最後のカンマを削除し、改行追加
    outputCSV = Left(outputCSV, Len(outputCSV) - 1) & vbCrLf
  End If
Next
Response.Write outputCSV
Response.End

```

結果を CSV 形式に変換してクライアントに返します。



ASP から Access のファイルへは ADO を用いて接続しています。OS の”データ ソース (ODBC)”の設定は不要です。

詳細は米国マイクロソフトの Web サイトをご覧ください。

```

Dim dbCN, dbRS, dbCOM

### コネクションオブジェクトを作り、データベースと接続する。
set dbCN = Server.CreateObject("ADODB.Connection")
dbCN.Open "provider=microsoft.jet.oledb.4.0;data source=C:\¥kadenshop¥kadenshop.mdb"

### コマンドオブジェクトを作り、それを実行して結果をレコードセットに格納
set dbCOM = Server.CreateObject("ADODB.Command")
set dbRS = Server.CreateObject("ADODB.RecordSet")
set dbCOM.ActiveConnection = dbCN

### select 文実行
dbCOM.CommandText = "select * from ..."
dbRS.Open dbCOM.CommandText, dbCN

dbRS.Close
dbCOM.Close
dbCN.Close
set dbRS = Nothing
set dbCN = Nothing
set dbCOM = Nothing

```

次に取得した売上一覧データを表示するロジックについて解説します。

## 3.5.5.2.2 表示ロジック

&lt; salesmanagement.crs &gt;

```
/* スクロールの幅+水平スクロールバーが出ないようにするためのマージン */
Number scrollbar_width = 20;

Form fsp {
  Scroll = $AUTO;

  Form fArray[salesmanagement.rows] {
    border = $FALSE;
    layoutMargin = 0;
    Scroll = $none;

    Label product_nm {} /* 商品名 */
    Label:Number sel_count_avg {} /* 平均販売数 */
    Label:Number sel_count {} /* 販売数 */
    Label first_shop {} /* Top 店 */
  }
}
```

売上一覧のオブジェクトです。

データ一覧表示方法の②の方法で作成されています。

fsp が (i) フォーム、fArray が (iv) フォームの配列、その中の4つの Label が (iii) Label 等に相当します。

<次ページへ続く>





## 一覧表のデータ表示関数

```
/* データ設定関数 */
function setData(csvDoc) {
    var border = new Number;
    if (csvDoc.rows < salesmanagement.rows) {
        border = csvDoc.rows;
    } else {
        border = salesmanagement.rows;
    }

    /* 表示データ set */
    for (var i = 0; i < border; i++) {
        fArray[i].product_nm.Value = csvDoc.getCell(i, 0);
        fArray[i].sel_count_avg.Value = val(csvDoc.getCell(i, 1));
        fArray[i].sel_count_avg.FgColor = $Black;
        fArray[i].sel_count.Value = val(csvDoc.getCell(i, 2));
        fArray[i].first_shop.Value = csvDoc.getCell(i, 3);
    }

    /* 不要な表示領域削除 */
    for (i = csvDoc.rows; i < salesmanagement.rows; i++) {
        /* 既に不要な表示領域が空になっている場合は処理を行わない */
        if (fArray[i].product_nm.Value == "") {
            break;
        }
        fArray[i].product_nm.Value = "";
        fArray[i].sel_count_avg.Value = 0;
        fArray[i].sel_count_avg.FgColor = fArray[i].sel_count_avg.BgColor;
        fArray[i].sel_count.Value = 0;
        fArray[i].first_shop.Value = "";
    }
}
}
```

売上一覧データを表示します。

まず表示する行数を求めます。

次に売上一覧のデータの入った csvDocument オブジェクトから1行ずつ値を設定します。

データが不要な行の値は、削除します。

## 販売数での表示の並べ替え

```

/* タイトルのラベルの OnClicked 関数 */
Function OnClicked(e) {
    /* "数"がクリックされたとき */
    if (e.From.index == 2) {
        ^.title[2].Value = ^.fsp.sortData(^.csvdata, 2, ^.title[2].Value);
        ^.fsp.setData(^.csvdata);
    }
}

-----

/* Form fsp の関数 */
/* 一覧を指定要素（今は"数"のみ）の昇順/降順、商品名の昇順で
並べ替える */
function sortData(csvDoc, col, disp) {
    /* col   : 並べ替え対象（ラベルの配列の添え字） */
    /* disp  : 変更前の状態を表すラベルの value */
    /* 戻り値 : 変更後の状態を表すラベルの value */
    if (csvDoc.rows == 0) return disp;

    var up_symbol = New String("▲");
    var down_symbol = New String("▼");

    if (disp != ("数" + down_symbol)) {
        /* 一覧を数の降順、商品名の昇順で並べ替える */
        csvDoc.sort(str(col) + "nd", "0");
        return "数" + down_symbol;
    } else {
        /* 一覧を数の昇順、商品名の昇順で並べ替える */
        csvDoc.sort(str(col) + "n", "0");
        return "数" + up_symbol;
    }
}

```

タイトルの“数”をクリックすることで、販売数でのソート、及びソート順の指定を行っています。

“数”がクリックされたかどうかは、e.From.index が 2 であることで判定できます。

売上一覧のデータの入った csvDocument オブジェクトをソートします。

ソート順は、降順(“数▼”)でない場合は降順に、それ以外の場合は昇順(“数▲”)にします。その後、それを元に一覧のデータを再表示します。

以上が、売上一覧機能の解説です。

### 3.5.6 入庫商品登録機能

|       |                 |
|-------|-----------------|
| 画面名   | 入庫商品登録画面        |
| ファイル名 | warehousing.crs |

メニュー画面から遷移します。

#### 主な機能

- ・入力した入庫伝票 No からデータベースを検索し、入庫伝票情報を表示(\*)
  - ヘッダー表示: 入荷日、入荷金額
  - 明細表示 : 明細 No、商品コード、商品名、数量
  - ※明細の一覧表示方法は、販売支援機能の商品一覧画面の追加済商品一覧と同じ
- ・登録(※現状、データベースへの登録は行わず、メニュー画面に戻る。)

以下、(\*)の機能について解説します。

```
TextBox:String WAREHOUSING_NO {

/* 伝票 NO から情報を取得 */
Function OnTouch( e ) {
    if (^WAREHOUSING_NO.Value == "") {
        MessageBox ("伝票 No を入力して下さい");
    } else {
        var csvdoc = new CSVDocument;
        /* sql 文生成 */
        var sql_txt = new String;

        sql_txt = "selectSQL=";
        sql_txt += "SELECT";
        sql_txt += "  TRADER_NM,";
        sql_txt += "  DELIVERY_DATE,";
        sql_txt += "  SUM";
        sql_txt += " FROM";
        sql_txt += "  WAREHOUSING";
        sql_txt += " WHERE ";
        sql_txt += "  WAREHOUSING_NO = " + ^WAREHOUSING_NO.Value + """;

        csvdoc.get("select.asp", sql_txt);
        if (csvdoc.rows == 0) {
            MessageBox ("指定した入庫伝票は存在しません");
        } else {
            ^TRADER_NM.Value      = csvdoc.getCell(0, 0);
            ^Delevery_date.Value = csvdoc.getCell(0, 1);
            ^Sum.Value            = val(csvdoc.getCell(0, 2));
            /* 入荷修正金額を入荷金額に更新 */
            ^Re_sum.Value        = ^Sum.Value;
        }
    }
}
```

入力された伝票 No で、入庫伝票テーブルを検索します。検索された取引先名、入荷日、金額を取得します。

<次ページへ続く>



```
/* 伝票明細を取得 */
sql_txt = "selectSQL=";
sql_txt += "SELECT";
sql_txt += "  DETAIL_NO,";
sql_txt += "  PRODUCT_CODE,";
sql_txt += "  PRODUCT_NM,";
sql_txt += "  PRODUCT_COUNT";
sql_txt += " FROM";
sql_txt += "  WAREHOUSING_DETAIL";
sql_txt += " WHERE ";
sql_txt += "  WAREHOUSING_NO = " + ^.WAREHOUSING_NO.Value + """;
sql_txt += " ORDER BY";
sql_txt += "  DETAIL_NO";

^.warehousing_slip_data.get("select.asp", sql_txt);
if (csvdoc.rows != 0) {
  /* 表示 (1 ページから) */
  ^.setMaxpage(^.warehousing_slip_data, ^.spLen);
  ^.setdata(^.warehousing_slip_data, 1, ^.spLen);
}
}
}
}
}
```

入力された伝票 No で、入庫伝票明細テーブルを検索します。検索された明細 No、商品コード、商品名、数量を取得します。  
取得した情報は、販売支援機能の商品一覧と同様の手法で一覧表示します。

以上が、入庫商品登録機能の解説です。

## 4. 参考資料

### 4.1 バーコード

PDA にはバーコードリーダーが付属している場合があります。それを用いるための社員番号、商品コード、入庫伝票 No のバーコードを用意いたしました。これらのデータはデータベースに登録されていません。

※コードは"Code39"です。

※バーコードスキャナードライバの設定は、以下の通りにして下さい。

出力フォーマット :Start/Stop コード無

チェックデジット :無し

チェックキャラクタ:出力する

#### 4.1.1 社員マスタ

| 社員番号      | 社員名   | バーコード   | 販売支援 | 売上管理 | 入出庫管理 |
|-----------|-------|---|------|------|-------|
| 200400001 | 池東 太郎 |   | ○    | ○    | ○     |
| 200400002 | 池東 花子 |  | ○    | ×    | ×     |
| 200400003 | 鈴木 一郎 |  | ×    | ×    | ○     |
| 200400004 | 新宿 二郎 |  | ○    | ○    | ○     |
| 200400005 | 池西 三郎 |  | ○    | ×    | ×     |
| 200400006 | 渋谷 四郎 |  | ○    | ○    | ×     |

## 4.1.2 商品マスタ

| 商品カテゴリ名            | 商品コード   | 商品名           | バーコード   |
|--------------------|---------|---------------|---|
| デジタルカメラ(400 万画素以上) | C760    | CAMEC760      |    |
|                    | D500    | IXZDIGITAL500 |    |
|                    | DSC-T1  | Cyber-T1      |    |
|                    | E5400   | HOTPIX5400    |    |
|                    | F710    | FinePooF710   |    |
|                    | P600    | EXILEP600     |   |
|                    | C720    | CAMEC720      |  |
|                    | D600    | IXZDIGITAL600 |  |
|                    | DSC-T2A | Cyber-T2A     |  |
|                    | E5500   | HOTPIX5500    |  |
|                    | F700    | FinePooF700   |  |

| 商品カテゴリ名           | 商品コード     | 商品名         | バーコード   |
|-------------------|-----------|-------------|---|
| デジタルカメラ(400万画素以下) | A340      | FinePooA340 |    |
|                   | DMC-LC70P | LAMIX-LC70  |    |
|                   | DSC-P43   | CyberP43    |    |
|                   | DSC-S4    | ZactiS4     |    |
|                   | E320-KK   | m-30DIGITAL |    |
|                   | IXYDL     | IXZDIGITALL |    |
|                   | A400      | FinePooA400 |  |
|                   | DMC-LC70W | LAMIX-LC70W |  |
|                   | DSC-P40   | CyberP40B   |  |
|                   | DSC-S3    | ZactiS3P    |  |
|                   | E280-TT   | T-28DIGITAL |  |



| 商品カテゴリ名                | 商品コード       | 商品名        | バーコード  |
|------------------------|-------------|------------|--|
| デジタルカメラ<br>(200 万画素以下) | COOLPIX2200 | HOTPIX2200 |    |
|                        | DMC-FZ2K    | LAMIX-FZ2  |    |
|                        | DSC-U40     | CyberU     |    |
|                        | EX-S20      | EXILE-S20  |    |
|                        | SV-AS10-A   | D-smapSV   |    |
|                        | COOLPIX2600 | HOTPIX2600 |    |
|                        | DMC-FZ03    | LAMIX-FZ03 |   |
|                        | DSC-U20     | CyberU20   |  |
|                        | EX-S10      | EXILE-S10  |  |
|                        | SV-AS20-B   | B-smapSV   |  |

| 商品カテゴリ名       | 商品コード     | 商品名         | バーコード   |
|---------------|-----------|-------------|---|
| テレビ<br>(フラット) | 21FX-01   | 21 型 FX-01  |  |
|               | 21T-M01   | 21 型-M01    |  |
|               | 21ZR8     | FOOT21ZR8   |  |
|               | AV-21K3   | FLATAV-21K3 |  |
|               | C-21B80S  | 21 型 B80(S) |  |
|               | KV-21DS55 | WIG-21DS55  |  |

| 商品カテゴリ名 | 商品コード    | 商品名         | バーコード   |
|---------|----------|-------------|---|
| テレビ(液晶) | 14AR8    | 14 型 AR8    |  |
|         | 14C-GM3  | 14 型 C-GM3  |  |
|         | 20C-GM3  | 20 型-GM3    |  |
|         | 21SC8    | 21 型 SC8    |  |
|         | AV-21N1  | AV-21N1     |  |
|         | C-20D20S | 20 型 D20(S) |  |

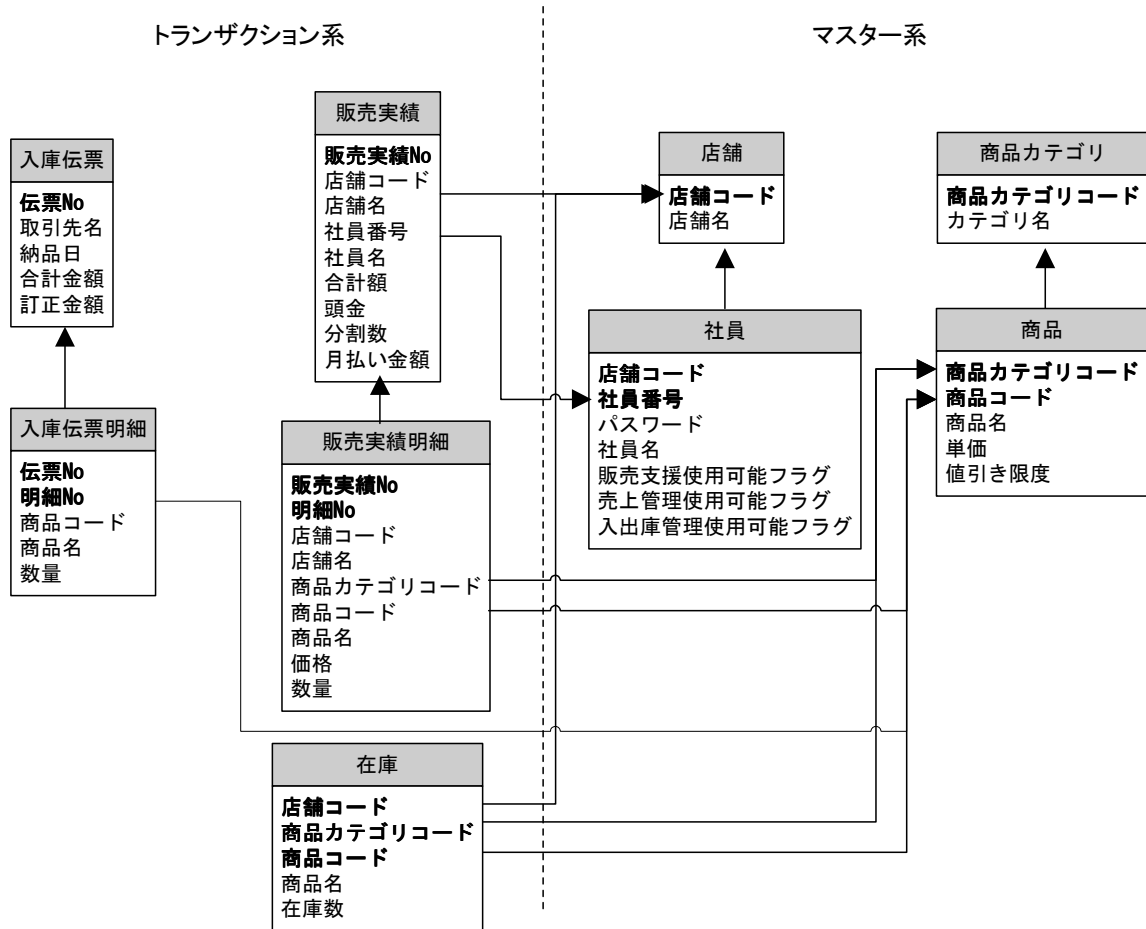
| 商品カテゴリ名   | 商品コード      | 商品名          | バーコード  |
|-----------|------------|--------------|--|
| テレビ(プラズマ) | 42P4000    | FOOT42P4000  |  |
|           | KDE-P32HV2 | WIG-P32HV2   |  |
|           | KDE-P42HX2 | WIG-P42HX2   |  |
|           | PDP-37HD3  | VIZPDP-37HD3 |  |
|           | PDP-37V2   | VIZPDP-37V2  |  |
|           | PDP-42HD3  | VIZ-42HD3    |  |

## 4.1.3 入庫伝票マスタ

| 伝票 No      | 取引先  | バーコード  |
|------------|------|--|
| 1000000001 | 太和物流 |  |
| 1000000002 | 目立搬送 |  |
| 1000000003 | 太和物流 |  |

## 4.2 テーブル定義

### 4.2.1 ER 図



### 4.2.2 テーブル一覧

| No. | 論理テーブル名 | 物理テーブル名            |
|-----|---------|--------------------|
| 1   | 店舗      | SHOP               |
| 2   | 社員      | STAFF              |
| 3   | 商品カテゴリ  | PRODUCT_CATEGORY   |
| 4   | 商品      | PRODUCT            |
| 5   | 販売実績    | SALES              |
| 6   | 販売実績明細  | SALES_DETAIL       |
| 7   | 在庫伝票    | WAREHOUSING        |
| 8   | 在庫伝票明細  | WAREHOUSING_DETAIL |
| 9   | 在庫      | STOCK              |

# 開発ガイド

## テーブル定義書

### SHOP

|       | カラム       |           | 属性    |     |    | NOT NULL<br>制約 | 説明 |
|-------|-----------|-----------|-------|-----|----|----------------|----|
|       | 論理カラム名    | 物理カラム名    | データ型  | サイズ | キー |                |    |
| 店舗コード | SHOP_CODE | SHOP_CODE | テキスト型 | 20  | PK | ○              |    |
| 店舗名   | SHOP_NM   | SHOP_NM   | テキスト型 | 50  |    | ○              |    |

### STAFF

|              | カラム           |               | 属性      |     |    | NOT NULL<br>制約 | 説明                |
|--------------|---------------|---------------|---------|-----|----|----------------|-------------------|
|              | 論理カラム名        | 物理カラム名        | データ型    | サイズ | キー |                |                   |
| 社員番号         | STAFF_NO      | STAFF_NO      | テキスト型   | 20  | PK | ○              |                   |
| 店舗コード        | SHOP_CODE     | SHOP_CODE     | テキスト型   | 20  | PK | ○              |                   |
| パスワード        | PASSWD        | PASSWD        | テキスト型   | 20  |    | ○              |                   |
| 社員名          | STAFF_NM      | STAFF_NM      | テキスト型   | 50  |    | ○              |                   |
| 販売支援使用可能フラグ  | AUTH_SALES    | AUTH_SALES    | Yes/No型 |     |    | ○              | FALSE:不可能、TRUE:可能 |
| 売上管理使用可能フラグ  | AUTH_CHECK    | AUTH_CHECK    | Yes/No型 |     |    | ○              | FALSE:不可能、TRUE:可能 |
| 入在庫管理使用可能フラグ | AUTH_STORE_IO | AUTH_STORE_IO | Yes/No型 |     |    | ○              | FALSE:不可能、TRUE:可能 |

### PRODUCT CATEGORY

|           | カラム                   |                       | 属性    |     |    | NOT NULL<br>制約 | 説明 |
|-----------|-----------------------|-----------------------|-------|-----|----|----------------|----|
|           | 論理カラム名                | 物理カラム名                | データ型  | サイズ | キー |                |    |
| 商品カテゴリコード | PRODUCT_CATEGORY_CODE | PRODUCT_CATEGORY_CODE | テキスト型 | 20  | PK | ○              |    |
| 商品カテゴリ名   | PRODUCT_CATEGORY_NM   | PRODUCT_CATEGORY_NM   | テキスト型 | 50  |    | ○              |    |

### PRODUCT

|           | カラム                   |                       | 属性    |     |    | NOT NULL<br>制約 | 説明 |
|-----------|-----------------------|-----------------------|-------|-----|----|----------------|----|
|           | 論理カラム名                | 物理カラム名                | データ型  | サイズ | キー |                |    |
| 商品カテゴリコード | PRODUCT_CATEGORY_CODE | PRODUCT_CATEGORY_CODE | テキスト型 | 20  | PK | ○              |    |
| 商品コード     | PRODUCT_CODE          | PRODUCT_CODE          | テキスト型 | 20  | PK | ○              |    |
| 商品名       | PRODUCT_NM            | PRODUCT_NM            | テキスト型 | 50  |    | ○              |    |
| 単価        | UNIT                  | UNIT                  | 長整数   |     |    | ○              |    |
| 値引き限度     | UNIT_LIMIT            | UNIT_LIMIT            | 長整数   |     |    | ○              |    |

### SALES

|        | カラム              |                  | 属性    |     |    | NOT NULL<br>制約 | 説明 |
|--------|------------------|------------------|-------|-----|----|----------------|----|
|        | 論理カラム名           | 物理カラム名           | データ型  | サイズ | キー |                |    |
| 販売実績No | SALES_NO         | SALES_NO         | 長整数   |     | PK | ○              |    |
| 店舗コード  | SHOP_CODE        | SHOP_CODE        | テキスト型 | 20  | FK | ○              |    |
| 店舗名    | SHOP_NM          | SHOP_NM          | テキスト型 | 50  |    | ○              |    |
| 社員番号   | STAFF_NO         | STAFF_NO         | テキスト型 | 20  | FK | ○              |    |
| 社員名    | STAFF_NM         | STAFF_NM         | テキスト型 | 50  |    | ○              |    |
| 合計額    | SUM              | SUM              | 長整数   |     |    | ○              |    |
| 頭金     | DEPOSIT          | DEPOSIT          | 長整数   |     |    | ○              |    |
| 分割数    | INSTALLMENTS_NUM | INSTALLMENTS_NUM | 長整数   |     |    | ○              |    |
| 月払い金額  | INSTALLMENTS     | INSTALLMENTS     | 長整数   |     |    | ○              |    |

### SALES DETAIL

|           | カラム                   |                       | 属性    |     |        | NOT NULL<br>制約 | 説明 |
|-----------|-----------------------|-----------------------|-------|-----|--------|----------------|----|
|           | 論理カラム名                | 物理カラム名                | データ型  | サイズ | キー     |                |    |
| 販売実績No    | SALES_NO              | SALES_NO              | 長整数   |     | PK, FK | ○              |    |
| 明細No      | DETAIL_NO             | DETAIL_NO             | 長整数   |     | PK     | ○              |    |
| 店舗コード     | SHOP_CODE             | SHOP_CODE             | テキスト型 | 20  | FK     | ○              |    |
| 店舗名       | SHOP_NM               | SHOP_NM               | テキスト型 | 50  |        | ○              |    |
| 商品カテゴリコード | PRODUCT_CATEGORY_CODE | PRODUCT_CATEGORY_CODE | テキスト型 | 20  | FK     | ○              |    |
| 商品コード     | PRODUCT_CODE          | PRODUCT_CODE          | テキスト型 | 20  | FK     | ○              |    |
| 商品名       | PRODUCT_NM            | PRODUCT_NM            | テキスト型 | 50  |        | ○              |    |
| 価格        | UNIT                  | UNIT                  | 長整数   |     |        | ○              |    |
| 数量        | COUNT                 | COUNT                 | 長整数   |     |        | ○              |    |

### WAREHOUSING

|      | カラム            |                | 属性    |     |    | NOT NULL<br>制約 | 説明           |
|------|----------------|----------------|-------|-----|----|----------------|--------------|
|      | 論理カラム名         | 物理カラム名         | データ型  | サイズ | キー |                |              |
| 伝票No | WAREHOUSING_NO | WAREHOUSING_NO | テキスト型 | 50  | PK | ○              |              |
| 取引先名 | TRADER_NM      | TRADER_NM      | テキスト型 | 50  |    | ○              |              |
| 納品日  | DELIVERY_DATE  | DELIVERY_DATE  | テキスト型 | 10  |    | ○              | YYYY/MM/DD形式 |
| 合計額  | SUM            | SUM            | 長整数   |     |    | ○              |              |
| 訂正金額 | REVISION_SUM   | REVISION_SUM   | 長整数   |     |    | ○              |              |

### WAREHOUSING DETAIL

|       | カラム            |                | 属性    |     |    | NOT NULL<br>制約 | 説明 |
|-------|----------------|----------------|-------|-----|----|----------------|----|
|       | 論理カラム名         | 物理カラム名         | データ型  | サイズ | キー |                |    |
| 伝票No  | WAREHOUSING_NO | WAREHOUSING_NO | テキスト型 | 50  | PK | ○              |    |
| 明細No  | DETAIL_NO      | DETAIL_NO      | 長整数   |     | PK | ○              |    |
| 商品コード | PRODUCT_CODE   | PRODUCT_CODE   | テキスト型 | 20  | FK | ○              |    |
| 商品名   | PRODUCT_NM     | PRODUCT_NM     | テキスト型 | 50  |    | ○              |    |
| 数量    | PRODUCT_COUNT  | PRODUCT_COUNT  | 長整数   |     |    | ○              |    |

### STOCK

|           | カラム                   |                       | 属性    |     |        | NOT NULL<br>制約 | 説明 |
|-----------|-----------------------|-----------------------|-------|-----|--------|----------------|----|
|           | 論理カラム名                | 物理カラム名                | データ型  | サイズ | キー     |                |    |
| 店舗コード     | SHOP_CODE             | SHOP_CODE             | テキスト型 | 20  | PK, FK | ○              |    |
| 商品カテゴリコード | PRODUCT_CATEGORY_CODE | PRODUCT_CATEGORY_CODE | テキスト型 | 20  | PK, FK | ○              |    |
| 商品コード     | PRODUCT_CODE          | PRODUCT_CODE          | テキスト型 | 20  | PK, FK | ○              |    |
| 商品名       | PRODUCT_NM            | PRODUCT_NM            | テキスト型 | 50  |        | ○              |    |
| 在庫数       | STOCK_COUNT           | STOCK_COUNT           | 長整数   |     |        | ○              |    |

---

## **Biz/Browser Mobile** 開発ガイド

---

2004年 11 月 12 日 V1.01 発行  
2006年 10 月 6 日 V1.1 改訂

発行: アクシスソフト株式会社  
〒170-0013 東京都豊島区東池袋 3-23-5 ダヴィンチ東池袋ビル

問い合わせ先: サポート係 [biz-qa@axissoft.co.jp](mailto:biz-qa@axissoft.co.jp)