株式会社ミガロ.

RAD事業部 技術支援課

[Delphi/400] iOS/Androidネイティブアプリケーション入門 ーマルチデバイス開発手法から社内配布

●はじめに

●スマートデバイスアプリケーションの種類 ●ネイティブアプリケーションの開発環境 ●ネイティブアプリケーションの開発手順 ●ネイティブアプリケーションの開発ポイント ●ネイティブアプリケーションの配布・運用 ●まとめ



現在の仕事内容

Delphi/400 や JC/400 の 製品試 験および月 100 件に及ぶ問い合わ せサポートやセミナー講師などを担 当している。

1.はじめに

この数年でスマートデバイスの導入 が個人だけではなく、企業でも急速に進 んできている。実際に 2014 年に総務省 が行ったアンケート調査では、国内企業 のスマートデバイス導入率は28.4%と推 計されており、3割近い企業でスマート デバイスが使用されていることになる (総務省・経済産業省「平成 24 年経済セ ンサス-活動調査))。

2011年の同調査での企業導入率が 10%未満だったことを考えると、飛躍的 に導入数が伸びていることがわかる。

こうしたスマートデバイスを導入し た企業の多くが「業務効率化」を目的と している。これはメールや Web の利用 だけではなく、社内システムの活用が強 く期待されていることである。

Delphi/400のテクニカルサポートに もスマートデバイスのお問い合わせをい ただくことが多くなってきており、これ からの社内システムにはスマートデバイ スアプリケーションが必要とされる機会 が増えてくると実感している。

では、実際にスマートデバイスを導入 している企業では、どういった機種が使 用されているのかというと、その大半が iOS (iPhone、iPad) や Android となっ ている。それ以外の Windows Phone などの機種は、まだ企業導入されていな いのが現状である。

つまり、これから企業のスマートデバ イスでアプリケーションを活用するに は、iOS、Android 機種へ、いかに対応 できるかが重要となってくる。

こうした大きな環境変化の背景もあ り、従来のクライアント / サーバー(以 下、C/S)、Web 型開発に加えて、iOS、 Android のネイティブアプリケーショ ン 開発 に も 対 応 し た Delphi/400 Version XE5 の新機能を紹介したいと 考えた。

本稿では、Delphi/400を使ったスマー トデバイス向けのネイティブアプリケー ション開発について、開発環境や開発手 順、配布・運用のポイントを説明してい きたい。



この章では、スマートデバイス上で動 作するアプリケーションの種類や特徴に ついて説明する。

スマートデバイスアプリケーション は、大きく2つの種類に分類できる。ア プリケーションの種類としては、Web アプリケーションとネイティブアプリ ケーションがある。Delphi/400 ではど ちらのアプリケーションも開発できる が、それぞれのアプリケーションの特徴 を把握してみる。

Webアプリケーション

Web アプリケーションは、PC での使 用と同様に Web ブラウザを使って使用 するアプリケーションである。Webア プリケーションは Web サーバー上で動 作しており、スマートデバイス端末では Web ブラウザを使って利用することが できる。【図1】



そのため、スマートデバイス端末には アプリケーションがインストールされる ことはない。

この特徴のメリットは、「アプリケー ションの配布が不要」という点と、「ブ ラウザに対応していれば機種の制限がな い」という点が挙げられる。逆にデメリッ トとしては、「スマートデバイスのネイ ティブ機能が十分に活用できない」「ネッ トワークに接続していない環境では使用 できない」という点がある。

ネイティブアプリケーション

ネイティブアプリケーションは、ス マートデバイス端末上にインストールし て使用するアプリケーションである。も ちろんネイティブアプリケーションは、 スマートデバイス端末上で動作する。【図 2】

この特徴のメリットとしては「スマー トデバイスのネイティブ機能を100%活 用できる」「ネットワークに接続してい ない環境でも使用できる」という点であ る。またスマートデバイス端末上で直接 実行するため、アプリケーションの動作 レスポンスは、一般的にWebアプリケー ションより優れている。

デメリットとしては、「iOS、Android ごとに別言語の習得・開発が必要となる」 という点が挙げられる。しかし Delphi/400 では、iOS、Androidのネ イティブアプリケーションを Delphi 言 語のみで開発することができる。そのた め、ネイティブアプリケーションの一般 的なデメリットも、Delphi/400 では逆 に長所となっている。【図 3】

これは「マルチデバイス開発」と呼ば れる画期的な開発手法によるものであ る。「マルチデバイス開発」については、 後述の開発環境で詳しく説明する。

それぞれアプリケーションの種類に よって、得手・不得手の部分があるが、 Delphi/400 ではどちらのアプリケー ションも開発でき、用途によって選択す ることができる。

例えば、機能面が優先であればネイ ティブアプリケーション、運用管理の軽 減が優先であれば Web アプリケーショ ンといった選択も可能である。 3.ネイティブアプリケー ションの開発環境

この章では、Delphi/400 のネイティ ブアプリケーション開発環境を詳しく説 明していく。

iOS/Android のネイティブアプリ ケーションは Delphi/400 で開発できる が、対象とするスマートデバイスによっ て必要となる開発環境が違ってくる。そ れぞれの開発で必要となる環境や設定ポ イントを簡単にまとめてみた。対象とす る開発環境部分をご一読いただきたい。

3-1. iOS向け開発環境

Delphi/400 で iOS 向けのアプリケー ションを開発する場合、Windows だけ ではコンパイルやアプリケーションの配 布が行えないため、OSX(Mac)が必 要となる。【図 4】

もちろん Mac 上に仮想環境(Windows) を構築すれば、1 台のマシンで開発環境 を用意することも可能である。

必要となる環境

- Windows端末 (Delphi/400 Version XE5)
- Mac 端末(OSX 10.7 ~ 10.9)
- iOS Developer Program (Xcode、 配布)
- iOS 実機 (iPhone、iPad など iOS 6.0 ~ 7.1)

Mac 環境の構築

iOS の開発環境では、Delphi/400 を インストールしている Windows 端末と は別に Mac 端末が必要になる。Mac 端 末は OSX 10.7Lion 以降をサポートして いる。

Xcode のインストール

Mac 端末には iOS6.0 以降に対応した Xcode のインストールが必要になる。 Xcode は App Store または Apple の Developer ページからダウンロードし てインストールすることができる。【図 5】

iOS Developer Program

iOS アプリケーションは Apple 社の 規約により、iOS へ配布するためには 「iOS Developer Program」に加入する 必要がある。「iOS Developer Program」 は1年間の有償プログラムとなってお り、用途別に2種類用意されている。「iOS Developer Program」と「iOS Developer Enterprise Program」である。

① iOS Developer Program https://developer.apple.com/jp/ programs/ios/

「iOS Developer Program」は、主に App Store 向けのアプリケーションを 配信するためのプログラムになる。Ad Hoc と呼ばれる機能で社内向けにアプ リケーションも配布することはできる が、台数は上限が100台に設定されて おり、端末の事前登録も必要になる。

② iOS Developer Enterprise Program https://developer.apple.com/jp/ programs/ios/enterprise/

「iOS Developer Enterprise Program」は社内専用向けアプリケー ションを配布するためのプログラムにな る。「iOS Developer Program」と違い、 社内向けに配布できる台数に制限がな く、端末の事前登録も必要ない。ただし App Store 向けにアプリケーションを 配信することはできない。

この2種類のプログラムは用途別に 用意されているが、社内用アプリケー ションを開発・運用する場合には、「iOS Developer Enterprise Program」のプ ログラムが目的に合っている。注意点と しては、プログラムの購入から手続きの 完了まで Apple 社の処理に数日かかる ため、開発前に事前に設定しておく必要 がある (本稿執筆時 2014 年 8 月時点の 情報。今後 Apple 社によって変更され ることもあるため、最新情報は Apple 社サイトなどでご確認いただきたい)。

実機の登録

iOS アプリケーションの開発は、Mac 上の iOS シミュレータでも実行できる が、実機の iPhone や iPad とは違う部 分も多い。そのため、実際の開発では実 機でのテストが必須といえる。

実機をテストで使用する iOS は、 Mac 上でキーチェーンアクセスを行っ て登録しておく必要がある。キーチェー ンアクセスは Mac 上の「アプリケーショ



ン | ユーティリティ」メニューから作業 できるので、Apple 社のマニュアルを 参考に登録作業を行う。

PAServer のインストール

Delphi/400 の Windows 開 発 PC か ら、コンパイルしたアプリケーションを 転送したり、デバッグを行うための PAServer (Platform Assistant Server)を Mac にインストールする。 Delphi/400 Version XE5 で は、開発 PC の下記フォルダに Mac 向けのイン ストーラが用意されている。

フォルダパス:

C:\Program Files\Embarcadero\RAD Studio\12.0\PAServer

このフォルダにある「RADPAServer XE5.pkg」を Mac 端末にコピーして、 Mac 上でダブルクリックするとインス トールが行える。【図 6】

インストールが完了すると、メニュー の「アプリケーション」に「RAD PAServer XE5」として登録されている ので、PAServer をダブルクリックして 起動する。【図7】

PAServer が起動するとコンソール画 面で「Enter キーを押す」と表示される ので、[Enter] キーを押してサービス の開始が完了する【図 8】(iOS7.1 使用 の場合は、Hotfix_6 (29795) を適用す る必要がある)。

接続プロファイルの作成

Delphi/400 開発 PC から、Mac に接 続する設定を作成する。Delphi 開発画 面の[ツール¦オプション]からオプショ ン画面を開き、「接続プロファイルマネー ジャ」を選択する。【図 9】

追加ボタンを押すと【図 10】のよう なダイアログが表示されるのでプロファ イル名を設定して「次へ」を押す。プロ ファイル名は任意で設定できるので、分 かりやすい名前(Mac など)にしてお くと使用時に便利である。

次に表示される設定画面で Mac 端末 の接続情報を設定し、接続テストが成功 すれば完了である。接続テストには「接 続テスト」ボタンが用意されている。【図 11】

SDK の取得

Delphi/400上で対象のデバイス OS に合わせた開発を行うために、SDK の 取り込みが必要になる。接続プロファイ ル同様に Delphi 開発画面の [ツール | オプション]からオプション画面を開き、 「SDK マネージャ」を選択する。【図 12】

追加ボタンを押すと【図 13】のダイ アログ画面が表示される。プラット フォームに「iOS デバイス」を選択して、 接続するプロファイルには作成済のプロ ファイルを選択して設定する。最後に接 続先から対象の SDK バージョンが自動 表示されるので、選択して「OK」ボタ ンを押下する。これだけで、自動的に SDK がダウンロードされて組み込みが 完了となる。

3-2. Android向け開発環境

Delphi/400 で Android 向けのアプリ ケーションを開発する場合、Windows 内に全ての開発環境を構築することがで きる。【図 14】

必要となる環境

- Windows 端 末 (Delphi/400 Version XE5)

- Android 実機

(Android 2.3.3 以 降 の ARM7 + NEON 対応デバイス)

開発環境の構築

Androidの開発環境では、iOSと異 なりWindows端末に全て環境を構築で きる。ただし、開発の対象となるAndroid 実機のPC接続用ドライバは事前にイン ストールが必要となる。Androidの機 種によってインストール方法が異なるた め、機種の製造元が提供する方法を確認 してインストールを行う。

SDK の取得

Delphi/400上で対象のデバイス OS に合わせた開発を行うために、Android でも SDK の取り込みが必要になる。 Delphi/400 では、Android SDK 専用 の管理ツールとして「AndroidTools」 が用意されている。

スタートメニューから [Embarcadero RADStudio XE5|Android SDKs] よ り「Android Tools」でツールを起動す

ることができる。【図 15】

起動すると AndroidSDKManager の 画面が表示されるので、開発対象の AndroidOS バージョンにチェックをし て、「Install」ボタンを押下する。これ だけで必要な SDK を取り込むことがで きる。Android は iOS に比べて OS の バージョンも非常に多いが、全ての SDK を取り込むとかなりのディスク容 量を使用するので、必要なバージョンだ けを取り込んだほうがよい。【図 16】

3-3. マルチデバイス開発

ここまででiOS、Android それぞれ の開発環境のポイントを説明したが、実 際の開発では、iOS も Android も同じ ようにネイティブアプリケーションのプ ログラムを開発することができる。 Delphi/400 のネイティブアプリケー ションの開発画面は、従来のWindows フォーム設計部にスマートデバイス画面 を表示して開発することができ、このス マートデバイス画面に対して、コンポー ネント配置、コーディングといった従来 のC/S型、Web 型と同じ手法で開発を 行う。【図 17】

スマートデバイス画面は、右上のプロ ジェクトマネージャにおいて、開発対象 となるデバイスが選択できるようになっ ており、対象のデバイス(iOS、 Android、Mac)を指定するだけで、1 つのプログラムからそれぞれのネイティ ブアプリケーションを生成できる。

これは FireMonkey と呼ばれる Delphi の新しいフレームワークを使用してお り、コンパイル先に指定したデバイス向 けのアプリケーションに自動で対応でき る。デバイスの違いは FireMonkey フ レームワークが吸収してくれるため、開 発者は Delphi 言語で開発するだけで、 Windows のみならず、スマートデバイ スにも対応できる。【図 18】

この開発手法は「マルチデバイス開発」 と呼ばれ、WindowsやiOS、Android など複数デバイスのアプリケーション開 発が必要となる場合に、Delphi/400な らではの高い生産性を発揮することがで きる。

	ě	Pages						
	12	Pages Photo Booth						
	ő	OuickTime P	aver					
		RAD PAServe	r XES					
- 1	0	Safari						
	0	Time Machin	e					
図8	PASer	verのコンソール	,					
	000	😭 s	upport — paserver ·	— paserver — 8	0×24	R _M		
	Last log	in: Tue Aug 12 0 :~ support\$ /App	08:51:10 on ttys00 olications/RAD\ PA	Ø Server∖ XE5.ap	p/Contents/MacOS,	/paserver		
	; exit; Platform	Assistant Serve	er Version 4.2.0.	05 alogica Tee				
	copyrigh 接続プロ	ファイル パフロ・	- K ノパフロードが	かい場合けただ	Fater キーた切せ			
	1女 枕 ノ 凵	279 N AXY		ゆいる 古 は /こ /こ	inter ャーを押り	-		
<u>×</u> 9	接続プ	ロファイルマネ-	-ジャ					
×9	接続プ	ロファイルマネ-	ージャ					
×9	接続プ Q オブション	ロファイルマネ-	-ジャ					X
¥9	接続プ ③ オブション ・ ^{現地プ} ーオブ	ロファイルマネ-	-ジャ ブロファイル(2):	707+1,4,71 7321-74	₩₹			×
図 9	接続プ ③ オブション ・ 環境オブ ・ ジッリ ー オガ ・ ジッリ ー コノオ	ロファイルマネ-	-ジャ フロファイルႲ):	ブロファイル ブ ブラットフォ オス	以存ィ ーム的: 」			×
<u>ञ</u> 9	接続プ (3 オブション 「 ^{環境オブ} - コオ - コポ - コポ - ネッシー - コ - コポ - マックー - コ - コポ - マックー - コ - コポ - マックー - コ - マック - コ - コ - マック - コ - コ - コ - コ - コ - コ - コ - コ	ロファイルマネー コファイルマネー パレット ・パレット キントッールバー 数 ムーア	-ジャ フロファイル២:	707г/<i>I</i>/7 759173 жд ж-Н	以存す ーム的: トるか: 生気の:			*
図9	接続プ (3 オブション 「環境オブ - オブジー」 - コンオ - フッツー」 - コンオ - フッツー」 - コッサ - マッジー - フッツー - コッサ - マッジー - フッツー - コッサ - マッジー - マッシー - マッシー - マッシー - マッ - マッシー - マッシー - マッシー - マッシー - マッ - マッシー - マッ - マッ	ロファイルマネー 200 200 200 200 200 200 200 20	-ジャ ブロファイルビ):	7077-(1, 7) 75917 8 #2 #2 72917 #2	1)パライ ーム创: ト名句: ード句: ード(1):		194	
<u>9</u>	接続プ (3 オブション 「 ⁽¹⁾ 「 ⁽¹⁾ 」 ⁽¹⁾ ⁽¹	ロファイルマネー コントインスペジタ パレット キャントツールバー 支数 シューア さユーア さユーア さユーア さユーア さユーア さユーア さユーア さユーア さユーア さユーア さユーア さユーア さユーア さユーア シュート インスペジタ ・ パレット ・	-ジャ 70ファイル֎):	ブロファイル ブ ブラットフォ ネス ボート ノマスワ	DK5-1 ームの: トるの: ートの: ートの: ートの:			× • • • •
図9	接続プ ③ オブション ・ 環境オブジーンオ ・ マート ・ マート ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	ロファイルマネー コファイルマネー コクトインスペクタ パレット ト ペントッールパー 数数 ビューア 電す) メニュー コクトのアップガルード カロシアイルマネージャ ジェックト ション・ジェッン・ コクトのアップガルード	-ジャ フロファイル煌):	プロファイル プ プラットフォ ホス ボート ノ(スワ	以存す ームの: ト名か: 手名か: ー トの: ー トの:		18,8	×
図9	接続プ	ロファイルマネー クロンアイルマネー クロン パレット キントッツールパー 整数 シューア ビューア 直す) メニュー マントのアップパレード プロファイルマネージャ デュージャ フローラ オブション イブラリ	-ジャ ブロファイル(2):	7037-(1, 7) 759173 жл ж–н 1(739	1)パライ ーム创: ト名创: 二 一千〇〇:			× • •
¥9	接続プ (3 オブション 「環境オブ 「マッー」 「現境オブ 「フォー 「現境オブ 「フォー 「 「フォー 「 「 「 「 「 「 「 「 「 「 「 「 「	ロファイルマネー パレット エクトインスペクタ パレット ション・インスペクタ パレット ション・ソールバー 支数 ション・ツールバー 支数 ション・フー マカトワップガルード カロンティルマネージャ ジョン・ フロンティルマネージャ ジョン・ パイスマネージャ フローラ パイプション イブラリ ・ 翻訳演み イブライブジリ	-ジャ フロファイルႲ):	プロファイル ク フラットフォ ホス ホート パスワ	以行すイ ームの: 上名の: 二 二 下の: 二 二 下の:			× • • • •
図 9	接続プ (3 オブション 「環境オブジー」」 ・ 環境オブジー」」 ・ マーレオ ・ マーレ ・ マート ・ マーレ ・ マーレ ・ マート ・	ロファイルマネー コン・インスペシタ パレット ・パレット ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	-ジャ フロファイル煌):	プロファイル プ プラットフォ ホス ボート ノ(スワ	以存す ームの: ト名か: 事号の: ードの:		· · · · · · · · · · · · · · · · · · ·	×
翌 9	接続プ 「環境オブション 「環境オブジー」 「現境オブジー」 「日本 「日本 「日本 「日本 「日本 「日本 「日本 」」 」」 「日本 」」 」」 」」 」」 」」 」」 」」 」」 」」 」	ロファイルマネー クロンアイルマネー クロン パレット キントッツールパー 整数 シューア ビューア 直引、シュュー シントのアップガレード プロファイルマネージャ テェントのアップガレード プロファイルマネージャ テェントのアップガレード プローラ はオジョン イブラリ イブーア レード	-ジャ プロファイル(2):	7037-(1, 7) 759173 жл 7,79 7,79	D/57イ ーム创: ト名句: 一下句: 一下句:			× • •
¥9	接続フ フ フ フ フ フ フ フ フ フ フ フ フ フ	ロファイルマネー クロンアイルマネー ショントインスペシタ パレット ショントリールパー 支数 2ユーア ジェントリールパー 支数 シューア ジェントリールパー 支数 シューア ジェントリールパー 支数 シューア ジェントリールパー 支数 シューア ジェント フローフ パイフタイブション イブラリ オブション イブラリ オブション マション 「フラーフ ジョン 「フラーブション マション 「フラーブション マション 「フラーブション マション 「フラーブション マション 「フラーブション マション 「フラーブション マション 「フラーブション 「フラーブション 「フラーブション 「フラーブション 「フラーブション 「フラーブション 「フラーブション 「フラーブション 「フラーブション 「フラーブション 「フラーブション 「フラーブション 「フラーブション 「フラーブション 「フラーブション 「フラーブション	-ジャ 707+1ル(2):	ブロファイル ブ プラットフォ ホス ホート パスワ	以行子イ 一ム①: 上名①: 二 二 上名①: 二 二 二 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 二 〇 〇 〇 二 〇			× • • • •
⊻9	接続プ ・ 環境オブション ・ 環境オブジー ・ フ ・ う ・ フ ・ う ・ う ・ う ・ う ・ う ・ う ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	ロファイルマネー ロファイルマネー ション パレット カー イントッールパー 変数 たューア 電引、メニュー マントのマップパレード フロファイルマネージャ ジョン メニュー マントのマップパレード フロファイルマネージャ ジョン ション なー フリー マントッールパー 変数 たューア モーア モーア モーア モーア モーア モーア モーア モ	-ジャ フロファイル@:	プロファイル フ ル プラットフォ ホート パスワ	D/5-4 - 上砂: ト名切: テろ切: - ド切:			* * *
¥9	接続 フ フ フ フ フ フ フ フ フ フ フ フ フ	ロファイルマネー コン コン コン コン コン コン コン コン コン コン	-ジャ プロファイル(2):	7077{}} 759579 #2 #~H /729	D/57イ ーム创: ト名句: ート句句: ート切:			ч ч т т т т
⊻9	接	ロファイルマネー クロンアイルマネー ション・シスペシタ パレット カー・シントツールパー 支致 2ユーア エットのアップブレード カロファイルマネージャ ブローフ エットのアップブレード カロファイルマネージャ ブローフ オブジョン イブラリ イブラリ イブラリ イブラリ イブラリ オブション ション ション ション ション ション ション ション	-ジャ 70ファイルႲ):	ブロファイル ブ ブラットフォ ホス ポート パスワ 名前の変更 (1)、 雨季	D/5-1 - ムロ: ト名田: - ド田: - 「 王グスポートロ	★-FO····		× • • •
¥9	接続 プション 「 環境オブション 「 「 「 」 」 」 」 」 」 」 」 」 」 」 」 」	ロファイルマネー クロンアイルマネー クロン パレット キントッツールパー 変数 ピューア 直引、シュー アクトロアのプバレード プロファイルマネージャ ジョン プローラ パブジリ・範囲に来み イブジリ・範囲に来み イブジリ・範囲に来み イブジリ・新聞に来み イブジリ・新聞に来み イブジリ・ション ジョン フォーマット ーションツール オブション ト ナ	-ジャ プロファイル(2):	プロファイル ブ ブラットフォ ポート ノロスワ 名前の変更広 雨原	1 1/5・ - ム砂: ト 砂: - ト 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇	*	(

4.ネイティブアプリケー ションの開発手順

この章では、Delphi/400 のネイティ ブアプリケーション開発の流れを、簡単 なアプリケーション開発例を題材に詳し く説明していく。

4-1. 基本的な開発手順

ネイティブアプリケーションの開発 は先の章でも述べた通り、従来のC/S 型、Web型のアプリケーションと同様 の手順で開発できるのでご安心いただき たい。

基本的な開発の流れは次のようにな る。【図 19】

①画面でコンポーネントを配置する
 ②イベントにプログラミングを行う
 ③コンパイルして実行(実行時にデバイスにインストールされる)

今回はスマートデバイスのカメラ機 能を組み込んだ簡単なアプリケーション を例として、開発の手順を説明していく。 【図 20】

ネイティブアプリケーションを新規 に作成する場合には、[ファイル|新規 作成]より「FireMonkey モバイルア プリケーション -Delphi」を選択する。

選択時にダイアログでテンプレート 選択画面が表示される。【図 21】今回は 基本となる「空のアプリケーション」を 選択する。他にもリスト形式画面などの テンプレートが用意されているので、新 規作成時には便利である。

新規作成されたスマートデバイス画 面では、右上のコンボボックスで、デバ イスに合わせた設計画面イメージを選択 することができる。【図 22】

選択肢としてはiPhoneやiPad、 Androidの主要機種(Nexus、Galaxy) が用意されている。またAndroidの場 合、市場の機種が多いため解像度ベース での画面イメージも選択できる。これは 画面表示サイズなどの設計イメージを容 易にするための機能なので、開発したプ ログラムには直接影響しない(Android の見た目で設計してiOSにコンパイル することも可能だが、表示デザインなど が調整しにくい)。

ここからはアプリケーションの開発

内容を説明する。

①画面でコンポーネントを配置する

今回はフォームに ToolBar、Button、 Image、ActionList のコンポーネント を【図 23】のように配置する。

ButtonはStyleLookupプロパティで、 選択しているデバイスのイメージに合わ せた表示スタイルが設定できるように なっているので、カメラのアイコンにな るスタイルを設定しておく。他のコン ポーネントも同様にStyleLookupプロ パティで表示スタイルの設定が可能であ る。別のデバイスを切り替えた場合には、 そのデバイスに用意された同様の表示ス タイルが自動で適用される。

②イベントにプログラミングを行う

アプリケーションの処理は、コンポー ネントのイベントにコーディングするこ とができる。今回は Button にカメラ機 能を使うイベントを設定してコーディン グを行う。

Button の Action プロパティで「標 準アクションの新規追加」から「メディ アライブラリ「TTakePhotoFromCamera Action」を設定する。【図 24】

これだけで、スマートデバイスのカメ ラ撮影機能を Button で利用することが できる。

イベントタブには OnDidFinishTaking というイベントがあるので、このイベン トをダブルクリックしてコーディング処 理部分を作成する。OnDidFinishTaking イベントはカメラ撮影が終わったあとに 実行されるイベント処理である。【図 25】

コーディングする内容は【ソース1】 のように1行だけ記述する。このデバイ スで撮影された画像を画面の Image コ ンポーネントにセットするというプログ ラムコードである。

③コンパイルして実行

ここまでの作業でネイティブアプリ ケーションのプログラム自体は完成して いる。最後にコンパイルを行ってアプリ ケーションの動作を確認する。

プロジェクトマネージャ画面にコン パイル先のデバイスが選択できるように なっている。【図 26】

選択できるのは Android、iOS シミュ

レータ、iOS デバイスである。それぞれ 「ターゲット」という部分に接続してい るデバイスの端末名が表示されるので、 端末名をダブルクリックして選択する。

今回は iOS のデバイスを選択してコ ンパイル実行(メニューの実行、または F9)でアプリケーションを生成して実 行してみる。実行するとコンパイル完了 あとに、iOS の実機上でアプリケーショ ンがインストールされ、作成したカメラ アプリケーションが起動する。アプリ ケーションのボタンを押すとカメラ機能 が起動し、撮影を行うことができる。撮 影した画像はアプリケーションの画面に セットされる。

これだけでカメラ機能を連携した iOS ネイティブアプリケーションが完成した ことになる。【図 27】

それでは Android ネイティブアプリ ケーションでは、どのように開発するか というと、実は今作成したプログラムの コンパイル先を変更するだけでよいので ある。

プロジェクトマネージャの「ターゲッ ト」に表示される Android 端末を選択 して、コンパイルしてみる。すると、プ ログラムは1行も変えていないので、同 じアプリケーションが Android 上にイ ンストールされて実行される。【図 28】

これが先に説明した Delphi/400 のマ ルチデバイス開発である。

1つのプログラムからコンパイル先の 指定だけで、複数のデバイスに対応でき る。この開発手順を試していただくと、 簡単にスマートデバイス向けのアプリ ケーションが開発できることを実感して いただける。

4-2. IBM i 活用手順

先の例では、ネイティブアプリケー ションの基本的な開発手順を説明してき た。ここからはネイティブアプリケー ションから、Delphi/400の機能を使っ て IBM i へ接続する方法を説明する。

通常、PC から社内のデータベースに 接続する場合は、PC にデータベース接 続用のドライバをインストールしてい る。

しかし、スマートデバイスでは、社内 のデータベースに直接接続することはで きない。これは IBM i に限らず、 Oracle や SQLServer など、どのデー

プロファイル指報 このウィザードは接続プロファイルの作成に役立ちます。プロジェクトをリモートマシンに配置して実行するには、排続プロファイルが必要です。 クロファイルのプロファイルの音振曲 このプロファイルのプロファイル名を指定しブラットフォームを選択してください。 フロファイル名を): フロファイル名を): Mac ブラットフォーム(Δ): OS X ② 選択したブラットフォームのデフォルトとして設定する(D) < < <	₹
といアイリーアイルの小必要です。PAServerのインストールの詳細 このプロファイルのグロファイル名を指定しブラットフォームを選択してください。 クロファイルのグロファイル名を指定しブラットフォームを選択してください。 クロファイル名(2): OS X ② 選択したブラットフォームのデフォルトとして設定する(2) << << <<	
たのプロファイルのプロファイル名を指定しプラットフォームを選択してください。 フロファイル名のピ: 「ロコァイル名のピ: 「ある」 「の「「「「」」」 「の」 「の 「の	
Cのプロファイルのプロファイル名を指定しブラットフォームを選択してください。 フロファイル名(2): Mac フラットフォーム(Δ): OS X ⑦ 選択したブラットフォームのデフォルトとして設定する(2) <<戻る(2) 次へ(2) >> 終了 キャンセル ヘルブ 接続プロファイルの設定	-
プロファイル名(2): Mac プラットフォーム(2): OS X ② 選択したブラットフォームのデフォルトとして設定する(2) <<< <<< (<< 反る(8) 次へ(10)>> 終了 キャンセル ヘルプ	•
Mac ブラットフォーム(Δ): OS X ⑦ 選択したブラットフォームのデフォルトとして設定する(D) <<戻る(B) 次へ(M) >> 終了 キャンセル ヘルブ 接続プロファイルの設定	•
ブラットフォーム(Δ): ○S x ② 選択したブラットフォームのデフォルトとして設定する① <<戻る⑥ 次へ℃ >> 終了 キャンセル ヘルブ 搭続プロファイルの設定	•
OS X ⑦ 選択したブラットフォームのデフォルトとして設定する① <<<戻る(B) 次へ(M)>> 終了 キャンセル ヘルブ 接続プロファイルの設定	•
✓ 選択したブラットフォームのデフォルトとして設定する① <<< 戻る⑧ 次へ№ >> 終了 キャンセル ヘルブ 接続プロファイルの設定	
ぼうのです。 マングレース (く見る)の (次への) (次への) (次への) (次への) (次への) (なんの) (xんの) (xんの) (xん) (xん) (xん) (xん) (xん) (xん) (xん) (xん	
<< 戻るB 次へN >> 終了 キャンセル ヘルプ 接続プロファイルの設定	
<<戻る(B) 次へ(M)>> 終了 キャンセル ヘルプ 接続プロファイルの設定	
接続プロファイルの設定	
接続プロファイルの設定	
接続プロファイルの設定	
接続プロファイルの設定	
リモートマシン名がリモートマシンの IP アドレスのどちらかを指定してください。リモートマシンに PAServer がインス トールされて稼働している場合にのみ、接続プロファイルにより、ターゲットプラットフォームに接続できます。	
PAServerのインストールのIF細	
リモートマシン (IP アドレスまたはマシン名)(R): 999.999.999.999	
ボート番号(D) 54211	
127-FW:	
パスワードの入力	
<<戻る(B) 次へ(N) >> 終了 キャンセル ヘルブ	

 タベースでも同じである。その理由は、 iOS や Android などのデバイス上には、 データベースに接続するためのドライバ がインストールできないからである。そ のため、スマートデバイスのネイティブ アプリケーションから IBM iに接続す る場合には、【図 29】のようにアプリケー ションサーバーを経由した3階層方式の 接続となる。

アプリケーションサーバーには、C/S アプリケーション同様に IBM i に接続 するサーバーアプリケーションが必要に なる。Delphi/400 では、このサーバー アプリケーションにも「DataSnap」と 呼ばれる専用開発機能が用意されてお り、容易に開発が可能である。

サーバーアプリケーションには、 SQLConnection や SQLQuery などの DB コンポーネントを設定したり、処理 関数をプログラミングすることで機能を 実装する。

「DataSnap」を使ったサーバーアプ リケーションの開発方法は、本稿では割 愛させていただくが、『Migaro. Technical Report No.5』に「DataSnap を使用し た3階層アプリケーション構築技法」と いうレポートで詳しくまとめているの で、こちらを参考にしていただきたい。

それでは、この「DataSnap」のサー バーアプリケーションに接続するネイ ティブアプリケーションの開発手順を説 明する。

今回は、IBM i上の得意先マスタ (CUSTOMER ファイル)の一覧を表示 するアプリケーションを例として、開発 手順を確認していく。【図 30】

①画面でコンポーネントを配置する

まずフォームに ToolBar、Label、 Switch、ListView を【図 31】のよう に配置する。Label には"得意先一覧" とタイトル名を設定しておく。

スマートデバイス上に配置する Label では、Label の表示文字が部品以上に長 い場合、"得意先…"というように画面 上で自動省略されてしまう。そのため、 Label の AutoSize プロパティを True に設定しておくことをお勧めする。この 設定をしておくと、表示文字の長さに合 わせて部品のサイズ側が自動調整してく れる。【図 31】

次に IBM iの接続コンポーネントを

配置する。フォームに SQLConnection、 DSProviderConnection、ClientDataSet を【図 32】のように配置する。

それぞれのコンポーネント設定内容 を説明する。SQLConnection コンポー ネントは ConnectionName プロパティ に"DataSnapConnection"を設定する。 【図 33】

そして Params プロパティに、アプ リケーションサーバーの IP アドレスと ポート番号を設定しておく。これにより 「DataSnap」のサーバーアプリケーショ ンに接続することができる。また、スマー トデバイス上ではダイアログが出ないの で、LoginPrompt プロパティ は False に設定しておく必要がある。

次 に、DSProviderConnection コ ン ポーネントを設定する。【図 34】

SQLConnection プロパティには、先 ほど設定した SQLConnection を指定す る。ServerClassName プロパティには、 サーバーアプリケーションで作成してい るクラス名を設定するが、デフォルト名 では"TServerMethodsl"を指定する。

最後に ClientDataSet コンポーネントの設定を行う。【図 35】

RemoteServer プロパティに、先ほど 設定した DSProviderConnection を指 定 す る。 こ の 設 定 を 行 う と、 ProviderName プロパティにサーバー アプリケーションの Provider が自動で 表示 されるので選択する。最後に CommandText プロパティに IBM iに アクセスしたい SQL 内容をセットする。 今回は得意先マスタ(CUSTOMER ファイル)にアクセスするため、次のよ うな SQL を記述する。

"SELECT * FROM CUSTOMER"

ここまでの設定で IBM i へ接続して、 得意先マスタのデータにアクセスするこ とができる。ClientDataSet をダブルク リックして、リストを右クリックから「す べてのフィールドの追加」を選択する。 これで IBM i から得意先マスタの項目 を取り込むことができる。

次に、アクセスしたデータをアプリ ケーションの画面上に表示する部分を作 成する。C/S型、Web型アプリケーショ ンの場合、DBGrid などのコンポーネン トが便利だが、残念ながら FireMonkey のフレームワークには同じコンポーネン トが存在しない。そのため、今回は LiveBindingというビジュアルリンク機 能を使用する。LiveBindingは取得し たデータを画面上のコンポーネントに自 動でリンクしてくれる便利な機能であ る。これは簡易作成機能なので、もちろ んプログラミングでデータをセットして も問題ない。

画面に配置した ListView にデータを 表示するには、フォームを右クリックか ら「ビジュアルにバインド」を選択する。 【図 36】

開発画面下部にビジュアルバイン ディングの設計画面が起動されるので、 この画面で項目のリンク設定を行う。

リンクの方法は簡単である。データ項 目と表示したいコンポーネントの項目を ドラッグ&ドロップするだけで、感覚的 にリンクを設定できる。【図 37】

設定ができたら ClientDataSet の Active プロパティを True に設定する と、表示結果を確認できる。【図 38】

このように、実際に開発設計画面上に IBM iのデータがリンク表示されるの で、プログラムをコンパイルしなくとも、 画面を細かく調整することができる。

②イベントにプログラミングを行う

アプリケーションで得意先マスタの データを表示 / 非表示ができるように Switch コンポーネントの OnSwitch イ ベントにコーディングを行う。プログラ ムを【ソース 2】のように1 行だけ記述 する。

これでスィッチの ON/OFF によっ て、データの表示 / 非表示を操作できる。 データの表示制御は LiveBinding が自 動で行ってくれるので細かいプログラミ ング制御は必要ない。

③コンパイルして実行

ここまでの作業でプログラムは完成 である。プロジェクトマネージャで対象 のデバイスを選択して、コンパイルを実 行する。マルチデバイス開発なので、 iOS でも Android でも可能である。【図 39】

このように IBM i のデータを活用す るネイティブアプリケーションも、ほと んどプロパティの設定だけで簡単に開発

図12 SDKの追加	
③ オブション ▲ 第3時オブション → 第3時オブション → SSK /5-プョン(5): SSK プロ//5-(
- 7/9/19/1 7/2/3 - 9/- #//19/ - 2/#-#2/1 9/- #/-	
- 環境実験 - 差分ピューア - マージジューア - 「聞き直す」メニュー	
- プロジェクトのアップグレード - 接換プロフィイルマネージャ - プロビジョニング 4 / フォーム デザイナ	
- デパイスマネージャ - エクスプローラ ▲ Debri オプション 	
- ライブラジ - 新聞を決み - カイコ ネイゴネロ - SOK マネージャ	
- シースオプション - 色 - 表示 - を一切的当て	
→ TVaP/3C → TWR - UveBindings → HTNL オプション	
- トロル、フォーマット - トランスレーション ツール オブション - 色 - フォント - 1850(2)	
CK #+>tth Nh7	
 図13 SDKの選択	
新規 SDK の追加	
プラットフォームの選択(5):	
ios デバイス	
接続するプロファイルの選択(S):	
IOS、(木人ト: 192.168.0.0 、木一ト番号: '64211) ▼	
 - SDK ハージョンの)選択(S):	
- 図 選択した SDK をアクティブにする(M)	
 ОК <i>キャンセル</i> ヘルプ	
図14 Androidアプリケーション開発環境	
 開発 コンパイル インストール・実行	
Android	
Windows	

できる。例えば、この得意先マスタの一 覧のデータをタッチすることで、得意先 の受注一覧を表示するといった機能カス タマイズも、同じような手順で作成でき る。【図 40】

こうしたアプリケーションであれば、 コンポーネントの設定と数行のコーディ ングで開発できてしまう。

5.ネイティブアプリケー ション開発のポイント

前章までは Delphi/400 のネイティブ アプリケーションの開発手順について説 明してきた。この章では、ネイティブア プリケーション開発時にヒントになりそ うなポイントをいくつか補足したい。

5-1. iOSとAndroidの違い

Delphi/400 では、iOS でも Android でも1つのプログラムで開発できるが、 iOS と Android ではデバイスの違いが あるため、設計上でいくつか考慮してお く点がある。

1つはハードウェアキーの違いであ る。Androidには「ホームボタン」や「戻 るボタン」「メニューボタン」が物理的 に存在するが、iOSには「ホームボタン」 しか用意されていない。【図 41】

例えば、iOSで「戻るボタン」が前提 のアプリを作成してしまうと意図した画 面遷移操作が行えなくなってしまう。そ のため、OS・ハードの違いを把握した 画面設計は非常に重要となってくる。

また、デバイスの構造が違うので、ア プリケーション内でファイルを扱う場合 にも考慮が必要である。

例えば、アプリケーションで音声や動 画を流したりする場合には、オーディオ ファイルなどをアプリケーションと一緒 に配布する必要がある。

しかし、当然ながらデバイス上の構造 が違うので、ファイルを保存するための パスも違ってくる。従って、ファイルの パスなどは、iOS/Android ごとに設定 をしておく必要がある。

ファイルの配置は [プロジェクト | 配 置]からデバイスごとに設定できるので、 iOS で あ れ ば ".¥StartUp ¥Documents¥"、Android で あ れ ば "assets¥internal¥" に設定する。【図 42】(アプリケーションの外に配置する 場合はパスも異なる)

こうしたアプリケーション固有の配 置ファイルパスをプログラムで取得する 場合には、コーディングも違ってくる。 配置した Alerm.mp3 というオーディオ ファイルを TMediaPlayer コンポーネ ントに設定する場合であれば、【ソース 3】のように記述することができる。

デバイスごとに異なるプログラム コードは、iOS であれば [\$IFDEF IOS]、Android であれば [\$IFDEF ANDROID] というタグを記述してお けば、特定のデバイス実行時のみ有効な コーディングも可能である。

もちろん配布したファイルを読み込 むだけでなく、C/S型アプリケーショ ンのように設定ファイルをデバイス上に 作成・保持することもできる。またネッ トワークに接続されていない場合に、 ローカル環境で動作するように CSV な どのファイルデータをデバイス内で保存 し、ネットワークにつながってから IBM i にデータを登録するといったこと も実現できる。

5-2. アプリケーションのカスタマイズ 設定

プログラミングとは別に、ネイティブ アプリケーションで設定しておけるカス タマイズ設定も補足しておく。

アイコンのカスタマイズ

例えば、スマートデバイスにインス トールしたアプリケーションのアイコン も設定が可能である。[プロジェクト¦ オプション]から「アプリケーション」 を選択するとデバイスごとにアイコンを 設定できるようになっている。【図 43】

ここで png などの画像ファイルで作 成した任意のアイコンを設定すると、イ ンストール時に自社用のアイコンで登録 することも可能である。アイコンはデバ イスによって解像度がさまざまなので、 対象のデバイスに合わせたサイズのアイ コンを用意する必要がある。

デバイス向きのカスタマイズ

また、同じ [プロジェクト | オプショ ン] から「アプリケーション」の設定画 面で「向き」というタブを選択すると、 デバイス固有の向き設定を固定化するこ ともできる。【図 44】 標準では縦横の画面変更時に表示調 整が行われるが、例えば縦専用で設計し た画面の場合、この設定を行えば横表示 にならないようにアプリケーション画面 を縦固定にすることができる。

セキュリティ権限のカスタマイズ

他にも Android アプリケーションの 機能で、特別な権限が必要な場合は、[プ ロジェクト | オプション] から「使用す る権限」の設定画面で、権限を付与した り、あるいはセキュリティ上で制限した りすることも可能である。設定はチェッ クの ON/OFF だけで、かなり細かい設 定まで行うことができる。【図 45】

5-3. ネイティブ機能の連携例

先に説明した開発手順では、ネイティ ブ機能連携の一例としてカメラ機能の連 携開発を説明したが、他にもさまざまな ネイティブ機能を連携することができる ので、一例を紹介しておきたい。

例えば、カメラ機能を応用して、バー コードを読み取ったり【図 46】、GPS の 位置情報を利用して GoogleMap を利用 することもできる。【図 47】

また、加速度センサーなどを使用すれ ば、デバイスの傾きなどを利用した画面 操作を行うことも可能である。さらに、 音声データを録音・再生したり、先にも 紹介したアイコンに通知を表示すること もできる。【図 48、図 49】

この章で紹介した設定機能やネイ ティブ連携機能は、Web アプリケーショ ンでは実現できない内容も多く、ネイ ティブアプリケーションならではの機能 性、拡張性の高さといえる。

6.ネイティブアプリケー ションの配布・運用

この章では開発したアプリケーショ ンをユーザーのスマートデバイスに、ど のように配布して運用するかについて、 説明する。

6-1.社内公開と一般公開

ネイティブアプリケーションのス マートデバイスの配布には大きく、社内 公開と一般公開の配布方法がある。それ ぞれの特徴は次の通りである。【図 50、 図 51】

5 Android Toolsの起動	
Embarcadero RAD Studio XE5	
(3) Delphi XE5	
RAD Studio Documentation	
RAD Studio XE5	
 RAD Studio コマンドプロンプト 	
● 変更、修復、アンインストール	
Android SDKs	
Android AVD Manager	
Android Tools	
3 Android SDK Manager	
Android SDK Manager Packages Tools	
SDK Path: C:¥Users¥Public¥Documents¥RAD Studio¥12.0¥PlatformSDKs¥adt-bundle-windows-x8	6-20130522¥sdk
Packages	
The API Rev. Status	
▷ 🕼 🖓 🖓 Android 4.4.2 (API 19)	
▷ ♥ L ₂ Android 4.3 (API 18)	
▷ □ □ □ Android 4.2.2 (API 17) ▷ □ □ □ Android 4.1.2 (API 16)	
▷	
▷ □ □ □ Android 4.0 (API 14)	
Cal Android 3.2 (API 13) Cal Android 3.1 (API 12)	
▶	
Show: Vpdates/New VInstalled Obsolete Select New or Updates	Install 6 packages
Sort by: API level Repository Deselect All	Delete 9 packages
	0.04
Done loading packages.	
7 スマートデバイスアプリケーション開発画面	
7 スマートデバイスアプリケーション開発画面	- G *
7 スマートデバイスアプリケーション開発画面 771% 離 聴 10000000000000000000000000000000000	
7 スマートデバイスアプリケーション開発画面 Project - Deolog X85 - Unit ファイル 戦 戦 戦 リファクタリング プロタエ の 予む む ゆ - 日日日日 の の ・ ・ 11 日 スマートデバイス形式開発 (FireMonkeyフレームワーク)	
7 スマートデバイスアプリケーション開発画面 Project - Delphi設計画面上で ひまむ 10 2 - 日日 2010 - 10 - 10 - 10 - 10 - 10 - 10 - 10	Q Max 0 D ・
7 スマートデバイスアプリケーション開発画面 Project - Deelphi設計画面上で フマイル 編集 編集 1977/999/29 709年 スマートデバイス形式開発 (FireMonkeyフレームワーク)	
7 スマートデバイスアプリケーション開発画面 Projecti - Dephr XES - UNIC 2776 略葉 第章 リファクタリング 7055 の 書き も ゆ - 日 日 日 日 日 日 日 日 日 日 日 上 で スマートデバイス形式開発 (FireMonkeyフレームワーク)	
7 スマートデバイスアプリケーション開発画面 77/1 編編編 2000000000000000000000000000000000	
7 スマートデバイスアプリケーション開発画面 Project Deelwarks Unit フマイル 編集 編集 U ファクタリング 705年 Delphi設計画面上で スマートデバイス形式開発 (FireMonkeyフレームワーク)	C # 2 C C # 2 C C # 2 C C # 2 C C # 2 C C # 2 C C # 2 C C C # 2 C C C # 2 C C
7 スマートデバイスアプリケーション開発画面 Project - Delphi設計画面上で ファイル 報知 報 取 リファクタリング プロン Delphi設計画面上で スマートデバイス形式開発 (FireMonkeyフレームワーク)	Comparing Total State Comparing T
7 スマートデバイスアプリケーション開発画面 27-1/2 「「「「「「「」」」」」 27-1/2 「「「」」」 27-1/2 「「」」 27-1/2 「「」」 27-1/2 「「」」 20-2 「」」 20-2 「」 20-2 「 20-2 「」 20-2 「」 20-2 「」 20-2 「」 20-2 「」 20-2 「」 20	
7 スマートデバイスアプリケーション開発画面 Project - Deephi設計画面上で スマートデバイス形式開発 (FireMonkeyフレームワーク)	Compared and a compared and compared and compared and a compared and a compared and a com
7 スマートデバイスアプリケーション開発画面 Projects - expension - e - =	C 000 C 0000 C 000 C 000 C 000 C 000 C 000 C 000
7 スマートデバイスアプリケーション開発画面 Project Delphi設計画面上で スマートデバイス形式開発 Free Project Delphi没計画面上で スマートデバイス形式開発 (FireMonkeyフレームワーク)	

社内公開

社内公開の場合、開発したネイティブ アプリケーションを社内用のWebサー バーに配置して、各スマートデバイスか らインストールする。社内専用のアプリ ケーションであれば、自由に開発、運用 することができるため、ストアなどの審 査も必要としない。アプリケーションの 配布タイミングも制約はないので、自由 にリリースすることができる。

ただしiOS の場合には、開発環境の 章でも説明した通り、iOS Developer Program に加入しなければ配信するこ とができず、また配信台数もプログラム の種類によって制限されるので考慮が必 要である。

一般公開

一般公開の場合、開発したネイティブ アプリケーションを専用ストアから配信 して、各スマートデバイスからインス トールする。専用ストアは iOS であれ ば App Store、Android で あ れ ば GooglePlayStore を利用することにな る。

ストアを経由して配信するため、イン ストールも容易となるが、誰でもインス トールすることができてしまうため、ス トア配信時には審査がある。そのため、 社内利用限定のアプリケーションは配信 することは難しく、またリリース後に変 更があっても、すぐにリリースはできな い。

6-2. ネイティブアプリケーションの 配布ファイル

社内公開で Web サーバーから配信す る場合、アプリケーションファイルを配 置して HTML からリンクすることにな る。

アプリケーションファイルはコンパ イルの際に生成することができる。iOS であれば、ipa、plistファイル、 Android であれば apk ファイルが配布 の対象となるので、これらを Web サー バー上に配置する必要がある。

HTML のリンクの記述内容は難しく はないが、例を記載しておくので参考に していただきたい。【iOS:ソース4, Android:ソース5】 また iOS の場合、Web サーバー上に

配置する plist ファイルは ipa ファイル

への URL リンクパスを含めて作成する 必要がある。メモ帳などで編集できるの で、以下のように作成する。【ソース 6】

6-3. Webサーバーの設定・考慮点

Web サーバーは IIS や Apache など 使用することができ、html は先に説明 した内容を公開すればアプリケーション を配布できる。

ここでも iOS では、1 つ注意点がある。 iOS7.1 からは Apple 社での仕様が大 きく変わり、SSL での配信でなければ、 インストールを行うことができなくなっ ている。つまり、通常の http での配信 を行えないので、https に対応した SSL 配信ができる Web サーバー環境が必要 になってくるので注意していただきたい (iOS7.0 までは http で配信が行える)。

また、「6-2」で説明した配布アプリケー ションファイルの拡張子は、Web サー バーに MIME タイプとして設定を登録 しておく必要がある。IIS の場合は、IIS マネージャを使用して、サーバーの「プ ロパティ」ページで次の MIME タイプ を追加する。Apache の場合は、mime. types に追加する。各拡張子の設定内容 は次の通りである。

ipa ファイル application/octet-stream plist ファイル text/xml apk ファイル application/vnd.android.packagearchive

ここまで設定が一度完了すれば、アプ リケーションを開発するごとにファイル を配置して、リンクの追加だけで配布・ 運用することができる。

7.まとめ

本稿では Delphi/400 の新機能である iOS/Android 向けの開発手順や運用ポ イントを説明してきた。スマートデバイ ス開発と聞くと、敷居が高く感じられる 開発者の方も多いだろうが、Delphi/400 では、従来の C/S、Web 型と同様の手 法でスマートデバイスのネイティブアプ リケーショも開発できる。 冒頭でふれた通り、これからスマート デバイス向けに社内アプリケーションが 必要とされる機会が増えてくる。

しかし、忘れてはならない重要なポイ ントがある。それは、これまで使用して いる Windows の社内システムがスマー トデバイスの社内システムに置き変わる ことは少ないということである。

実際に Windows 開発者に対して実施 されたアンケート調査結果では、モバイ ルアプリケーションはデスクトップアプ リケーションの置き換えにはならないと いう意見が9割を超えていた。【図 52】

これはスマートデバイスと PC は使い 勝手や用途が異なるので、Windowsの 社内システムは今後も必要とされるとい うことを意味している。つまり今後は、 両システムの併用が一般的になるかもし れない。

そうしたとき、今回紹介した Delphi/400のマルチデバイス開発は、 これまでのWindows向けの開発スキル をスマートデバイスでも活かせる技術で あり、社内開発を拡張していくための最 適解の1つといえる。Delphi/400を使っ て、今後の社内システムでスマートデバ イスに対応される場合には、本稿を開発 の足がかりにしていただければ幸いであ る。

Μ

















図39 iOS/Androidアプリの実行







	ン (Android - すべての構成)	
▲・Delphi コンパイラ	ターゲット(1): すべての構成 - Android ブラ	ットフォーム ● 通用(A) 保存(S)
ービントと警告	+ Bluetooth	🗖 false 🖉 🔺
	田 Bluetooth 管理 日	false
A·リソース コンパイラ	Image: MMS メッセージの受信	false
ディレクトリと条件定義	■ NFC ■ STP サービスの使用	Talse
ービルドイベント	■ SMS メッセージの受信	Talse
12月90催吸	 Image: SMS メッセージの書き込み 	false
アブリケーション	■ SMS メッセージの読み取り	Talse
- バージョン情報	王 SMS メッセージの送信	End faile
▲·パッケージ	WakeLock	Taise
二天行時パッケージ	■ WAP PUSH メッセージの受信	🗖 false
▲·フハッカ ↓ ↓…シンボルテーブル	 Wi-Fiマルチキャスト状態の変更 	false
環境ブロック	· Wi-Fi 状態の変更	false
プロビジョニング	田 With 状態へのアクセス 田 おおよその位置情報へのアクセス	Taise
	1 アカウントの取得	T false
	1 アカウントの管理	false
	■ アカウントの認証	☐ false
	アクセシビリティサービスのバインド	and false
	■アラックーション キャッシュのシリア ■アラームの設定	alse
	ヨインターネット	V true
	オンプットメソッドのバインド	false
	せ カメラの使用	
	ヨカレンダーの読みみの	V tue
	ヨキーロックの無効化	false
	団 グローバル検索	🗖 false
バーコードの連携 DB2/400 IBM i で 読み込んだコード でマスタチェック	EA * 2021 バーコード読込 DELPHI/400 R大会社さガロ、 JC/400 R大会社さガロ、 SMARTPADAI R式会社さガロ、 BUSINESS4M R式会社さガロ、 EUSINESS4M R式会社さガロ、	バーコード読取 機能を実行 他のソフトと連携 できるものもあり





ソース4

ダウンロード用HTML例(iOS)

環境の構築・運用が必要。

<h1>iOSダウンロードサイトサンプル </h1>
<form>

アプリケーションダウンロード </br>
</form>

公開には審査が必要。

(自社用アプリの公開は難しい)

ソース5

ダウンロード用HTML例(Android)

<h1>Androidダウンロードサイトサンプル</h1>

<form>

アプリケーションダウンロード</br>

ソース6

ダウンロード用plist**例**

xml vers</th <th>ion="1.0" er</th> <th>ncoding="UT</th> <th>F-8"?></th> <th></th> <th></th>	ion="1.0" er	ncoding="UT	F-8"?>		
DOCTYPI<br <plist td="" versi<=""><td>E plist PUBLI on="1.0"></td><td>C"-//Apple/</td><td>/DTD PLIST 1.</td><td>0//EN" "http</td><td>://www.apple.com/DTDs/PropertyList-1.0.dtd"></td></plist>	E plist PUBLI on="1.0">	C"-//Apple/	/DTD PLIST 1.	0//EN" "http	://www.apple.com/DTDs/PropertyList-1.0.dtd">
<dict></dict>					
	<key>item</key>	s			
	<array></array>				
		<dict></dict>			
			<key>asset</key>	ts	
			<array></array>		
				<dict></dict>	
					<key>kind</key>
					<string>software-package</string>
					<key>url</key>
					<string>https://Webサーバ /Sample.ipa</string>
			<key>meta</key>	adata	
			<dict></dict>		
				<key>bund</key>	dle-identifier
				<string>co</string>	m.hogehoge.hogehoge
				<key>kind</key>	
				<string>so</string>	ftware
				<key>title<</key>	
				<string>Sa</string>	imple

