

Android でのMagic RIA

はじめの一歩



magic 目次

- ≗ 目的と前提条件
- Hello Magic プログラム
- Android RIAプログラムの特性
- ・ カスタムモジュールの作成
- 付録

magic 目的と前提条件

- 目的: Magic xpa 2.2 を使った、Android RIA プログラム開発の基本 を理解することが目的です。
- 環境的な前提条件としては:
 - Windows PC上に、Magic xpa 2.2 Enterprise Studio (あるいは体験版) をインストールしてあること。
 - Android デバイスを所有していて、テストに使えること。※ Android SDKのエミュレータでも可能です。
 - Magic xpa Studio のあるPCへ、Androidからネットワーク接続できること。
- ・ 読者のスキルの前提条件としては:
 - Magic xpa 2.2 (あるいは uniPaaS V1Plus) で、Windows RIA システムを作成した経験があること。
 - 2 Androidアプリの開発経験は前提としません。(経験あればもっと良い)
 - ※ サポートしているWindowsバージョン、Androidバージョン等については、 弊社HPを参照してください。

http://devnet.magicsoftware.co.jp/magicxpa/magicxpa2/dbmsmagicxpa2/

- ・本書は、製品添付「モバイルRIA開発者ガイド」をもとに、基本的なことだけを抜き出し、ステップバイステップで実行できるように編集したものです。
- ・「モバイルRIA開発者ガイド」は、モバイル RIA (Android, iOS 共に) に固有な技術情報を包括的にまとめてあり、本書に掲載 されていない詳しい情報も多くありますので、モバイルRIA開発 に先立ち、そちらもご一読ください。



Hello Magic プログラム

何はともあれ、まずは最も簡単なプログラムを例に

取って、Android RIAを動かしてみましょう

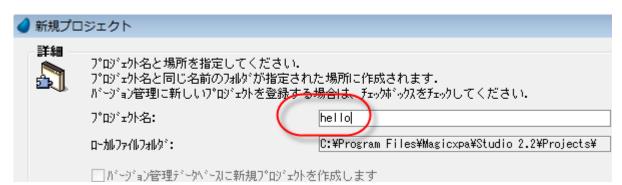


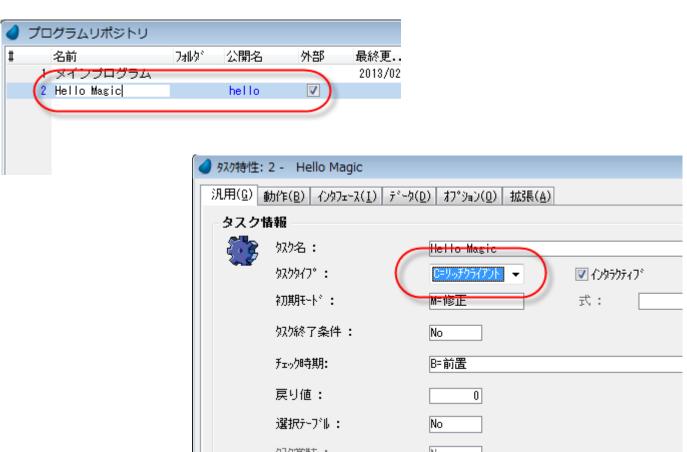
Android デバイス上で実行する前に、まずPC上での準備をしておきましょう。

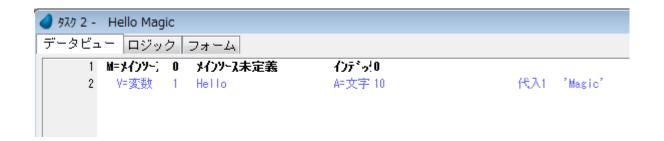
- Magic xpa Enterprise Studio をインストールします(説明省略)
- 2. Pervasive PSQL をインストールします(説明省略)
- 3. RIAサンプルプログラム Hello Magic を作成します。
- 4. Windows RIAで動作確認します。

Android RIA アプリケーションも、Windows RIAと同じようにして作成します。

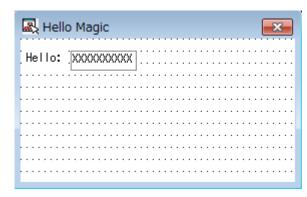
- 1. Hello プロジェクト作成
- 2. Hello Magic プログラム作成 (公開名hello)
 - ∘ 変数表示のみのタスク
 - ・※「外部」にチェックを入れるのを忘れないように







フォームエディッタ

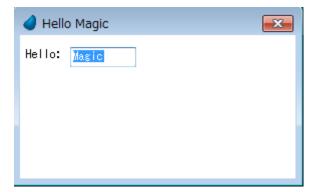




F7実行 (Windows RIA)で確認

- Studio プログラムリポジトリから、F7 でテスト実行してみます。
- Magicが正しくインストールされていれば、 問題なく実行されるはずです。
- ここで何らかのエラーが出るようであれば、 Android に進む前に、エラーを解決しておいてください。

F7実行



Maroid 上で実行してみましょう

- Windows RIAで実行を確認したら、いよいよ Android 上で実行してみましょう。
- 手順は、大きく分けて、以下のようになります:
 - 1. Androidモジュールをインストールします。
 - 2. DevProps.txt を編集、Webサーバ上に置きます。
 - 3. RIAサーバを待機状態にしておきます。
 - 4. Androidモジュールを起動します。
- → これで動くはず!!



インストールに先立って準備

- Androidのデフォルトの設定では、Google Play からダウンロードしたアプリしかインストールできないようになっています。(エラーが出ます)
- Magic のRIAモジュールをインストールするには、この制限を解除しておく必要があります。
 - ※ インストール時にだけ解除して、インストール後には元に戻して良いです。

インストールがブロックされました。		
ご使用の端末は、セキュリティのため、Google Playから 取得していないアプリケーションのインストールをブ ロックする設定になっています。		
キャンセル	設定	

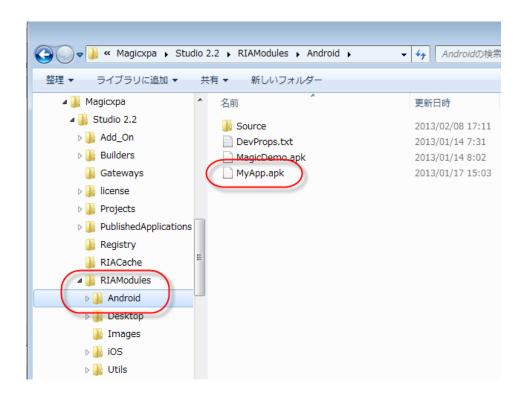


アプリインストール制限を解除するには

- Android の「設定」→ セキュリティ
 → 提供元不明のアプリ にチェック
 を入れます。
 これで解除されます。
- 制限をもとに戻すには、このチェックを外します。



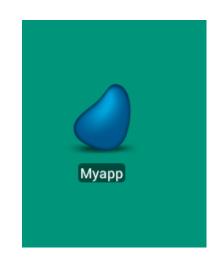
Magic xpa の Android RIA 用のモジュールは、 RIAModules¥Android¥ MyApp.apk にあります。





Androidモジュールをインストール

- 1. アプリ MyApp.apk をAndroid デバイスにコピーします。
 - ※ ファイルをコピーするには、次のいずれかの方法によります。
 - Webサーバからダウンロード (IISのMIME設定必要 → 付録参照)
 - ② マイクロSD
 - ≗ USB接続
 - ♪ メール
 - クラウドファイル共有サービス
- 2. MyApp.apk をダブルクリックします。
- → モジュールがインストールされます。



- 添付 MyApp.apk は、開発時動作確認用の暫定モジュールです。
- ・ 実運用時には、MyApp.apk は使わず、アプリケーション用にカスタマイズしたモジュールをビルドして利用します。
- ・カスタマイズの項目と方法 → 別冊で説明。

- ・ DevProps.txtは、実行しようとするMobile RIA アプリケーションについて、サーバ側情報を定義したファイルです。
- C:¥InetPub¥wwwroot¥DevProps.txt に、以下のような内容でファイルを作成します。
- ※ サーバ構成/プロジェクト名/公開プログラム名に合わせて、設定を変更してください。



(補足) DevProps.txt という名前について

- DevProps.txt という名前は、任意に変更して構いません。
- ・以下では、DevProps.txt という名前を前提に説明します。異なる名前にした場合には、それに応じて変更してください。
- 拡張子は .txt のままにしておくのが無難です。

※ 拡張子を変えると、IISのセキュリティ機能 に引っかかることがあります。

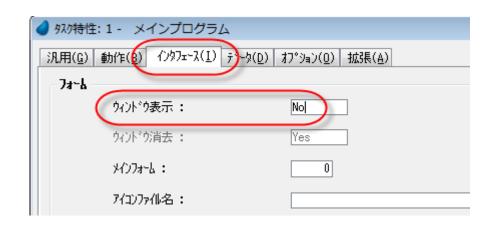
- Webサーバでアクセスできることを確認してください。
- 例: ファイルが C:¥InetPub¥wwwroot¥DevProps.txt に置いてあれば、URL は http://MGSTUDIO/DevProps.txt となります。
- ・Webブラウザから、このURLを指定して、DevProps.txtの内容が正しく表示されることを確認します。
- ※ この URL は、RIAクライアント起動時にアクセスします

- DevProps.txt は、PC上のWebブラウザだけでなく、Androidデバイス上でのWebブラウザからも行なってみてください。
- (経験上) Androidでは、LAN上のホスト名解決ができないことがあるようです。その場合には、「サーバが見つからない」というようなエラーとなります。
- もし、Windows PC上のWebブラウザではOKなのに、Android のWebブラウザでダメならば、Windows PCのホスト名 (MGSTUDIO)を数字のIPアドレス (例:192.168.1.8 とか。 ネットワーク環境により違います)に変えてみてください。
- ※ その場合には、DevProps.txt 内のホスト名も、IPアドレスに変えてください。



メインプログラム画面の表示を抑制します

- サーバを待機状態にしておく 前に、一点、プロジェクトを修 正します。
- メインプログラム → タスク特性 → インターフェース タブ → ウィンドウ表示を No に設定します。
- これにより、メインプログラム の画面が表示されないように なります。



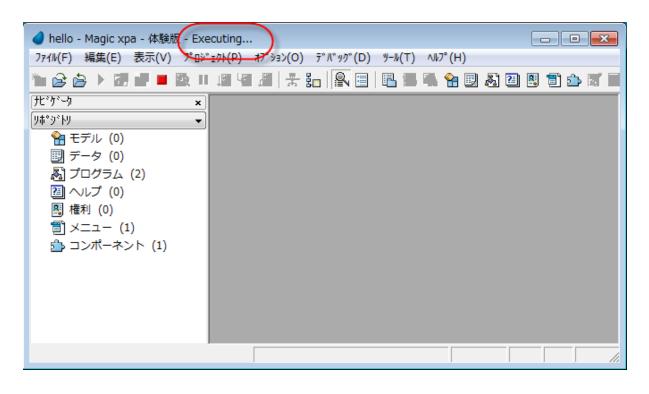
参考: メインプログラムの画面を非表示にする理由は、付録「メインプログラムのウィンドウ表示」を参照してください。



サーバを待機状態にしておきます

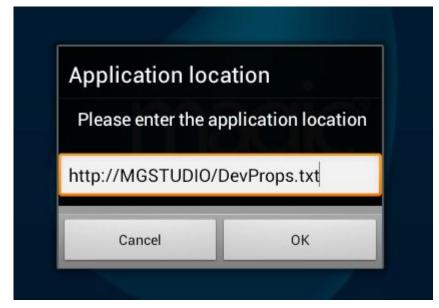
・「デバッグ → プロジェクトの実行(J)」で 待機状態にします→ タイトルが Executing... になります。





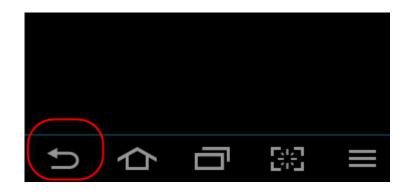
- Android 上で、MyApp をタップします。
- アプリケーション情報を入力する ダイアログが出るので、 DevProps.txt のURLを指定します。
- 👱 成功!!(のはず)

- ➤ このURLの情報は記憶されるので、最初に 一度入力したら、二回目以降は出て来ま せん。
- ▶ ホスト名が正しく認識されない場合には、 IPアドレスを指定してください。



Hello Magic		
Hello:	Magic	

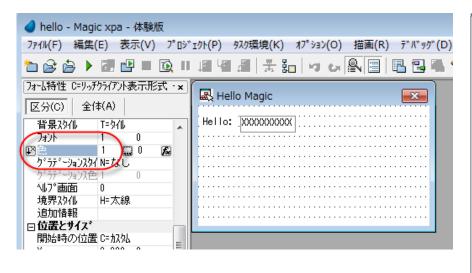
・終了するには「戻る」ボタンを押します。

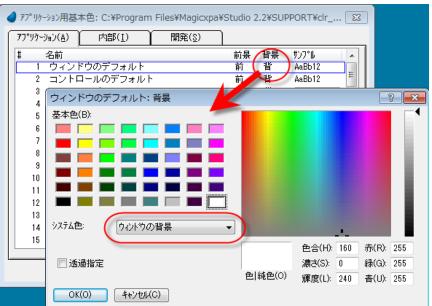




気になる背景色を変更しましょう

背景が黒色で表示されている。変えたい・・・← フォームの色 1 が、Windows システム色のためです。





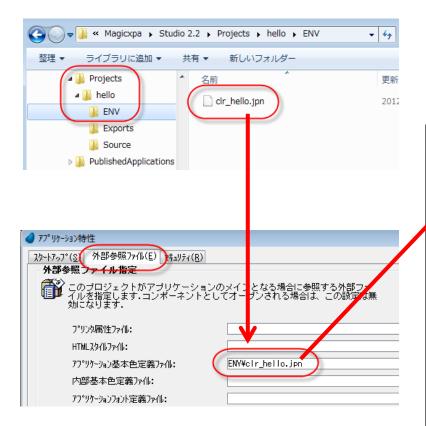
※ Windows システム色の場合、Android のデフォルトの動作(=黒色)となります。

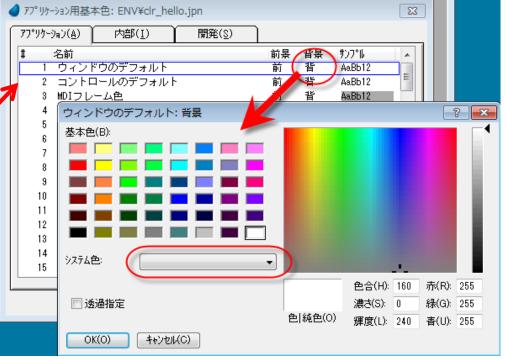
magic

色定義ファイルを変更します

- clr_rnt.jpnをコピーして、このアプリケーション専用の色設定ファイルにしましょう。
 - プロジェクトのフォルダの下に、ENVフォルダを作成します。
 - clr_rnt.jpn を、プロジェクトのENVフォルダにコピーして、 clr_hello.jpn に改名します。
 - このファイルを、アプリケーション特性 → 外部参照ファイル タブ → アプリケーション基本色定義ファイル に設定します。

 - ・ 色 1 の背景で、システム色の設定を外します。







再度実行して確認しましょう

- 再度実行してみます。
 - → 今度は背景が白色になったはずです。





Android RIAの特性を理解しましょう



- Android のような、MobileデバイスとWindows PCを比べると、 デバイスのハードウェア特性の違い、OS の違い、ユーザ・イン ターフェース (UI) の基本動作やコンポーネントなどが大きく違います。
- このため、Windows RIAの開発と同じ感覚で作成していると、 Android デバイス上で実行すると、非常に使いにくいものになってしまう可能性があります。
- ・ アプリケーションを開発するに先立って、Android デバイスの動作の特徴をよく理解して、アプリケーションを設計しましょう。

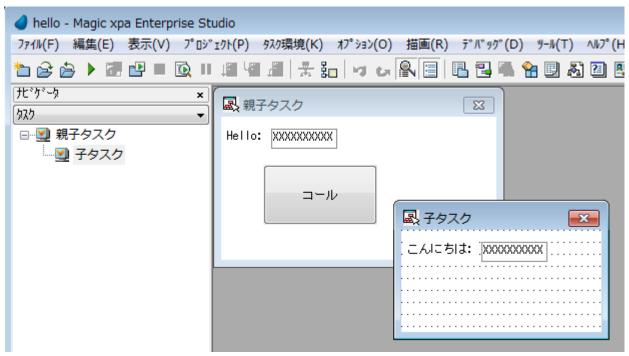


Android RIAアプリのフォーム表示について

- Mobile デバイス (Android、iOSとも) 上のアプリの一般的な特性として、Windows アプリとはUIがかなり異なります。
 - ② (基本的に) すべてフルスクリーン画面 → モーダル
 - ② タスクの終了 = 「戻る」ボタン (ESCキーに相当)
 - ・ HOMEキーでは終了しない (背後で動作し続けている)



- ※ 簡単な親子タスクを作って、実際に確認してみましょう。
- ボタンで子タスクを呼び出すだけの、簡単な親子タスクを作ります。



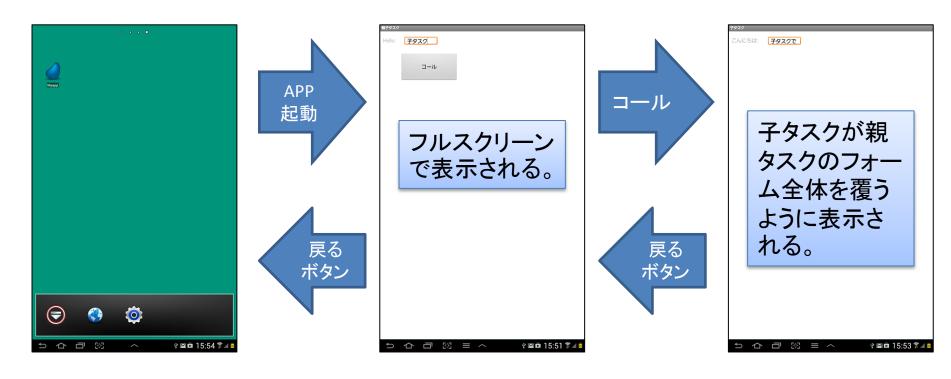
親タスク、子タスクの画面が、フォームエディタで定義された 通りの大きさと位置で表示されます。





Android RIAでは・・・

- DevProps.txt を編集し、公開名を、新たに作成した親子タスクに変更します。
 - cproperty key="prgname" val="hello2"/>
- 実行すると、子タスクはフルスクリーンで表示され、親タスクのフォーム全体を覆うように表示されます。



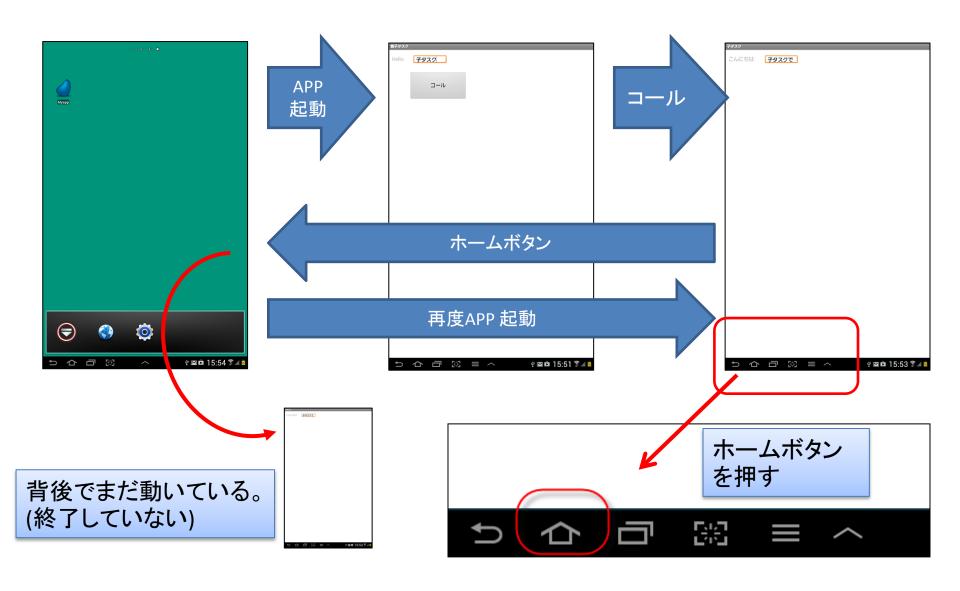
アプリが終了 する。



「ホーム」ボタンの動作は?

- Android には、「戻る」ボタンの他に、「ホーム」ボタンがあります。
- どちらを押しても、アプリの画面が消えるので、一見、どちらでも 良さそうに思えますが、動作に大きな違いがあります。
 - ホームボタンでは、アプリは終わりません。
 - ・ホームボタンは、アプリの画面を隠すだけ。Windowsの「最小化」みたいなものです。
 - 事度アプリを起動すると、直前の画面が復元されます。





正しいアプリの終わり方は?

- 最初のタスクで、「戻る」ボタンを押す。
- ・ 最初のタスクで、「クローズ(C)」イベントを発行する。
- 「終了(X)」イベントを発行する。※「終了」イベントではありません。

magic キーボード入力は?

- Androidデバイスでは、基本的にハードウェアキーボードやマウスがついていません。
- ② 次のようなケースではハードウェアキーボードを使えますが、一般的には使えないという前提でアプリケーションを設計する必要があります。
 - タブレット型のデバイスには、ハードウェアキーボードが付いているものもあります。
 - Bluetooth キーボードをサポートしているものもあります。
- データ入力は、次のものを使って行います。
 - ・ 指でのタップ (マウス代わり)
 - ソフトキーボード
- ハードウェアキーボードがないので、Windows RIAで当たり前にできていたことができなくなることが多々あります。
- ・「何ができて、何ができないか」は、ソフトキーボードの機能に大きく依存します。
- Androidの場合、ソフトキーボードは各社各様ですので、利用予定のデバイスに付属のソフトキーボードの仕様も把握しておく必要があります。



ソフトキーボード例







- ※ ソフトキーボードは各社各様ですが、一般的な制約事項を挙げると・・・
- Tab キーがない → 指でタップして移動
- Ctrl キーがない →
 - ・ モード変更 (Ctrl+M, Ctrl+E、Ctrl+Q) できない
 - ² コピペ (Ctrl+C、Ctrl+P、Ctrl+V)できない
- ファンクションキーがない
 - ズーム (F5)、キャンセル (Ctrl+F2) 不可
- 矢印キーがない (↑、↓、→、←)
 - ・カーソル移動は指でタップするしかない
 - ≥ 新規行作成できない





キーボードの代わりの手段を提供しましょう

これらの制限に対応する手段を提供するようにしましょう。

- ボタンの追加: 行作成、行削除、タスクモード切り替えなどをボタンに割り当てます。
- メニュー: メニューに内部イベントを 設定します。
- プログラムロジック: 行の最後の項目に入力がされれば、「次行」イベントを発行する、などのロジックを組込みます。



#	タイフ。	コーザ メニュー名	論理メニュー名	ハ°ラメータ	
1	V={\^*)\	登録(&C)		内部个"小:	登録(C)
2	V=ፈላ*ንኑ	修正(&M)		内部个`外:	修正(M)
3	V={\^*}\\	照会(&Q)		内部个`外:	照会(Q)
4	V={\^*}\\	行作成(&R)		内部个`沙:	行作成()
5	V=Z^*)F	行削除(&D)		内部个`가:	行削除(D)

照会 - 顧客マスタ

顧客番号	顧客	名	住所		
1008	千葉/	ペット:	千葉県千葉	東市高柳 1	
3201	コジ	マペット	東京都足立	Z区綾瀬 3 -	
3220	ペッ	トショ	東京都江戸	川区南篠崎	
3321	ヤザ	‡金魚	東京都杉立	区高井戸東	
3440	ペット	トサロ:	東京都文京	区小石川 2	
3550	ペット	トワール	愛知県名さ	屋市名東区	
4920	ANIN	1AL HO	京都府京都	『市伏見区醍	
5133	山⊞⊄	ペットノ	岡山県岡山	」市表町3 -	
5387	フク	トミ鳥(佐賀県唐津	門市和多田本	
5493	わんね	bんペ [、]	北海道札幌	現市厚別区厚.	
5678	サン	シャイ:	東京都豊島	昌区池袋 4	
6238	アラ	ジンア:	宮城県仙台	計青葉区川	
6588	ペット	トハウン	秋田県秋田	B市東通3 -	
6688	犬猫:	ブラザ-	山形県山形	/市平久保 1	
7834	酒田县	兵衛門	福島県郡山	」市安積町成	
7963	(有)ト	・ンビ鷹	茨城県鹿嶋	島市高天が原	
1.7	‡ ュ-	ーダブリ	埼玉県浦利]市常盤 5 -	
	録(C)			修正(M)	
照会(Q)		行作	-成(R)	行削除(D)	



フォームやコントロールの特性を理解しましょう

- Magic Studio のフォームエディッタは、Windows のGUI部品によって表示されます。
- 一方、Androidデバイス上で実行する際は、Android のGUI部品に よって表示されます。
- したがって、フォームエディタ上と実際の画面で見え方が異なります。
- アプリのフォーム設計を始める前に、フォームエディタと実際の表示との違いについて、よく把握しておきましょう。
- 主な確認項目は次のようなものです:
 - ・フォームサイズ
 - ・ 文字の大きさ
 - コントロールの形状 (エディット、プッシュボタン、コンボボックス、 チェックボックス) とサイズ
 - [●] Windowsシステム色の効果
 - Android独自のフォント

- iOS の場合には、iPhone、iPad、iPad mini で、デバイスの種類はそう多くはありません。
- Androidの場合には、非常に多種多様なデバイスがあり、画面解像度も全部違います。
- ・「すべてのデバイスに対応したフォーム」を設計するのは、事実上不可能です。
- ・利用予定の機種を選定し、それを前提としたフォーム設計を行います。











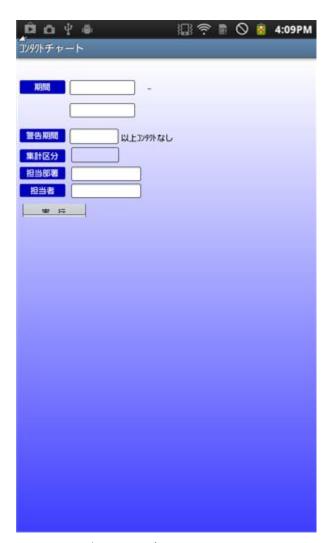


- ・フォーム設計の第一歩は、フォームサイズの把握です。
- ・ 大きすぎても、左右・上下にスクロールして見ることはできますが、使用感は悪くなります(※)。
- Magicのフォームサイズ自動調整機能は有効で、多少の画面サイズの違いは、これで吸収可能です。 しかし、あまり大きく違うと画面が間延びしたり、重なってごちゃごちゃ詰め込みすぎになったりしてしまいます。

※ スクロールについての注意事項があります → 「フォームのスクロールの基本」以下参照。

□ □ ♀ → □ ○ № 4:14PM OM 受注
受注番号 900 顧客情報 受注日 :013
商品番号 商品名 4
明細合計額
消費税額 受注合計額
一

自動調整がうまく行かず、ごちゃごちゃしすぎの画面



まのびしすぎた画面



画面サイズを測りましょう

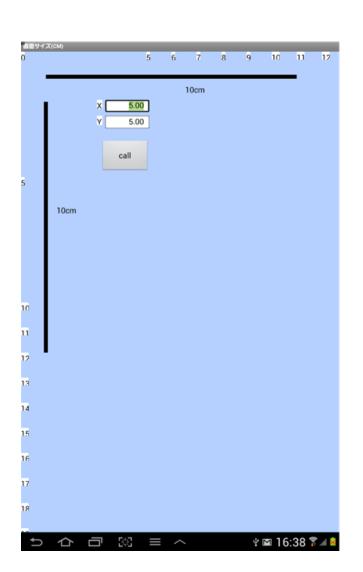
- 寸法単位は、「C=センチメートル」にするのが、一番簡単なようです。
- 実際のデバイスの画面サイズを cm で測り、フォームのサイズとします。



理屈通りにはいかない・・・

現実の Android デバイスで表示させてみると・・・

- 寸法単位を「C=センチメートル」として も、指定した通りのサイズに表示され るとは限りません。
- 更に、縦と横で長さが微妙に違うもの もあります。
- 右のような画面を作って、実際のサイズを計測してみましょう。



画面サイズ計測例

- ² フォームエディタ上で 10 cm のものが・・・
 - Samsung Galaxy SC-01C: ほぼ 10cm で表示された。 (正確)
 - Samsung Galaxy Tab 7.7 (SC-01E): 8.1cmで表示された。
 - Nexus 7: 横は9.1cm、縦は9.4cm で表示された

参考:なぜ、このような結果になるのですか?

- ➤ Magicは、Android OS のAPIを通して、画面の dpi (Dot per inch) を取得して、それをもとにコントロールのサイズ計算を行います。
- ▶ フォームエディタ上で10cmになるはずのものが、10cmで表示されないのは、APIが返すdpi値と、現実のdpi値とが異なっているためです。

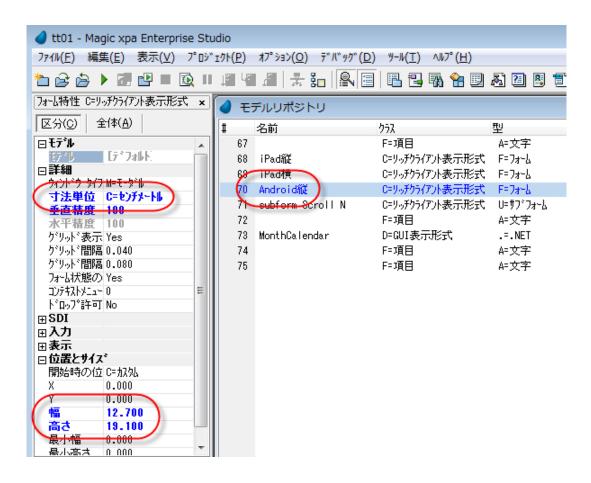
magic フォームサイズの計算

- 画面の物理サイズ = フォームサイズ × 縮小倍率なので、
 フォームサイズ = 画面の物理サイズ ÷ 縮小倍率となります
- 例: Galaxy Tab 7.7 SC-01E の場合
 - ・ 画面の物理サイズ (タイトル部分を除く): 10.3 × 15.5 cm
 - ・縮小倍率: 10cm のものが 8.1cm で表示されたのだから、0.81
 - ・ ∴ フォームサイズは
 - ^⁰ 横 (W): 10.3/0.81 = 12.7cm
 - ² 縦 (H): 15.5/0.81 = 19.1 cm



フォームモデルを登録しましょう

基本のフォームサイズが決まったら、モデルに登録しておきましょう





基本フォームサイズを決めます

- ・上記のような計測を、利用予定の各デバイスについて行います。
- その中で、小さめのものを基準にして、基本フォームサイズを決定しましょう。
- ・ 画面の大きさが大きく異なるフォームモデルを、2~3種類ほど 登録しておくのがよいでしょう。
 - ・ 大 → タブレット向け
 - 小 → スマホ向け
 - ・中 → (必要に応じ) 小型タブレット

- コントロールについても、Studioのフォームエディタ上の表示と、 Androidデバイス上での実際の見え方は異なります。
 - エディットコントロール
 - プッシュボタン
 - コンボボックス
 - チェックボックス
- ∘ 一般に、Windows RIAと比べ、
 - ・余白や境界線が大きいので、そのままだと文字が潰れます。
 - サイズを大きめに作成してください。
 - 機種により見え方が異なるので、実機で確認しましょう。





Galaxy SC-01C

Nexus 7



文字の大きさを確認しましょう

- フォームが決まったら、次に、標準文字サイズを決めましょう。
- 同じ9ptのフォントでも、実際に表示される大きさが異なることが多いようです。
- 文字が大きすぎると、表示できる内容 が少なくなります。小さすぎると読み にくくなります。
- ボタンが小さすぎると、指で押すのに 苦労します。
- これも、実機で使用感を確認しておきましょう。

画面フォント	
1234567890	123456789012345678
gothic 5	あいうえお12345ABCDE
gothic 6	あいうえお12345ABCDE
gothic 7	あいうえお12345ABCDE
gothic 8	あいうえお12345ABC
gothic 9	あいうえお12345A
othic 10	あいうえお12345
2	button 5pt
3	
4	button 6pt
5	button 7pt
6	button 1pt
7 8	button 8pt
-	hutton Ont
20	button 9pt
2	hutton 10nt

mogic エディットコントロールでのデータ入力

- Android上でエディットコントロールにデータを入力するには、一般的にはソフトキーボードを使います。
- ソフトキーボード操作については、すでに「ソフトキーボードでの 制約は?」以下で説明しました。
- それ以外に次のような点に留意してください。
 - マスク文字の扱い:入力完了後に適用されます。
 - ② IME制御 (「漢字入力」特性): ソフトキーボードの種類の自動設定として実装されています。
 - 日付・時刻型: DateTime Picker が使えます。
 - ソフトキーボードを表示させないようにするには?



マスク文字の扱い

マスク文字

マスクなし:

- ・ 入力中は、マスク文字 が適用されません。
- エディットコントロール から抜けた時点で適 用されます。

Mask: 示されている X/XX/XXX/XXXX マスクあり: 111 マスクなし: 入力中:マス マスク文字 ク文字は表示 Mask: X/XX/XXX/XXXX されない。 マスクあり: mgまほう マスクなし: 入力終了後 (フォーカ マスク文字 ス移動): 入力データ にマスク文字が適用 Mask: X/XX/XXX/XXXX されて表示される。 マスクあり: m/g/ま/ほう

初期状態:マ

スク文字が表

参考: ソフトキーボードの 入力予測機能を活用する ための仕様です。



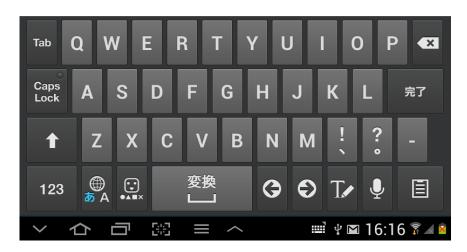
• Android RIAでのIME制御機能は、ソフトキーボードの種類の 指定、という形で実現されています。



IME制御: 文字型およびUNICODE型の場合

- 標準:かな漢字入力あり。入力 予測あり。
- 半角英数:かな漢字機能なし。入力予測なし。
 - 2 「漢字入力」特性では、8 (半角英数) のみ有効。
 - K書式では、KO および K8 (いずれも、 半角英数)のみ有効。

注意: ソフトキーボードの機能に依存します。







- 数値:数字キーパッドが表示されます。
- かな漢字機能、入力予測はありません。





日付・時刻型: DateTimePickerが使えます

- 日付型、あるいは時刻型のエディット項目にカーソルが移動すると、 DateTimePicker が表示され、値を設定することができます。
- (注意)時刻型でDateTimePickerを使 うと、秒単位の指定ができません。秒は 00 になります。





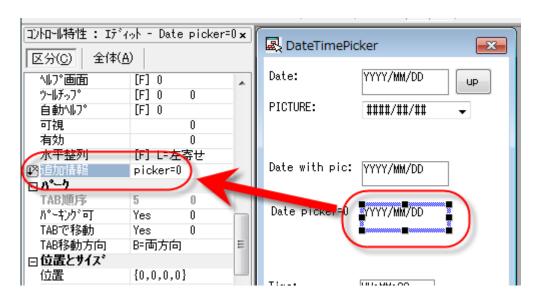
TimePicker

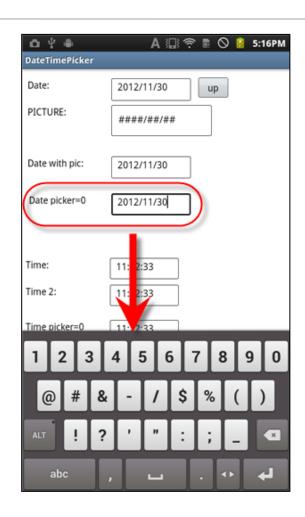
DatePicker



DateTimePickerが出ないようにするには・・・

DateTimePicker を表示させないで、 ソフトキーボードで入力できるように したい場合には、エディットコント ロールの「追加情報」特性に、 picker=0 と指定します。





ソフトキーボード が表示される



ソフトキーボードを表示させないようにしたい・・・

- ソフトキーボードは、画面の1/3~半分の領域を占有してしまうので、入力を行わない項目では、ソフトキーボードを表示させないようにしたい、と思うことがあります。
- ソフトキーボードは、入力(修正)ができないところでは表示されないので、Magicプログラム側で、修正できないように設定してやります。
 - タスクを照会モードとする → 全てのエディットコントロールで、 ソフトキーボードが出ないようになります。
 - ュディットコントロールの「修正許可」を No にします。

※ Windowsシステム色が指定されると、Androidのネイティブの表示方法で表示されます。

フォーム:背景が黒になります





エディット: 大きめの枠が表示されます Edit + Windows Sys ColorWindowsシステム色でない:魔法のリッチWindowsシステム色:魔法のリッチ透過背景:魔法のリッチ

・ コンボボックス: 下矢印が表示されます



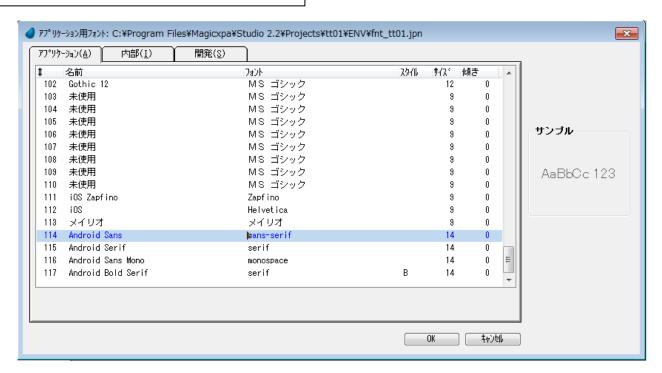
- Android に標準で備わっているフォントの数は限られています。
 - sans-serif → 飾りが無い (ゴシック似)
 - serif → 飾りがある (明朝似)
 - ⁹ monospace → 文字幅が一定
- これらの名前のフォントはStudioのフォントテーブルでは設定できないので、フォント定義ファイルにテキストエディッタで直接修正します。



Android フォント設定例

. . .

iOS Zapfino,Zapfino,9,128,0
iOS,Helvetica,9,128,0
メイリオ,メイリオ,9,128,0
Android Sans,sans-serif,14,128,0
Android Serif,serif,14,128,0
Android Sans Mono,monospace,14,128,0
Android Bold Serif,serif,14,128,0,Bold





Androidフォント 表示結果

Droid Fonts

Default: ijklmnlJKLMN123いろはイロハ

Sans: ijklmnlJKLMN123いろはイロハ

Serif: ijklmnIJKLMN123いろはイロハ

Sans Mono:

ijklmnIJKLMN123いろはイロハ

フォームの「追加情報」欄にフラグを指定して、以下のようなことができます。

- orientation: 縦長、または横長に強制的に設定する。
- popup: ポップアップフォームにする。
- VerticalScroll, HorizontalScroll: スクロールの方法を指定。



フォーム「追加情報」① Orientation

- ・ 方向の固定をしたい場合には・・・
 - ^⁰ orientation=Portrait (横長に固定)
 - ≗ orientation=Landscape (縦長に固定)



フォーム追加情報 ② Popup

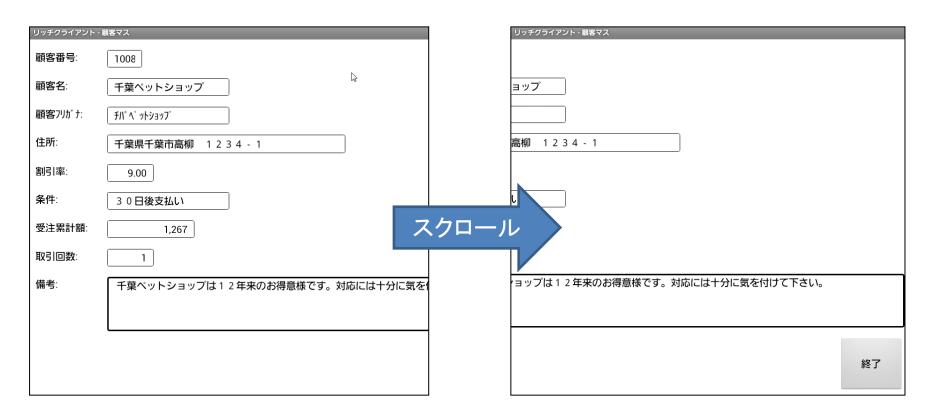
- ポップアップフォーム
 - popup=Y
- → ポップアップウィンドウになります。





フォームのスクロールの基本

- ・フォームが画面サイズより大きい場合、スワイプ動作でスクロールできますか?
- ・ フォーム全体がスクロールします。(縦方向、横方向ともに可能)





テーブルがある場合のスクロール

- ただし、テーブルがある場合には
 - ・レコードがスクロールします(縦方向)
 - 横方向、フォーム全体はスクロールしません。

- ※ iOS の場合には、テーブル内のレコードのスクロール、 フォーム全体のスクロール両方が可能でしたが、Android の 場合にはどちらか一方だけが可能です。(これはAndroidの制限事項です)
- この動作は、テーブルの他に、ブラウザ、サブフォームがある場合にも適用されます。



テーブルがある場合のスクロール

顧客番号	顧客名		顧客フリガナ	できる。	割引率
1008	千葉ベットシ	ョッブ	<i>チハ</i> ゚ペットショップ	千葉県千葉市高柳 1234-1	9.00
3201	コジマベット		コシ゚ マペ ット	東京都足立区綾瀬 3 - 1 1 - 5	5.00
3220	ベット	グワンワ	ላ [*] ፇトショップ ワ ンワン	東京都江戸川区南篠崎町 3 - 3 2 2	10.00
3321	ヤザキ		ヤザキキンキ ゙ョ	東京都杉並区高井戸東 3 - 5 - 6	5.00
3440	ペット	ヤザワ	ላ [*] ፇトサロンヤサ [*] ワ	東京都文京区小石川 2 - 5 - 7	3.00
3550	ベット	ドハート	ላ [*]	愛知県名古屋市名東区高針 3 6 5 2	8.00
4920	ANIMA	SE	アニマルハウス	京都府京都市伏見区醍醐東大路町23	5.00
5133	山田へ	ウス	ヤマタ゛ペ ットハクス	岡山県岡山市表町3-6-9	3.00
5387	フクト	友	フクトミトリノトモ	佐賀県唐津市和多田本村5-5	5.00
5493	わんわ	۲	ワンワンヘ * ット	北海道札幌市厚別区厚別西3条22-3	3.00
5678	サンシ	ベット	サンシャインヘ゛ット	東京都豐島区池袋 4-1-12	15.00
6238	アラジンアニ	マル	アラシ゛ンアニマル	宮城県仙台市青葉区川平1-5-3	5.00
6588	ベットハウス	シャボン	ペット <i>ハウスシャホ</i> ゚ン	秋田県秋田市東通3-11-22	4.00

フォーム全体のスク ロールはできない。 (縦、横とも)







テーブルがある場合のスクロール

- · ということは・・・
 - 画面外、右下にあるボタンは、表示できません。
 - 右側にはみ出したカラムは表示できません。

はみ出た部分は表示できません

これが画面範囲とすると・・・

	顧客名	顧客ツがナ	住所	割引率	条件	受注累		取引回数	
1008	千葉ベットショップ	₹パペットショッフ°	千葉県千葉市高柳 1234-1	9.00	30日後支払い		1,267	- 1	千葉ペットショップは12年来のお得意様
3201	コジマペット	J9*7^*%	東京都足立区綾瀬 3-11-5	5.00	70.00		2,035	1	犬(自家繁殖あり)・美容・ホテル
3220	ベットショップワンワ	ላ°ットショッフ°ワンワン	東京都江戸川区南篠崎町 3-322	10.00	30日後支払い				小動物・小鳥・観賞魚
3321	ヤザキ金魚	† † °‡ † >‡°₃	東京都杉並区高井戸東 3-5-6	5.00	30日後支払い		8,140	1	宅配、通信販売
3440	ベットサロンヤザワ	ペットサロンヤザワ	東京都文京区小石川 2-5-7	3.00	現金				美容・ホテルあり
3550	ベットワールドハート	^* >トワールドハート	愛知県名古屋市名東区高針 3652	8.00	45日後支払い				犬猫鳥からエキゾチックアニマルなど何で
4920	ANIMAL HOUSE	アニマルハウス	京都府京都市伏見区醍醐東大路町23	5.00	現金				障害犬・猫専門
5133	山田ペットハウス	ヤマタドヘキットハウス	岡山県岡山市表町3-6-9	3.00	現金				ベット用品の通信販売もあり
5387	フクトミ鳥の友	フクトミトリノトモ	佐賀県唐津市和多田本村5-5	5.00	現金				犬・猫・美容・ホテル・小動物・小鳥
5493	わんわんペット	<i>የ</i> ንምን^°ット	北海道札幌市厚別区厚別西3条22-3	3.00	45日後支払い				犬・美容・ホテル・小動物
5678	サンシャインペット	サンシャインへ°ット	東京都豊島区池袋 4-1-12	15.00	45日後支払い				海水無脊椎・カメ・グリーンイグアナなど
6238	アラジンアニマル	アラジドンアニマル	宮城県仙台市春葉区川平1-5-3	5.00	現金				犬種予約注文販売
6588	ベットハウスシャボン	^°っトハウス シャボン	秋田県秋田市東通3-11-22	4.00	現金				小動物・爬虫類・昆虫も取り扱い
6688	犬猫ブラザーズ	イヌネコフドラサドースド	山形県山形市平久保1-35	2.00	現金				フード・トリミング・ホテル
7834	酒田兵衛門錦鯉園	サカクヒョウベエニシキゴイエシ	福島県郡山市安積町成田29-3	5.00	現金				金魚・錦鯉の生産、販売
7963	(有)トンビ鷹	わたずか	茨城県鹿嶋市高天が原3-5-1	3.00	現金				鳥類専門
8135	キューダブルイー	キュータッフッルイー	埼玉県浦和市常盤5-6-7	7.00	現金				総合ベットショップ・
8256	春日部ガーデン	カスカヘッカットテック	埼玉県春日部市八丁目100-1	4.00	現金				爬虫類・小鳥・昆虫ほか
8378	ブラックバード	ブ [*] ラックバード	神奈川県横浜市春葉区元石川町1101-1	5.00	現金				無料しつけ教室実施中
8574	エンゼルドッグ	エンセドルト・ックド	山梨県東山梨郡勝沼町勝沼34-5	8.00	現金				ホテルあり
8755	新型アクアハウス	-/h*4747N47	新與且新與市高差町 3-33	10.00	祖全				熱芸角・海水角・全角・綿鯉・水槽飼育器



スクロール関係の「追加情報」特性

- デフォルトの動作を変えたい場合には・・・
- スクロールの動作を指定するため、フォームの「追加情報」特性に、以下を指定します。
 - ScrollVertical=Y/N (縦方向のスクロール指定)
 - ScrollHorizontal=Y/N (横方向のスクロール指定)

ScrollVertical=	Υ	N
フォーム全体	スクロールする	スクロールしない
テーブル内レコード	スクロールしない	スクロールする

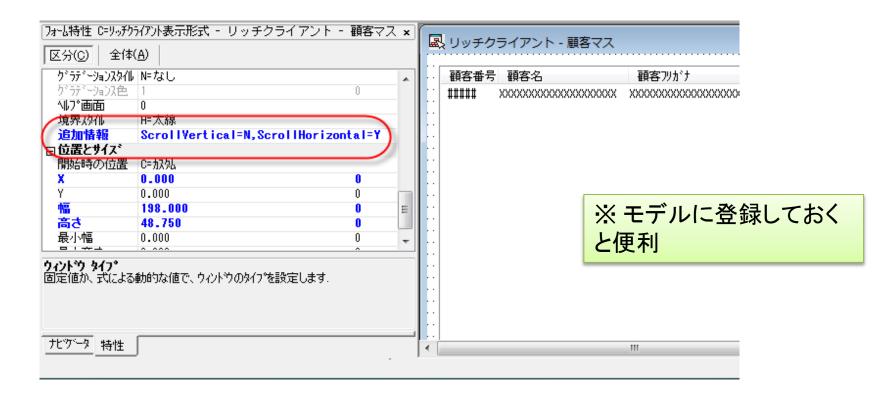
ScrollHorizontal=	Y	N
フォーム全体	スクロールする	スクロールしない
テーブル内レコード	スクロールしない	スクロールしない



テーブルのある場合のスクロール(続)

縦方向にはレコードのスクロールをして、横方向にはフォーム全体のスクロールをする(右側にはみ出したカラムも表示させたい)、という動作を可能にするには、

ScrollVertical=N,ScrollHorizontal=Y と設定しておく。

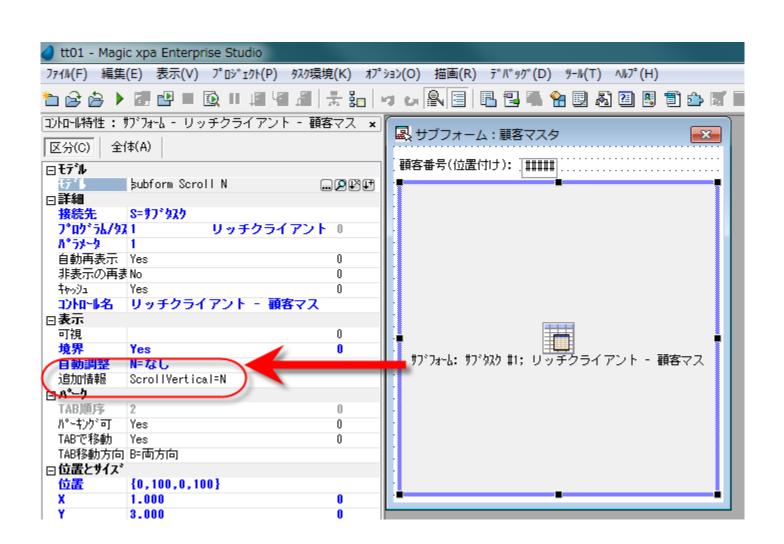


magic サブフォーム内のスクロール

サブフォームの中の子タスクのスクロールについては、

- デフォルトの動作として、フォーム全体がスクロールされます。
 - → サブフォーム中のテーブルのレコードはスクロールしません。
- テーブルレコードをスクロールさせるには、
 - VerticalScroll=N をサブフォームの「追加情報」に指定します。

※ ブラウザコントロールの場合も同様です。





パフォーマンスに関する考慮事項

- Mobile デバイスは、CPUのパワーや、ソフトウェアの実装方法の違いにより、Windows PCよりかなり遅いです。
- Magic のMobile RIAも遅いです。
- ということは、画面設計、ロジックに考慮が必要になります。
- 実験: 精密な測定ではないが、傾向を見てみましょう。
 - ・ ブロックWhileでループ
 ・
 - ≗ コントロール数の影響



ブロック While ループ

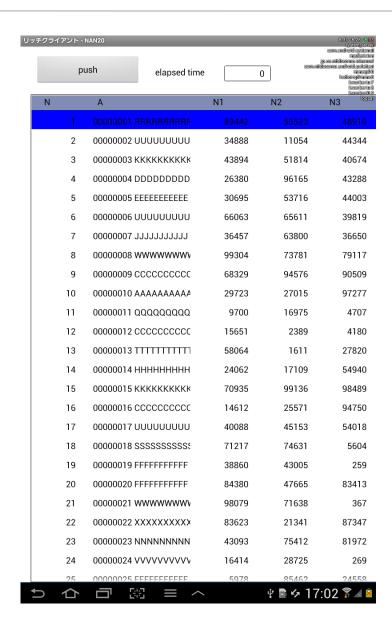
- 純粋なMagicクライアントモジュールの処理速度比較として、ブロック While コマンドで、10,000回のループを回すプログラムを実行し、時間を計測してみます。
- ・ Windows RIA: 0.47 秒 (Core2 Quad Q6600 @2.4GHz → 4年前のCPU)
- Android RIA: 29.7秒 (Galaxy Tab 7.7、APQ8060 1.5GHz(Dual Core) → 最新の CPU)
- ※ その差 60倍以上!
- ※ クアッドコアのCPUだと、何割か早くなるが、劇的な改善は期待できない。

```
ዓスን 245.1 - Record Loop (msec).loop
データビュー ロジック フォーム
     1 回 R=レコート* S=後
              ₩=While 1
        ブロック
                             {LoopCounter() <= Loop Count
        項目更新 V=項目 I
                                          値:
                                               2 num + LoopCounter
                           num
        フ゜ロック N=End
         項目更新 V=項目
                           VAL
                                          値:
                                               3
                                                  num
```



コントロール数の多寡による違い

- 次に、画面上のコントロール数が多くなると、処理速度がどう変わるかを見てみましょう。
- APGでテーブル形式表示プログラムを作成します。
- タスク起動から、全データが表示されるまでの時間を計測します。
- 表示するカラムを追加・削除して、 表示する全コントロール数を変 えて、測定・比較します。



magic 計測結果

- Windows RIA: 1秒以内に表示されます。
- Android RIA:
 - ・ 7カラム×25行 (175コントロール) → 5.3秒
 - 22カラム×25行 (550コントロール) → 16秒
- コントロール数が増えると、非常に遅くなります。(画面上に表示されているか否か、「可視」特性の設定に関わりません)
- スクロールも同様に遅くなります。例: 22×25のテーブルなどでは、ほとんど動かないと言っていいほどの遅さになります。



パフォーマンスの考慮(まとめ)

- Mobile デバイスは、Windows PCに比べて、文字通り桁違いに 非力です。(iOS デバイスでも、同様のレベルです)
- Windows RIAと同じ感覚で多くのコントロールを配置して画面設計していると、非常に遅くなる可能性があります。
- 画面に配置するコントロール数は極力減らすようにしてください。
- ロジックも簡単化してください。
- チャンクサイズは、処理レコード数に影響するので、大きすぎるとやはり遅くなります。必要最小限にしてください。

参考: サーバ側ロジック(バッチタスクなど)は、 サーバ上で実行されるものなので、Mobile RIAでも Windows RIAでも処理時間は同じです。



付録



magic'

Android RIAで使えるコントロールは?

Android RIAでは、以下のコントロールが利用できます。

- ラベル
- エディット
- ・ボタン
- グループ
- ・ ライン
- ・ イメージ
- ブラウザ
- チェックボックス
- コンボボックス
- <u>・</u> テーブル
- ・ カラム
- 🍨 サブフォーム

注意: これ以外のコントロールは、 Studio のフォームエディッタで配置して も、実行時に無視されます。

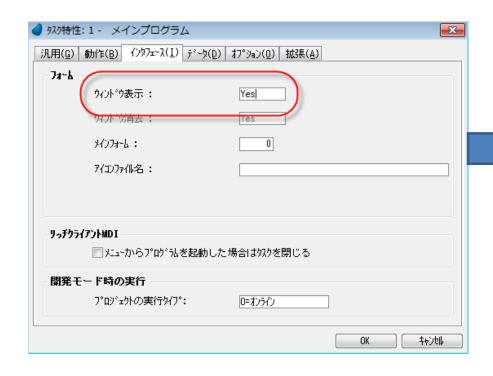


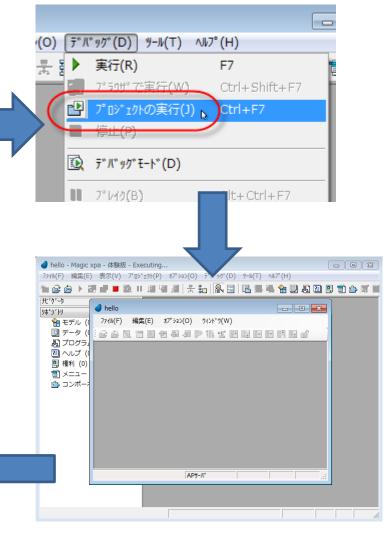
メインプログラムのウィンドウ表示

- Pello Magic実行前に、プロジェクトの変更を行い、メインプログラム → タスク特性 → インターフェース タブ → ウィンドウ表示に No を設定しました。
 これはなぜ必要だったのでしょうか?

- 「ウィンドウ表示」特性を設定しない (Yesのまま) で、「デバッグ
 → プロジェクトの実行(J)」で待機状態にすると、ランタイムの画
 面が表示されます。
- この状態で、Android RIA を起動すると、エラー -138 が出ます。
- エラーコード -138 の意味は、「オンラインのMDIが開いている 場合、リッチクライアントのリクエストは処理できません」です。

※ Mgerror.log にも記録されます。



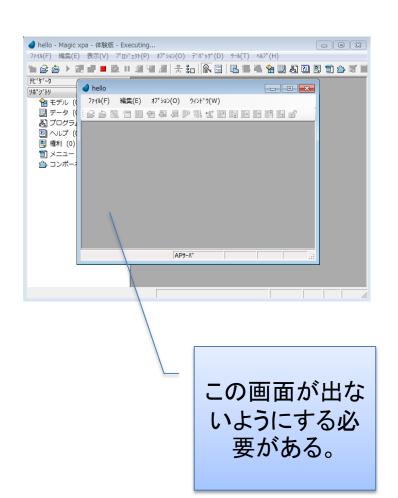


An error occurred (-138). Please contact your system administrator.

OK



待機状態で、オンライン の MDI 画面を開かない ようにしておく必要があり ます。





対応方法 1: バックグラウンドで起動

- 動作環境の「実行モード」を「B= バックグラウンド」にセットして Studio を起動すると、オンライン のMDI画面が表示されないので、 エラーが出なくなります。
- ・ RIAサーバで実際に運用する場合には、この設定で実行します。
- Studioでこの設定を使っていると、 オンラインのプログラムが実行できません。

例: データリポジトリからAPGで表示、プログラムリポジトリからF7でオンラインプログラムの実行などができない。

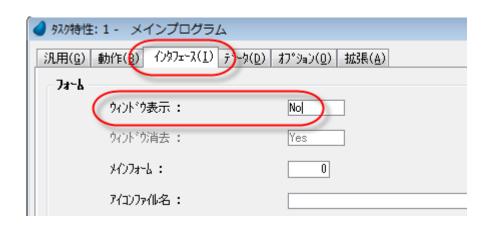
→ 開発時にはちょっと不便な設定かも。





対応方法2: ウィンドウ表示をNoに

- 別の対応方法としては、メインプログラムの「ウィンドウ表示」を No にしておけば、オンラインのMDI画面が表示されなくなります。
- メインプログラムの画面が出なくなるだけで、テーブルリポジトリからのAPGとか、プログラムリポジトリから F7 でのオンライン実行などは、できます。
- Studio で開発時には、こうしておくのが便利でしょう。

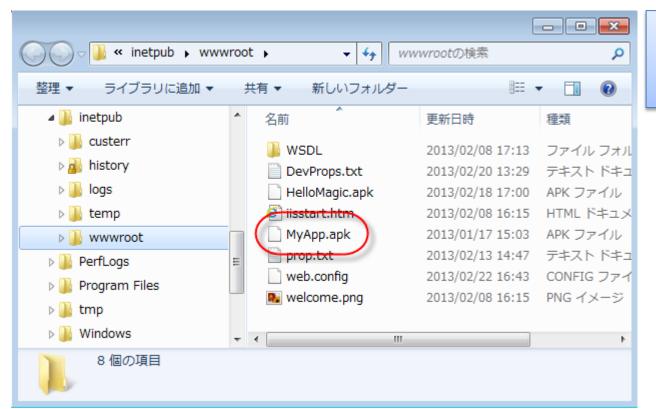


参考: より細かな制御をしたい場合には、RunMode() 関数を使って、「ウィンドウ表示」を式で設定することも考えられます。



.apk ファイルをIISでダウンロード可能にするには?

- apk ファイルをWebサーバの仮想ディレクトリ上に置いて、 Android のWebブラウザからダウンロードしようとすると、エラーが出ます。
- ・ ダウンロード可能にするには、IISでMIME設定が必要です。



MyApp.apk をIIS のルートディレ クトリに置く。

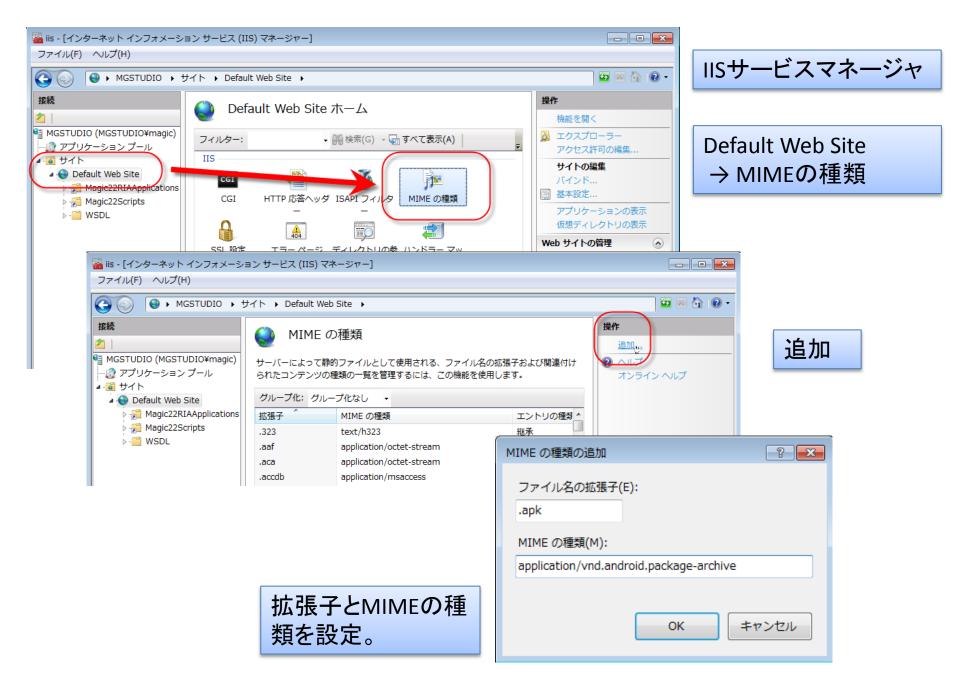
> Androidからダ ウンロードしよう とするとエラー になる。



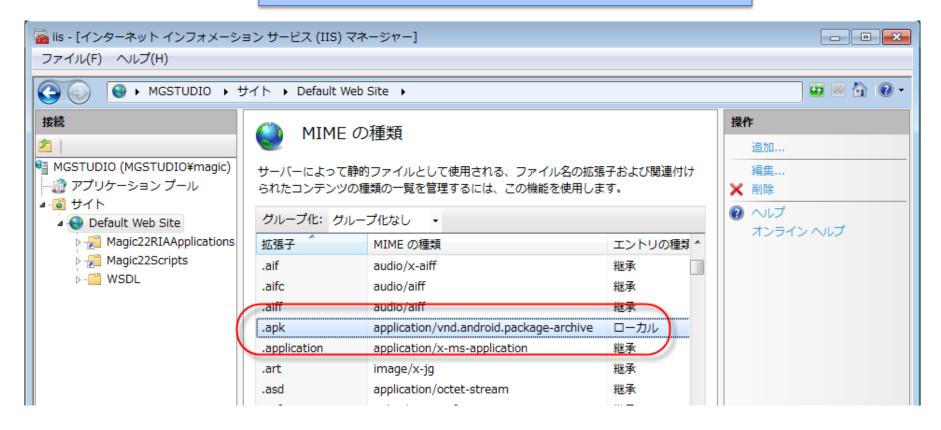
参考: IISのバー ジョンにより、エ ラーメッセージ が違うことがあ ります。 エラーを回避するには、以下のように、IISでMIME設定を行う必要があります。

≗ 拡張子: .apk

MIMEの種類: application/vnd.android.package-archive



このようになれば、ダウンロードできるようになります。





THANK