

Magic xpa 3.1 対応 モバイル用アプリ  
「スマートセールス for 顧客大臣」

開発者ガイド

Ver. : 1.00 作成 : 平成 29 年 8 月 日

株式会社ビシオス

## 目 次

1. 本システムの提供ファイル .....	2
2. モデルリポジトリ(初心者向け) .....	3
3. データリポジトリ(初心者向け) .....	6
4. プログラムリポジトリ(初心者向け) .....	7
5. リッチクライアント開発の特徴 (クラサバ開発経験者で、RIA 開発未経験者向け)	
5-1 フォームはすべて Magic で作成 .....	8
5-2 色・フォントファイルはそのまま、キー割付も有効 .....	8
5-3 クライアント側に実行版のインストール作業は不要 .....	9
5-4 印刷は PDF や ReportsMagic .....	9
5-5 バッチタスクはそのまま .....	9
5-6 RM 互換(レコードメイン互換)タスクは不可 .....	9
5-7 ファントムタスクは不可 .....	9
5-8 リッチクライアントに関して、コール出来ない組み合わせがある .....	10
5-9 遅延トランザクションモードを使用 .....	10
5-10 使用出来ない関数がある .....	10
5-11 使用出来ない機能がある .....	11
5-12 オンラインタスクのようなレコードロックの概念はない .....	11
5-13 パフォーマンスに関する注意点 .....	11
6. モバイル開発の特徴 (RIA 開発経験者で、モバイル開発未経験者向け)	
6-1 基本的な考え方 .....	13
6-2 画面サイズの違い .....	13
6-3 ユーザインタフェース .....	13
6-4 使用出来ないコントロール .....	13
6-5 フォントについて .....	14
6-6 色について .....	14
6-7 デバイス特性の取得関数 .....	14
6-8 デバイス機能の利用 .....	15
7. テクニック集	
7-1 グローバル変数／グローバル関数 .....	16
7-2 地図表示 .....	17
8. 顧客大臣連携について	
7-1 グローバル変数／グローバル関数 .....	19
付録(モデル一覧・テーブル一覧・プログラム一覧).....	20

## 第1章 本システムの提供ファイル

本システムで提供されているファイルを、フォルダ毎に紹介します。

(本ドキュメントでは SMARTSALES のモジュールを C ドライブのルートディレクトリに解凍した場合を想定し記述してありますので、C:¥ の部分は解凍先ディレクトリに置き換えて作業を行って下さい)

フォルダ名	ファイル名	内 容
C:SMARTSALES	SMARTSALES.edp	本システムのプロジェクトファイル
ENV ENV¥Android ENV¥iOS	clr_mobile.jp fnt_mobile.jp fnt_mobile.jp fnt_mobile.jp	色定義ファイル フォント定義ファイル Android 用フォント定義ファイル iOS 用フォント定義ファイル
images images¥icons images¥Android images¥iOS	*.jpg *.png	ロゴおよび壁紙ファイル システム内で使用するイメージファイル Android 用アイコン・スプラッシュイメージファイル iOS 用アイコン・スプラッシュイメージファイル
root¥work	(各種ファイル)	一時ファイルを格納するフォルダ
Source	*.xml	プログラム等のソースファイル
Template	MapFan_tplt.html	地図表示用 html ファイル

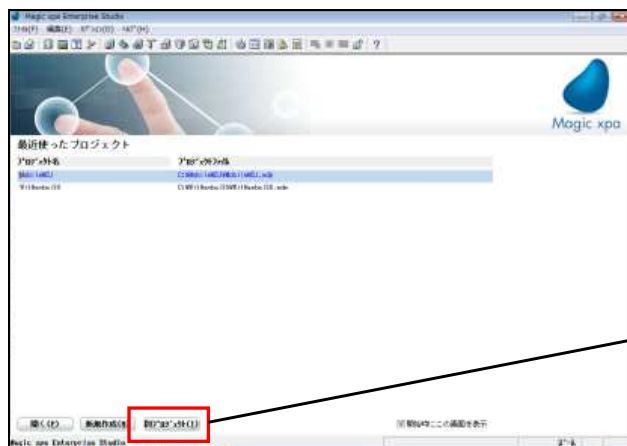
## 第2章 モデルリポジトリ

初心者向け

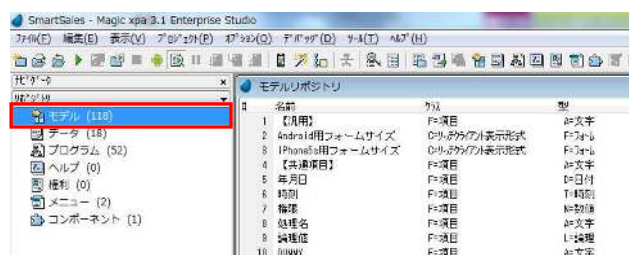
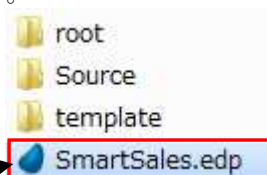
モデルリポジトリでは、システム内で使用する「項目の型や桁数」の設定や、「フォームの大きさや色」の設定を行います。

ここで設定した値は、システム全体を通じて一括して管理することができ、もし将来、設定した値が変更になったとしても、対象となるモデルの値を変更するだけで、対象となるモデルを使用しているテーブルやプログラムの値も、自動的に変更することが出来ます。

まずは、モデルリポジトリを開きます。



Magic xpa 開発版を起動し、「別プロジェクト」をクリックします。  
そして、「C:\¥SMARTSALES¥SMARTSALES.edp」を選択します。



開発画面が表示されますので、モデルリポジトリを開きます。

※画面左端にナビゲータが表示されていない場合は、  
表示するようにして下さい。  
[表示]→[ナビゲータ] を選択します。

## (1) 「項目」のモデル例

## ■ 文字型

モデルリポジトリ			
#	名前	クラス	型
1	【汎用】	F=項目	A=文字
2	Android用フォームサイズ	C=リッチテキスト表示形式	F=フォーム
3	iPhone5s用フォームサイズ	C=リッチテキスト表示形式	F=フォーム
4	【共通項目】	F=項目	A=文字
5	年月日	F=項目	D=日付
6	時刻	F=項目	T=時刻
7	権限	F=項目	N=数値
8	処理名	F=項目	A=文字
9	論理値	F=項目	L=論理
10	DUMMY	F=項目	A=文字
11	共通区分	F=項目	N=数値
12	メッセージ	F=項目	A=文字
13	顧客CD	F=項目	N=数値
14	郵便番号	F=項目	A=文字

項目特性 A=文字

区分(C) 全体(A)

モデル [デフォルト]

詳細

書式 X30

型 A=文字

範囲 データの一部 Yes

表示名

入力

選択モード 0

起動モード B=前置

表示

ヘルプ画面 0

ツールチップ 0

モデル 8 番「処理名」は、

「クラス:項目」

「型:文字」

となっています。

また、特性画面内では、

「書式:X30」

となっています。

これは、「処理名」というモデルが、「文字項目で、30 桁(30 バイト)」であることを意味しています。

したがって、システム内でこのモデルを使っている箇所はすべて「文字項目の 30 桁」として一括管理されます。

## ■ 数値型

モデルリポジトリ			
#	名前	クラス	型
1	【汎用】	F=項目	A=文字
2	Android用フォームサイズ	C=リッチテキスト表示形式	F=フォーム
3	iPhone5s用フォームサイズ	C=リッチテキスト表示形式	F=フォーム
4	【共通項目】	F=項目	A=文字
5	年月日	F=項目	D=日付
6	時刻	F=項目	T=時刻
7	権限	F=項目	N=数値
8	処理名	F=項目	A=文字
9	論理値	F=項目	L=論理
10	DUMMY	F=項目	A=文字

項目特性 N=数値

区分(C) 全体(A)

モデル [デフォルト]

詳細

書式 #2

型 N=数値

範囲 データの一部 Yes

表示名

入力

選択モード 0

起動モード B=前置

表示

ヘルプ画面 0

ツールチップ 0

モデル  
フォーム/コントロール/項目の各モデルに対するすべての特性を再継承したり継承を解除することができます。

モデル 7 番「権限」は、

「クラス:項目」

「型:数値」

となっています。

また、特性画面内では、

「書式:#2」

となっています。

これは、「権限」というモデルが、「数値項目で、マイナスありの 2 桁で表示する」ということを意味しています。

システム内でこのモデルを使っている箇所はすべて、「数値項目のマイナスありの 2 桁」として一括管理されます。

## (2) 「フォーム」のモデル例

モデルリポジトリ			
#	名前	クラス	型
1	【汎用】	F=項目	A=文字
2	Android用フォームサイズ	C=リッチクライアント表示形式	F=フォーム
3	iPhone5s用フォームサイズ	C=リッチクライアント表示形式	F=フォーム
4	【共通項目】	F=項目	A=文字
5	年月日	F=項目	D=日付
6	時刻	F=項目	T=時刻
7	権限	F=項目	N=数値
8	処理名	F=項目	A=文字
9	論理値	F=項目	L=論理
10	DUMMY	F=項目	A=文字

モデル 2 番「Android 用フォームサイズ」は、  
「クラス:リッチクライアント表示形式」  
「型:フォーム」  
となっています。

フォーム特性 C=リッチクライアント表示形式

区分(C)	全体(A)
フォント	1
色	1
グラフィックスタイル	N=なし
グラフィック色	1
画面	0
境界スタイル	H=太線
ロケーション	
開始時の位置	C=中央
X	0.000
Y	0.000
幅	6.220
高さ	11.050
最小幅	0.000
最小高さ	0.000

カット間隔(X)  
フォームのカット間の垂直間隔を指定します。

リソース 特性

また、特性画面内では、

「フォント」  
「色」  
「幅」  
「高さ」

などが設定されています。

システム内でこのモデルを使っているフォームはすべて、色と大きさが統一されます。

## 第3章 データリポジトリ

初心者向け

データリポジトリでは、システム内で使用する「テーブル(ファイル)」を定義します。  
「テーブル名(ファイル名)」、「カラム(項目)」、「インデックス(キー)」などを設定します。

## ■ データ(ファイル)定義の例

#	名前	テーブル名	データベース
1	マスタ		Default Database
2	顧客データ	MST_KOKYAKU	Local(Mst)
3	貸借人データ	MST_TAISYAKUNIN	Local(Mst)
4	担当者マスタ	MST_TANTOSHA	Local(Mst)
5	商品・保守マスタ	MST_HOSHU	Local(Mst)
6	請求先マスタ	MST_SEIKYUSAKI	Local(Mst)
7	消耗品サブライマスタ	MST_SHOMOHIN	Local(Mst)
8	メーカーマスタ	MST_MAKER	Local(Mst)
9	住居区分マスタ	MST_JUKYOKUBUN	Local(Mst)
10	顧客ランクマスタ	MST_KOKYAKU_RANK	Local(Mst)
11	項目マスタ	MST_KOMOKU	Local(Mst)
12	選択項目データ	MST_SENTAKUDATA	Local(Mst)
13	■■■■■データ■■■■■		Default Database
14	機器一覧データ	DAT_KIKIJOUHOU	Local(Tran)
15	メンテナンス履歴データ	DAT_MAINTENANCE	Local(Tran)
16	営業履歴	DAT_EIGYO_RIREKI	Local(Tran)
17	ログイン状況	DAT_LOGIN	SQLite(Tran)
18	アクセスログ	DAT_ACCESS	SQLite(Tran)

データ 5 番「商品・保守マスタ」は、

「データソース名:MST\_HOSHU」

「データベース:Local(Mst)」 (Local)

となっています。

## &lt;カラム&gt;

#	名前	モデル	型	書式
1	商品CD	44 商品CD	N=数値	#4
2	名称	45 名称	A=文字	X40
3	メンテナンスサイクル(月)	20 メンテナンスサイクル(月)	N=数値	#2
4	メーカーCD	48 メーカーCD	N=数値	#4
5	商品区分	46 商品区分	N=数値	#2

5 個のカラムが設定されており、それぞれモデルを使用しています。

「型」と「書式」は、モデルの値が継承されます。

## &lt;インデックス&gt;

#	名前	タイプ	プライマリキー
1	KEY01/U	U=重複不可	<input checked="" type="checkbox"/>

#	カラム	名前	サイズ	順序
1	1	商品CD	2	A=昇順

上の部分がインデックスの名前です。

インデックスが 1 つ定義されており、「重複不可」に設定されています。

下の部分には、インデックスの実際の中身のカラム(セグメント)が定義されています。

## 第4章 プログラムリポジトリ

初心者向け

プログラムリポジトリでは、プログラムを記述します。  
「データビュー／ロジック／フォーム」の3つを定義することによってプログラムが成立します。  
プログラムを「書く」というよりも「設定」というイメージになります。

### ■プログラム定義の例

#	名前	場所
11	//メイン	
12	スタート	
13	ログイン	
14	トップメニュー	
15	訪問予定検索	
16	担当者選択	
17	顧客一覧	
18	顧客情報	

プログラム 13 番「ログイン」を見てみます。  
F5(ズーム)キーを押すと、プログラムの中身を見ることができます。



画面上部に3つのタブがあります。

#### <データビュー>

データビュー		ロジック	フォーム
1	M=メインソース	0	メインソース未定義
2	V=変数	1	V.ログインID
3	V=変数	2	V.パスワード
4			
5	V=変数	3	V.区分

プログラムで使用するビュー項目を定義します。

「メインソース(メインファイル)」  
「メインソースのカラム」  
「変数項目」  
「パラメータ項目」

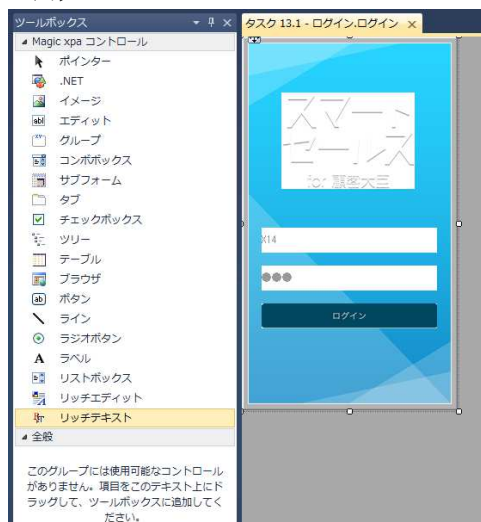
#### <ロジック>

S	1	T=タスク	P=前	
	2	項目更新	V=項目	P G.プログラムNo
M	3	R=レポート	S=後	
C	4	項目更新	V=項目	0 G.デバイス区分
C	5	項目更新	V=項目	0 G.デバイス区分
	6			//ログイン画面表示
M	7	コール	S=タスク	1 ログイン

ロジックを記述します。

「いつ、どこで、何をするか」

#### <フォーム>



フォーム(画面)を作成します。



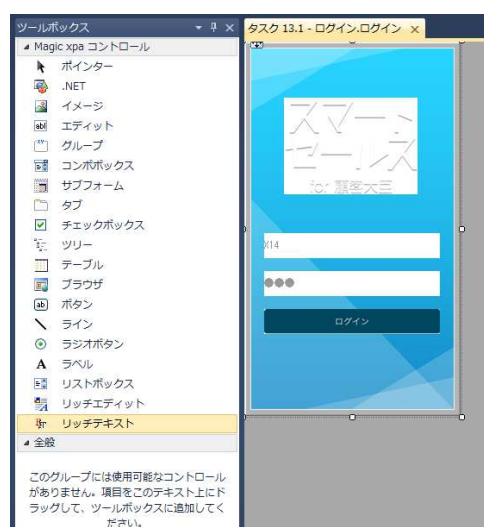
## 第 5 章 リッチクライアント開発の特徴 クラサバ開発経験者で、RIA 開発未経験者向け

この章では、クラサバ開発は経験したことがあるが、RIA 開発は初めてという方を対象にして、「パソコン向け RIA 開発の特徴」を挙げてみました。なお、「モバイル向け RIA 開発の特徴」については次章で紹介しています。

※ここでは簡単な説明だけにとどめておりますので、詳細をお知りになりたい方は、Magic 付属のマニュアル及び MSJ 発行の資料等を参照して下さい。

### 5-1 フォームはすべて Magic で作成

フォームはすべて Magic で作成しますので、HTML の知識や FrontPage などのオーサリングツールの経験も必要ありません。オンラインタスクでフォームを作るのと同じ要領です。

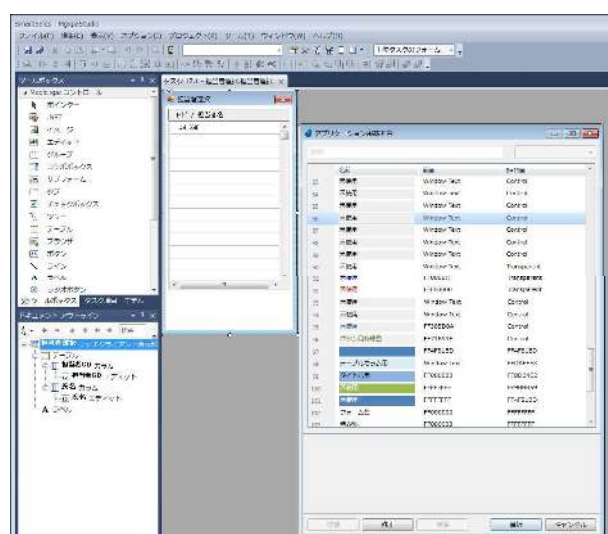


※オンラインタスクの時と同じ

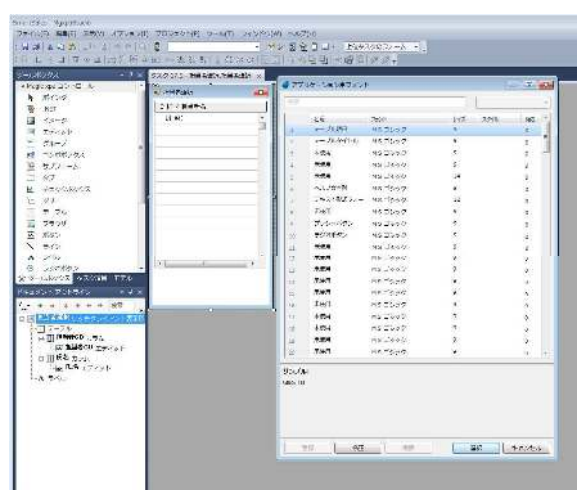
### 5-2 色・フォントファイルはそのまま、キー割付も有効

色ファイルやフォントファイルは今までのオンラインタスクの時と同じです。また、キーボード割付も今まで通り有効です。

<色テーブル>



<フォントテーブル>



<キーボード操作の一例>

登録:Ctrl+E、修正:Ctrl+M、照会 Ctrl+Q、終了:Esc、行削除:F3、行作成:F4

キャンセル:Ctrl+F2、次項目:Tab、前項目:Shift+Tab、テーブル先頭:Ctrl+Home、テーブル末尾:Ctrl+End

前画面:PgUp、次画面:PgDn、コピー:Ctrl+C、貼り付け:Ctrl+V など

### 5-3 クライアント側に実行版のインストール作業は不要

クライアント側に、Magic 実行版のインストール作業は必要ありません。  
初回アクセス時に、リッチクライアント用の Magic エンジンが自動でインストールされます。  
(.Net Framework 2.0 SP2 以上の環境があればいいです)

### 5-4 印刷は PDF や ReportsMagic

クライアント側から印刷するためには、PDF や ReportsMagic を利用するのが便利です。  
(本システムには、印刷処理はありません)

<PDF>



<ReportsMagic>



### 5-5 バッチタスクはそのまま

今まで使っていたオンラインタスクをリッチクライアントタスクに変換するには、タスク特性内のタスクタイプを、「O=オンライン」から「C=リッチクライアント」に変更するわけですが、バッチタスクにおいては、そのまま利用することができます。

### 5-6 RM 互換(レコードメイン互換)タスクは不可

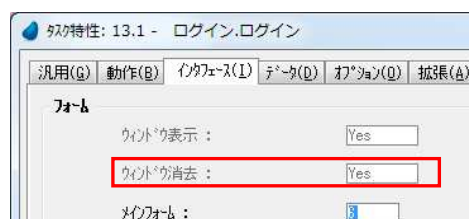
RM 互換が存在するオンラインタスクは、リッチクライアントタスクに変換できません。プログラムチェック時に次のようなメッセージが出てきます。

[RM 互換]ロジックユニットは、オンラインタスクでのみサポートします

したがって、イベントドリブン型の開発手法に変更する必要があります。

### 5-7 ファントムタスクは不可

受注入力プログラムのように、親タスクでヘッダデータを入力し、子タスクで明細データを入力するようなプログラムを作る際に、子タスクの画面を最初に表示したいため、親タスクのレコード前処理で子タスクをコールし、子タスクの終了条件欄に「Level(1)='RP'」、ウィンドウ消去欄に「No」を設定する方法がありました。  
このようなタスクのことを「ファントムタスク」と呼んでいますが、リッチクライアントタスクにおいては、ファントムタスクのような仕組みでは動作しません。なぜなら、リッチクライアントタスクにおいては、ウィンドウ消去欄が「Yes」固定になっているためです。



ウィンドウ消去欄が「Yes」固定になっており、変更できません。

したがって、サブフォームを使用する手法に変更する必要があります。

## 5-8 リッチクライアントに関して、コール出来ない組み合わせがある

リッチクライアントタスクから別タスクをコールする場合や、別タスクからリッチクライアントタスクをコールする場合に、コール出来ない組み合わせがあります。以下の表をご覧ください。

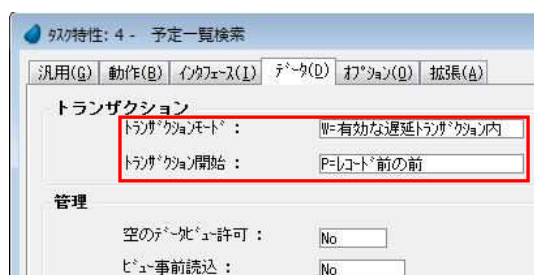
コール元	コール先	可(○)・不可(×)
オンラインタスク	リッチタスク	×
バッチタスク	リッチタスク	×
リッチタスク	リッチタスク	○
リッチタスク	オンラインタスク	×
リッチタスク	バッチタスク	○

今までバッチタスクからオンラインタスクをコールしていた場合には、バッチタスクからリッチクライアントタスクはコール出来なくなります。

ただし、リッチクライアントタスクのタスク特性内で、「インタラクティブ」チェックボックスのチェックを外すとユーザの操作を受け付けなくなり、バッチタスクのような動作になります。この機能を利用することによりバッチタスクからリッチクライアントタスクを呼び出すような処理が実現できます。(詳細は Magic のヘルプをご覧ください)

## 5-9 遅延トランザクションモードを使用

リッチクライアントタスクではトランザクションモードが、物理モードではなく、遅延モードになります。



オンラインタスクでは、トランザクションモードは主に「P=物理」を使っていましたが、リッチクライアントタスクでは「P=物理」がなくなって、「W=有効な遅延トランザクション内」になります。

「N=新規の遅延トランザクション」というものもありますが、ここでは説明を省略します。

物理モードでは、レコードに対して何らかの修正等を行なった場合、レコード後処理を通過した後に、ファイルに反映されますが、遅延モードでは、修正等の情報はコンテキスト(サーバのメモリ上)に蓄積され、トランザクションが終了する時点でファイルに反映されるという動きになります。

したがって「トランザクション開始」の設定にも注意が必要となり、この設定によってトランザクションの範囲が決定されます。

※メニューだけを表示するようなタスクの場合は、トランザクションモードを「O=なし」にした方が望ましいです。これにより、メニューから起動されたタスクはすべて独立したトランザクション処理を行なうようになります。

※遅延トランザクションの詳細につきましては、Magic xpa のホワイトペーパー「データ管理 Magic xpa」を参照して下さい。

## 5-10 使用出来ない関数がある

リッチクライアントタスクでは使用できない関数がいくつかあります。

これについては、Magic ReadMe 内の「参考技術情報/リッチクライアントとオンラインの違い」に載っています。

<使用できない関数>

Blob2Req、File2Req、KbGet、KbPut、Text、TransMode、UDF 関数、Variant 関数、WinHelp、XML 関数 など

使用出来ない関数を使っている場合には、別のコマンドに置き換えるか、ロジックを変更する必要があります。また、端末番号(Term 関数)という概念も意味がなくなります。

## 5-11 使用出来ない機能がある

リッチクライアントタスクは、オンラインタスクの機能を 100% サポートしているわけではありません。  
 (ほぼ 100% に近づきつつありますが)、どのような機能がサポートされていないかは、Magic ReadMe 内の「参考  
 技術情報／リッチクライアントとオンラインの違い」に載っています。  
 使用出来ない機能を使っている場合には、機能を制限するか、ロジックを変更する必要があります。

## 5-12 オンラインタスクのようなレコードロックの概念はない

データビューに定義したメインソース特性内で「アクセス」と「共有」を「W=書出」としても、オンラインタスクのような  
 レコードロックはかかりません。オンラインタスクの場合は、タスク特性内に「ロック方式」の設定欄がありましたが、  
 リッチクライアントタスクには、その設定自体がありません。

&lt;オンラインタスク&gt;

&lt;リッチクライアントタスク&gt;

Magic xpa のリッチクライアントタスクのように、遅延トランザクションを有効にしたタスクでは、レコードをロックする  
 方式ではなく、更新しようとするレコードが既に更新されていないかを更新時に自動的に確認する、いわゆる“楽  
 観的ロック”による排他制御を行っています。  
 遅延トランザクションについての詳細は、ホワイトペーパー「データ管理 Magic xpa」をご覧ください。

## 5-13 パフォーマンスに関する注意点

## 1) ロジックについて

リッチクライアントタスクにおいては、サーバ側で処理されるコマンド(関数)と、クライアント側で処理されるコマ  
 ンド(関数)を自動的に判断しています。そして、これらの処理が混在するようなロジックの場合、プログラムチ  
 ェック時にエラー(ワーニング)が出てきます。

サーバ側の関数なのかクライアント側の関数なのかは、ヘルプ内に載っています。

[タスク前処理]	....	サーバ側で実行(クライアント側の処理は実行されない)
[タスク後処理]	....	処理内容に応じてサーバ側あるいはクライアント側で実行
エラーコマンド	....	クライアント側で実行
リンクコマンド	....	サーバ側で実行
バッチタスク	....	サーバ側で実行 となっています。(これは一例です)

サーバ側での処理(関数)とクライアント側での処理(関数)が混在するような場合は、パフォーマンスに影響  
 が出てきますので、できる限り、サーバ側の処理(関数)とクライアント側の処理(関数)をグループ化し、混在  
 する回数が少なくなるようにすることがポイントです。

どちら側の処理なのかは、データビューやロジック部において、行の先頭の記号でも判断できます。

S・・・サーバ側処理、C・・・クライアント側処理、M・・・サーバ側／クライアント側 処理混在

	1	W=ログイン	0	ログイン未定義	
S	2	V=実行	1	V.ユーザID	
S	3	V=実行	2	V.パスワード	
	4	V=実行	3	V.IDチェック	
	5	V=実行	4	V.担当者CD	
	6	V=実行	5	V.氏名	
S	7	L=照会リク	2	担当者マスタ	
	8	C=カム	1	担当者CD	
	9	C=カム	2	氏名	
	10	C=カム	3	フリガナ	
	11	C=カム	4	課CD	

	12	項目更新	
S	13		
	14	コル	
	15	項目更新	
M	16	日 E=バタ	
M	17		
S	18	日 T=奴	
S	19	アクション	

→ サーバ側処理は赤色背景

## 2) フォームレイアウトについて

オンラインタスクでは、1つの画面に多くのコントロールを配置しても、パフォーマンスの影響は軽微でした。しかし、リッチクライアントタスクにおいては、1つの画面に多くのコントロールを配置すると、パフォーマンスの影響を受けます。

コントロール前・検証・後などのロジックや項目変更ロジックなど、コントロール間のカーソル移動により実行されるロジックがあります。これらのロジックに処理を実装していると、カーソルが移動すれば、そのロジックが呼び出されることになり、サーバへの通信やロジックの処理が実行されることに繋がるわけです。

サーバへの通信やロジックの実行が頻繁に発生すれば、パフォーマンスが低下することは、先の説明の通りです。

1つの画面には多くのコントロールを配置するのではなく、出来るだけシンプルな画面にして、別の画面を呼び出すなど、画面レイアウトを工夫すると良いでしょう。

## 第 6 章 モバイル開発の特徴

### RIA 開発経験者で、モバイル開発未経験者向け

この章では、RIA 開発は経験したことがあるが、モバイル開発は初めてという方を対象にして、「モバイル向け RIA 開発の特徴」を挙げてみました。

※ Magic 付属のマニュアル及び MSJ 発行の資料等も参考にして下さい。「モバイル RIA 開発者ガイド」が参考になるかと思います。また、Android の場合には、「Android での Magic RIA (はじめの一步)」は必読です。

#### 6-1 基本的な考え方

モバイルでどのようなことをやりたいのかを、はっきり決めておく必要があります。

当然ながらモバイルは画面が小さくなりますので、あまり複雑な画面にしまうと見づらく操作性も悪くなるかと思えます。

モバイル用 RIA アプリを開発すること自体は、パソコン用 RIA アプリとはほぼ同様な考え方で開発できますが、そもそも画面の大きさやユーザインタフェースが違いますし、サポートされていない機能や OS 固有の制限事項もあります。したがって、動作テストする場合は必ず実機で行なう必要があります。

#### 6-2 画面サイズの違い

モバイルデバイス毎に画面サイズが違ってきます。

そのため、プログラム内ではそのデバイスサイズに合わせたフォームを作らなければなりません。もし、画面サイズの違うデバイスが複数存在する場合には、複数のフォームを作っておいて、使用するデバイスごとに IF 文や CASE 文を使ってフォームを切り替えるような仕組みが必要になります。

#### 6-3 ユーザインタフェース

##### ■ウィンドウに関して

新規に開くウィンドウは前のウィンドウの上に重なって表示されるので、「モーダル」の機能になります。

フォーム特性内の値を変更すれば、ポップアップウィンドウの設定は可能です。

現在のウィンドウを閉じて前のウィンドウに戻るには、プログラム内で「終了」イベントを発行するか、またはモバイルデバイスの「Back」ボタンを押します。

また、マウスポインタがないので画面を移動したり、サイズ変更したりすることは出来ません。

##### ■キー操作に関して

キーボード入力が必要な項目にフォーカスすると、画面下部にキーボードが自動的に表示されるので、その時は、画面の下部が見えなくなります。また、パソコンのキーボードとは違い、項目間を移動するためのキーは存在せず、その項目をタッチすることで項目を移動します。

#### 6-4 使用できないコントロール

以下のコントロールは、モバイルでは使用出来ません。

- ・ラジオボタン
- ・リストボックス
- ・タブ
- ・ツリー
- ・リッチテキスト など

また、インクリメンタルサーチ(照会モード時の自動位置付け機能)も、モバイルでは対応されていません。



## 6-5 フォントについて

WindowsOS とは異なる独自のフォントを持っていますので、それぞれのフォントの種類を調査しておく必要があります。

また、Magic のフォント定義画面では、モバイル用フォントを定義(ズーム選択)することができないため、フォントテーブル (fnt\_XXX.jpN) をメモ帳などで直接編集することになります。

## &lt;Android フォントの例&gt;

Android font1,Sans-Serif,8,0,0  
Android font2,Sans-Serif,8,0,0,Bold

## &lt;iOS フォントの例&gt;

iOS font1,Helvetica,8,0,0  
iOS font2,Helvetica-Bold,8,0,0

## &lt;メモ帳で編集した結果&gt;

アプリケーション用フォント: C:\JOB¥SmartSales¥Env¥fnt\_mobile.jpN

#	名前	フォント	スタイル	サイズ	傾き
101	ゴシック(8B)	MS ゴシック	B	9	0
102	ゴシック(10B)	MS ゴシック	B	10	0
103	ゴシック(12B)	MS ゴシック	B	12	0
104	ゴシック(14B)	MS ゴシック	B	14	0

定義したフォントが、モバイルデバイス側で見つからなかった場合は、デバイスが持っているデフォルトのフォントが使用されます。

## ■フォントファイルの自動切り替え機能

フォントファイルが「ENV¥fnt\_mobile.jpN」だった場合、「ENV」フォルダ内に、「Andriod」フォルダと「iOS」フォルダを作り、それぞれに独自のフォントファイルを置くと、実行時に、使用する OS に応じてそれぞれのフォントファイルが使用されるようになります。(これらのファイルパスを明示的に指定する必要はありません)

「ENV」フォルダ内



それぞれのフォルダに独自の「fnt\_mobile.jpN」ファイルを置くと  
使用する OS に応じて自動的に切り替わります。

## 6-6 色について

使用するデバイスによって微妙に見た目の色が変わりますので、必ず実機で確認しておいて下さい。

## ■色ファイルの自動切り替え機能

色ファイルが「ENV¥clr\_mobile.jpN」だった場合、「ENV」フォルダ内に、「Android」フォルダと「iOS」フォルダを作り、それぞれに独自の色ファイルを置くと、実行時に、使用する OS に応じてそれぞれの色ファイルが使用されるようになります。(これらのファイルパスを明示的に指定する必要はありません)

「ENV」フォルダ内



それぞれのフォルダに独自の「clr\_mobile.jpN」ファイルを置くと  
使用する OS に応じて自動的に切り替わります。

## 6-7 デバイス特性の取得関数

デバイス側の特性を取得するための関数が用意されています。(以下、主なものです)

関数	説明
ClientOSEnvGet('device_os')	デバイスの OS (Android、iOS) を返します。
ClientOSEnvGet('device_os-version')	デバイスの OS バージョンを返します。
ClientOSEnvGet('device_model')	デバイスのモデル番号または名前を返します。
ClientOSEnvGet('device_magic-version')	RIA クライアントのバージョンを返します。
ClientOSEnvGet('device_location')	縦方向の物理画面の高さをインチで返します。

※詳細については、関数のヘルプをご覧ください。

## 6-8 デバイス機能の利用

デバイスが持っている機能を利用することができます。(以下、主なものです)

機能	方法(文法)
位置(GPS)	ClientOSEnvGet('device_location')関数 …… 緯度と経度が返ります。

※外部コール「OS コマンド」のコマンド特性内の「実行」の値は、「C=クライアント」にします。



## 第 7 章 テクニック集

この章では、本システムのプログラム内で使用しているテクニックをいくつか紹介いたします。

### 7-1 グローバル変数／グローバル関数

メインプログラム(プログラム 1 番)で定義されている「変数／関数」は、システム全体で利用できる「グローバル変数／グローバル関数」になります。

#### ■グローバル変数

変数名	説明	
G.ユーザ ID	「ユーザ独自の値」 ログイン時に変数にセットしています。	
G.担当者コード		
G.担当者名		
G.課 CD		
G.課名		
G.Math.PI	円周率。距離計算で使用	地図表示で使用
G.Phy.Earth_R	地球の赤道半径	
G.MapFanAuthServer	MapFanAPI の認証サーバ	
G.MapFanSerchServer	MapFanAPI の住所検索サーバ	
G.ClientCache	イメージファイルの表示時に使用	顧客情報画面で使用
G.会社名	顧客大臣の会社情報取得用変数	システム情報画面で使用
G.プログラム No	プログラムの判別用変数	プログラムの実行時に使用
G.デバイス区分	デバイス情報の取得用変数	各画面表示時に使用
G.顧客 From	訪問予定検索画面で指定した顧客コードの範囲(開始)	顧客データ取得時に使用
G.顧客 To	訪問予定検索画面で指定した顧客コードの範囲(終了)	
G.日付 From	訪問予定検索画面で指定した日付の範囲(開始)	
G.日付 To	訪問予定検索画面で指定した日付の範囲(終了)	
G.担当者 CD	ログインしたユーザの担当者コード	
G.顧客 CD	顧客コード	

#### ■グローバル関数

関数名	説明	
getMapKey	MapFan に関して、認証 ID からアクセスキーを取得する パラメータ: 認証 ID 戻り値 : ステータス[成功か失敗か] とアクセスキー(文字型)	地図表示で使用
getGeoCode	住所から緯度と経度を求める パラメータ: アクセスキー、住所 戻り値 : 緯度と経度(文字型)・・・結果は xml なので解凍が必要	

## 7-2 地図表示

本システムでは、「MapFanAPI【別途要商用ライセンス】」を利用して地図表示しています。  
方法としては、地図表示用の html ファイルを用意し、その html の中に、地図表示に必要なデータを Magic から流し込むようなイメージです。

なお、本システムでは、地図上の顧客の場所(住所)に目印(マーカー)を立てるようにしていますが、そのためにあらかじめ、顧客の住所から緯度と経度を関数で取得しています。

関数名は「getGeoCode」で、これは独自に作った関数です。メインプログラム(プログラム 1 番)内で定義されており、得意先 CSV データを取り込むタイミングで、この関数を使って緯度と経度を取得しています。

## 【手順】

1. 地図表示に必要な html を用意します。(「MapFan」側で用意されています)
2. その html ファイルの中で、Magic からデータを流し込む箇所に修正を加えていきます。  
修正済みの html ファイルが、「SMARTSALES¥template」フォルダ内にあります。

<MapFan\_tplt.html の例>

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv='content-type' content='text/html; charset=utf-8' />
  <title>MapFanAPI</title>
  <style type="text/css">
    #sample_map {
      width: <!--$MG_width-->px;
      height: <!--$MG_height-->px;
    }
  </style>
  <script type='text/javascript' src='http://api-auth-pre.mapfan.com/v1/jslib/js/map/mfjsapi'></script>
  <script type='text/javascript'>

    /** * * * マーカー表示 * * *
    //--- イニシャル処理 ---
    window.onload = function() {
      Mfapi.mapHost = 'api-map-pre.mapfan.com';
      Mfapi.auth(<!--$MG_ID-->,task_func);
    }

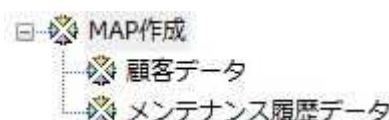
    //--- 各種処理 ---
    function task_func(authStatus) {

      //--- 認証結果確認と地図作成 ---
      if( authStatus != 'success' ) {
        return;
      }
    }
  </script>
</head>
<body>
  <div id='sample_map'>
    <!--$MG_body-->
  </div>
</body>
</html>
```

<!--\$MG\_名前-->の箇所をタグといい、Magic 用に追加したものです。このタグの箇所に Magic からデータが流し込まれます。

3. Magic の中からタグの箇所にデータを流し込むためのプログラムを作成します。設定は以下の通りです。

- 設定① バッチタスクを作ります。  
(プログラム 10.1 番、10.2 番)

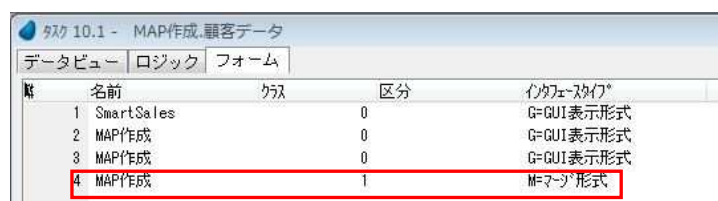


- 設定② 入出力ファイルを定義します。



ファイル名は、「ユニークになるような html ファイル名」にします。

- 設定③ フォームを追加します。



インタフェースタイプは「M=マージ形式」にします。

- 設定④ 今追加したフォームの「フォーム特性」内の値を設定します。



ファイル名は、【手順 2】の html ファイル名です。

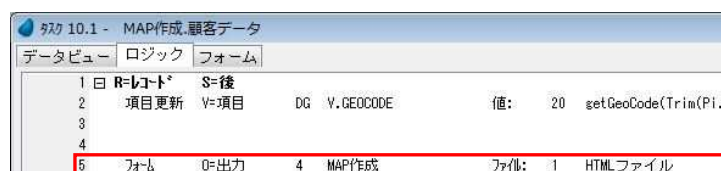
- 設定⑤ タグの名前と、タグに流し込むデータを定義します。

上の図の「タグテーブル」欄で F5 (ズーム) します。



html 内の「タグ名」と、それに流込むデータ(式も可)を定義します

- 設定⑥ フォーム出力コマンドを使って、データを出力します。



4. 地図を表示します。

リッチタスクのフォーム内に、ブラウザコントロールを貼り付け、そのコントロールの中身を html ファイルにします。  
(プログラム 19.1.1 番のフォーム)

## 第 8 章 顧客大臣連携について

本モジュールは応研株式会社様の DOP（Daijin ODDS Partner）様向けとなっております。

顧客大臣との連携については応研株式会社様各マニュアルを参照して下さい。

## 付録(モデル一覧・テーブル一覧・プログラム一覧)

### 【モデル一覧】

#	名称	クラス	型	書式
1	【汎用】			
2	Android 用フォームサイズ	リッチクライアント表示形式	フォーム	-
3	iPhone5s 用フォームサイズ	リッチクライアント表示形式	フォーム	-
4	iPhone6Plus 用フォームサイズ	リッチクライアント表示形式	フォーム	-
5	【共通項目】			
6	年月日	項目	日付	YYYY/MM/DD
7	時刻	項目	時刻	HH:MM:SS
8	権限	項目	数値	2
9	処理名	項目	文字	30
10	論理値	項目	論理	5
11	DUMMY	項目	文字	1
12	共通区分	項目	数値	1
13	メッセージ	項目	文字	30
14	顧客 CD	項目	数値	4
15	郵便番号	項目	文字	10
16	住所 1	項目	文字	40
17	住所 2	項目	文字	40
18	TEL	項目	文字	18
19	FAX	項目	文字	18
20	MailAddress	項目	文字	50
21	メンテナンスサイクル(月)	項目	文字	2
22	項目区分	項目	数値	5
23	項目連番	項目	数値	5
24	項目コード	項目	数値	5
25	項目名称	項目	文字	20
26	地図認証 ID	項目	文字	80
27	緯度・経度	項目	数値	3.7
28	距離	項目	数値	5.2
29	ファイル(フルパス)	項目	文字	256
30	レスポンス文字列	項目	文字	500
31	地図アクセス KEY	項目	文字	500
32	GEOCODE	項目	文字	500
33	地図情報文字	項目	文字	1000
34	地点情報	項目	文字	300
35	件数	項目	数値	3
36	【担当者マスタ】			
37	担当者 CD	項目	数値	4
38	氏名	項目	文字	40
39	フリガナ	項目	文字	40
40	課 CD	項目	数値	1
41	課名	項目	文字	40
42	ユーザ ID	項目	文字	14
43	パスワード	項目	文字	14
44	【商品・保守マスタ】			
45	商品 CD	項目	数値	4
46	名称	項目	文字	40

47	商品区分	項目	数値	2
48	【メーカーマスタ】			
49	メーカーCD	項目	数値	4
50	メーカー名	項目	文字	80
51	【消耗品サブライマスタ】			
52	消耗品 CD	項目	数値	4
53	製品名	項目	文字	80
54	型番	項目	文字	30
55	単価	項目	数値	8
56	【請求先マスタ】			
57	請求先 CD	項目	数値	4
58	名称 1	項目	文字	40
59	名称 2	項目	文字	40
60	カナ	項目	文字	40
61	【住居区分マスタ】			
62	住居区分 CD	項目	数値	2
63	住居区分名称	項目	文字	40
64	【顧客ランクマスタ】			
65	顧客ランク CD	項目	数値	2
66	顧客ランク名称	項目	文字	40
67	【顧客マスタ】			
68	顧客 CD	項目	数値	4
69	顧客名 1	項目	文字	50
70	顧客名 2	項目	文字	50
71	【納入先データ】			
72	納入先名 1	項目	文字	80
73	納入先名 2	項目	文字	80
74	FAX	項目	文字	12
75	登録日	項目	日付	YYYY/MM/DD
76	契約台数	項目	数値	2
77	当社営業担当者 CD	項目	数値	4
78	当社作業担当 CD	項目	数値	4
79	【貸借人データ】			
80	商号	項目	文字	80
81	設立日	項目	文字	80
82	本社所在地	項目	文字	20
83	担当者 部署	項目	文字	40
84	担当者 氏名	項目	文字	40
85	代表者	項目	文字	60
86	住居年数	項目	数値	2
87	住居区分	項目	数値	2
88	生年月日	項目	日付	YYYY/MM/DD
89	【機器情報データ】			
90	ユニークキー	項目	数値	9
91	機器情報コード	項目	数値	#9P0
92	購買累計金額	項目	数値	10
93	顧客ランク CD	項目	数値	2
94	納入日	項目	日付	YYYY/MM/DD
95	区分	項目	数値	2
96	保守開始日	項目	日付	YYYY/MM/DD
97	保守終了日	項目	日付	YYYY/MM/DD
98	リース開始日	項目	日付	YYYY/MM/DD
99	リース終了日	項目	日付	YYYY/MM/DD

100	【メンテナンス履歴】			
101	メンテナンス履歴コード	項目	数値	#9P0
102	受付日	項目	日付	YYYY/MM/DD
103	作業予定日	項目	日付	YYYY/MM/DD
104	受付担当者 CD	項目	数値	4
105	作業担当者 CD	項目	数値	4
106	機器 CD	項目	数値	4
107	内容	項目	文字	40
108	次回作業予定日	項目	日付	YYYY/MM/DD
109	【営業履歴】			
110	営業履歴コード	項目	数値	#9P0
111	最終アクション日	項目	日付	YYYY/MM/DD
112	次回訪問予定日	項目	日付	YYYY/MM/DD
113	経過月	項目	数値	3
114	訪問予定日	項目	日付	YYYY/MM/DD
115	訪問者 CD	項目	数値	4
116	【選択項目データ】			
117	選択項目インデックス	項目	数値	10
118	選択項目区分	項目	数値	5
119	選択項目コード	項目	数値	10
120	選択文字	項目	文字	20

## 【テーブル一覧】

#	名称	カラム	インデックス	外部キー	データソース	データベース
1	■■■■■マスタ■■■■■					Default Database
2	顧客データ	7	1	0	MST_KOKYAKU	Loal(Mst)
3	貸借人データ	5	1	0	MST_TAIYAKUNIN	Loal(Mst)
4	担当者マスタ	2	2	0	MST_TANTOSHA	Loal(Mst)
5	商品・保守マスタ	4	1	0	MST_HOSHU	Loal(Mst)
6	請求先マスタ	6	1	0	MST_SEIKYUSAKI	Loal(Mst)
7	消耗品サプライマスタ	2	1	0	MST_SHOMOHIN	Loal(Mst)
8	メーカーマスタ	2	1	0	MST_MAKER	Loal(Mst)
9	住居区分マスタ	3	1	0	MST_JUKYOKUBUN	Loal(Mst)
10	顧客ランクマスタ	2	1	0	MST_KOKYAKU_RANK	Loal(Mst)
11	項目マスタ	4	1	0	MST_KOMOKU	Loal(Mst)
12	選択項目データ	4	2	0	MST_SENTAKUDATA	Loal(Mst)
13	■■■■■データ■■■■■					Default Database
14	機器一覧データ	15	1	0	DAT_KIKIJOUHOU	Local(Tran)
15	メンテナンス履歴データ	16	1	0	DAT_MAINTENANCE	Local(Tran)
16	営業履歴	11	2	0	DAT_EIGYO_RIREKI	Local(Tran)
17	ログイン状況	10	2	0	DAT_LOGIN	SQLite(Tran)
18	日報入力	19	2	0	DAT_NIPPOU	Local(Tran)
19	■■■■■顧客大臣■■■■■					
20	メンテナンス履歴	26	2	0	RIREKI3	KOKYAKU_DAIJIN
21	営業履歴	19	2	0	RIREKI4	KOKYAKU_DAIJIN

## 【プログラム一覧】

#	名称	フォルダ	公開名
1	メインプログラム		
4	マスタデータ取得		
5	履歴データ取得		
6	項目名称取得		
7	認証 ID 取得		
8	会社情報取得		
9	MAP 作成		
12	スタート		Start
13	ログイン		
14	トップメニュー		
15	訪問予定検索		
16	担当者選択		
17	顧客一覧		
18	顧客情報		
19	当日メンテナンス地図表示		
20	メンテナンス一覧		
21	システム情報取得		
22	ナビゲーション(トップメニュー)		
23	ナビゲーション(顧客情報)		
24	ナビゲーション(地図検索)		
25	ナビゲーション(システム情報)		
26	ナビゲーション(日報入力)		
27	日報入力		
29	顧客選択		