



**DB2 Everyplace Mobile Application Builder**  
**バージョン 8.1.4**





# DB2 Everyplace Mobile Application Builder バージョン 8.1.4



# 目次

<b>第 1 章 タスク</b> . . . . .	<b>1</b>
Mobile Application Builder 入門 . . . . .	1
Mobile Application Builder メニュー・バー . . . . .	4
Mobile Application Builder ツールバー . . . . .	7
Mobile Application Builder インターフェース・ペ イン . . . . .	10
キーボード・ショートカットおよびアクセラレータ ーの使用 . . . . .	12
MAB バージョン 8.1.4 のセットアップおよび構成 . . . . .	15
MAB バージョン 8.1.4 Palm のセットアップおよ び構成 . . . . .	15
MAB バージョン 8.1.4 Symbian Crystal のセット アップおよび構成 . . . . .	22
MAB バージョン 8.1.4 Symbian UIQ のセットア ップおよび構成 . . . . .	26
MAB バージョン 8.1.4 WinCE のセットアップお よび構成 . . . . .	30
MAB バージョン 8.1.4 Sharp Zaurus のセットア ップおよび構成 . . . . .	35
Mobile Application Builder のアンインストール . . . . .	38
MAB プロジェクト . . . . .	38
ウェルカム・ダイアログからのプロジェクトのオー プン . . . . .	38
新規プロジェクトの作成 . . . . .	39
既存のプロジェクトのオープン . . . . .	40
プロジェクトとアプリケーションの命名規則 . . . . .	41
プロジェクトまたはアプリケーションの名前変更 . . . . .	42
プロジェクト・プロパティの構成 . . . . .	43
プロジェクトへのソース・ファイルの追加 . . . . .	43
プロジェクトへのライブラリー・ファイルの追加 . . . . .	44
プロジェクトの保管とクローズ . . . . .	45
プリファレンスの設定 . . . . .	46
一般プリファレンスの設定 . . . . .	46
ビルド・プリファレンスの設定 . . . . .	47
アプリケーション・プリファレンスの設定 . . . . .	48
スクリプト・プリファレンスの設定 . . . . .	48
印刷プリファレンスの設定 . . . . .	49
前提条件ツールのためのプリファレンスの設定 . . . . .	50
アプリケーション ID . . . . .	50
アプリケーション ID の登録 . . . . .	51
アプリケーション ID の設定または変更 . . . . .	51
表およびデータ・ソース . . . . .	52
表定義ファイルの作成およびインポート . . . . .	52
表定義ファイルの更新 . . . . .	53
表定義ファイルの削除 . . . . .	54
リスト制御のデータ・ソースの定義 . . . . .	54
フォーム . . . . .	56
新規フォームの作成 . . . . .	57
アプリケーション初期フォームの設定 . . . . .	58
フォームのズーム率の変更 . . . . .	58
フォームのサイズ変更 . . . . .	59
フォームの削除 . . . . .	59
制御 . . . . .	60
フォームへの制御の追加 . . . . .	60
フォームからの制御の削除 . . . . .	61
制御の位置合わせ . . . . .	61
制御の移動 . . . . .	62
制御のサイズ変更 . . . . .	63
制御の再配列 . . . . .	63
制御プロパティの構成 . . . . .	64
デバイス制御プロパティの構成 . . . . .	64
制御のフォントの変更 . . . . .	65
制御の色の変更 . . . . .	66
Palm 5 ウェイ・ナビゲーションを使用可能にす る . . . . .	67
ポップアップ・トリガーの使用 . . . . .	68
ラジオ・ボタンの作成 . . . . .	69
リスト内でのフィールドの表示順序の定義 . . . . .	70
固有値とフィールド制御との関連付け . . . . .	71
プロジェクト・リソース . . . . .	72
新規アラートの作成 . . . . .	72
ビットマップの作成および表示 . . . . .	75
新規アイコンの作成 . . . . .	77
新規メニュー項目の作成 . . . . .	79
新規メニュー・バーの作成 . . . . .	80
新規ストリングの作成 . . . . .	81
イベント、アクション、およびターゲット . . . . .	82
制御イベントの構成 . . . . .	83
制御イベントの再配列 . . . . .	84
制御へのデータベース・プリファレンス・ダイア ログを表示するイベントの追加 . . . . .	85
「レコードの削除」アクションへの物理削除オペ ションの追加 . . . . .	86
Mobile Application Builder 中のスクリプト . . . . .	87
Java スクリプトを使用して MAB 生成コードに アクセスする . . . . .	87
グローバル・スクリプトの作成 . . . . .	89
グローバル変数の定義 . . . . .	90
グローバル・スクリプトの編集または削除 . . . . .	91
イベント・スクリプトの作成 . . . . .	91
イベント・スクリプトの編集または削除 . . . . .	92
イベント・スクリプトの名前変更 . . . . .	93
スクリプト・ウィンドウへのヘルパー機能の追加 . . . . .	94
スクリプト・ウィンドウへの制御の追加 . . . . .	95
フォーム・イベントの SQL を変更 . . . . .	95
フォーム・イベントの SQL を削除 . . . . .	96
ホスト変数の使用 . . . . .	97
ホスト変数の定義 . . . . .	97
フィールド制御の「ホスト変数の更新」プロパテ ィーの設定 . . . . .	98
リスト制御または選択リスト制御の「ホスト変数 の更新」プロパティの設定 . . . . .	99

フォームの選択基準の設定	99
リスト制御または選択リスト制御の選択基準の設定	100
アプリケーションへの印刷機能の追加	101
PalmPrint のインストール	101
フォームへの印刷ボタンの追加	101
フォームへの印刷メニュー項目の追加	102
アプリケーションへのバーコード・スキャン機能の追加	103
フォームへのソフト・スキャン・バーコード・スキャン機能の追加	104
フォームへのハード・スキャン・バーコード・スキャン機能の追加	106
アプリケーションへの同期機能の追加	107
Mobile Application Builder アプリケーションのビルド	108
プリプロセッサ・ステートメントの定義	108
アプリケーションのビルド	109
メッセージ・ログの表示およびクリア	109
メッセージ・ログの保管	110
コード生成/ビルド・ファイル記述	110
Mobile Application Builder アプリケーションのテスト	113
表の作成および移植	114
Palm アプリケーションのテスト	116
Symbian Crystal アプリケーションのテスト	119
Symbian UIQ アプリケーションのテスト	121
WinCE アプリケーションのテスト	123
Sharp Zaurus アプリケーションのテスト	125
<b>第 2 章 概念/参照</b>	<b>127</b>
パレット制御アイコン	127
プロジェクト・プロパティ	128
プロパティの制御	128
フォーム・ビットマップ・プロパティ	129
blob プロパティ	130
ボタン・プロパティ	131
チェック・ボックス・プロパティ	133
選択リスト・プロパティ	134
コマンド・ボタン配列 (CBA) プロパティ	135
フィールド・プロパティ	136
Graffiti シフト・プロパティ	138
Ink プロパティ	139
ラベル・プロパティ	140
リスト・プロパティ	140
ポップアップ・トリガー・プロパティ	142
プッシュボタン・プロパティ	143
繰り返しボタン・プロパティ	144

スクロール・バー・プロパティ	146
選択トリガー・プロパティ	147
リソース・プロパティ	148
フォーム・プロパティ	148
アラート・プロパティ	149
ストリング・プロパティ	150
メニュー・プロパティ	151
メニュー・バー・プロパティ	151
アイコン・プロパティ	152
ビットマップ・プロパティ	153
イベントの説明	153
アクションの説明	155
スクリプト記述サポート	157
Java コード・プラットフォームのためのスクリプト記述サポート	157
Palm C コード・プラットフォームのためのスクリプト記述サポート	158
印刷サポート	158
印刷 API	159
バーコード・スキャン API	162
Palm アプリケーションでサポートされる 1 次元バーコード・タイプ	169
WinCE アプリケーションでサポートされるバーコード・タイプ	170
トラブルシューティング	170
他の情報ソースへのリンク	175

### 第 3 章 例 **177**

DB2 Everyplace のサンプル・アプリケーション	177
CREATE TABLE ステートメントの例	179
データベース照会の定義	179
巡回医療サービス・サンプル・アプリケーションのオープン	180
ユーザー・アプリケーションでデータベース照会を定義するためのガイドライン	180
複数のパラメーターを使用する照会の生成	181
サンプル照会 — すべてのレコードの選択	181
サンプル照会 — 特定のレコードの選択	182
サンプル照会 — 2 つの表の結合	182
サンプル照会 — 列の値が一致するレコードの選択	183
スクリプト内での DB2 CLI 関数の使用	184
変数のメモリの解放	187

### 索引 **189**

### 特記事項 **193**

商標	195
----	-----

---

## 第 1 章 タスク

本節では、Mobile Application Builder のタスク情報を提供しています。

---

### Mobile Application Builder 入門

#### DB2 Everyplace とは何か？

DB2 Everyplace は IBM のモバイル・コンピューティングのためのソリューションの一部です。DB2 Everyplace を使用すると、モバイル・プロフェッショナル (販売員、検査員、監査員、現場の技術者、医者、不動産業者、保険査定人など) は、オフィスから離れた場所で必要な重要データを扱うことができます。



組織において、DB2 の企業データをモバイル・デバイスに送信することができます。DB2 Everyplace を使用して、ご使用のモバイル・デバイス上のデータベースにアクセスし、更新を行い、企業内の他の DB2 データベースを持つデスクトップと同期を取ることができます。

#### Mobile Application Builder

DB2 Everyplace はモバイル・デバイス上のリレーショナル・データベースです。アプリケーションを使用して、モバイル・デバイス上のデータにアクセスすることができます。DB2 Everyplace Mobile Application Builder は、オーダーメイドのモバイル・アプリケーションを容易に生成するためのビジュアル・オーサリング・ツールです。Mobile Application Builder は、アプリケーション開発者を強力にサポートします。

---

#### タスク・リスト:

##### 基本タスク

- 新規プロジェクトの作成
- プロジェクト・プリファレンスの設定
- アプリケーションのビルド

- アプリケーションのテスト

#### 拡張タスク

- アプリケーションへの印刷機能の追加
- アプリケーションへのバーコード・スキャン機能の追加
- アプリケーションへの同期機能の追加
- Mobile Application Builder 中のスクリプト

この他にも多くのタスクを Mobile Application Builder で実行することができます。タスクのリストについては、ブラウザの左側に表示されている、このヘルプの目次を参照してください。

---

### Mobile Application Builder のインターフェース:

Mobile Application Builder には、次のエレメントがあります。

#### メニュー・バー

メニュー・バーを使用して、Mobile Application Builder のオブジェクトで作業したり、オンライン・ヘルプにアクセスしたりします。詳しくは、4 ページの『Mobile Application Builder メニュー・バー』を参照してください。

#### Mobile Application Builder プロジェクト・ツールバー



「プロジェクト」ペインの上にある「プロジェクト」ツールバーのアイコンを使用して、新規プロジェクトを作成したり、既存のプロジェクトをオープンしたり、現行のプロジェクトを保管したりします。これらの機能は、「ファイル」メニューでも選択できます。詳しくは、7 ページの『プロジェクト・ツールバー・オプション』を参照してください。

#### Mobile Application Builder 編集ツールバー



「プロジェクト」ペインの上にある「編集」ツールバーのアイコンを使用して、標準の切り取り、コピー、および貼り付けのアクションを実行します。これらの機能は、「編集」メニューでも選択できます。詳しくは、8 ページの『編集ツールバー・オプション』を参照してください。

#### Mobile Application Builder 制御順序ツールバー



「プロジェクト」ペインの上にある「制御順序」アイコンを使用して、選択した制御をフォームの前または後ろへ移動します。これらの機能は、「フォーマット」メニューでも選択できます。詳しくは、8 ページの『制御順序ツールバーのオプション』を参照してください。

## Mobile Application Builder 制御配置ツールバー



「フォーム」ペインの上にある「制御配置」ツールバーのアイコンを使用して、選択した制御を位置合わせします。これらの機能は、「フォーマット」メニューでも選択できます。詳しくは、9ページの『制御配置ツールバー・オプション』を参照してください。

## Mobile Application Builder ビルド・ツールバー



「フォーム」ペインの上にある「ビルドおよびテスト」ツールバーのアイコンを使用して、アプリケーションのビルドおよびテストを行います。これらの機能は、「ビルド」メニューでも選択できます。詳しくは、9ページの『ビルド・ツールバー』を参照してください。

### 「プロジェクト」ペイン

Mobile Application Builder インターフェースの左側にある「プロジェクト」ペインを使用して、プロジェクト・リソースを表示したり、プロジェクト・リソースで作業を行ったりします。詳しくは、10ページの『「プロジェクト」ペイン・オプション』を参照してください。

### 「プロパティおよびイベント」ペイン

Mobile Application Builder インターフェースの下部にある「プロパティおよびイベント」ペインを使用して、選択した制御のプロパティやイベントを選択したり、編集したりします。「プロパティ」ページで、制御の概観、および制御の振る舞いを決定します。「イベント」ページで、制御が特定のアクションにどのように反応するかを決定します。各制御に、異なるアクションを取らせることができます。詳しくは、11ページの『「プロパティおよびイベント」ペイン・オプション』を参照してください。

### 「フォーム」ペイン

「フォーム」ペインでは、現行のフォームを表示します。パレットから制御を選択し、その制御を置く場所をフォーム上でクリックすることができます。詳しくは、11ページの『「フォーム」ペイン・オプション』を参照してください。

## 制御パレット



制御パレットでは、指定されたデバイスで使用可能なすべてのフォーム制御を表示します。使用可能な制御は、ターゲット・プラットフォームによって異なります。詳しくは、12 ページの『制御パレット・オプション』を参照してください。

---

## Mobile Application Builder メニュー・バー

Mobile Application Builder メニュー・バーには、次のメニューが含まれています。

### ファイル

このメニューを使用して、Mobile Application Builder プロジェクトで作業を行ったり、Mobile Application Builder を終了したりします。このメニューの中のいくつかの機能は、「プロジェクト」ツールバーの中のアイコンをクリックしても実行できます。詳しくは、『ファイル・メニュー・オプション』を参照してください。

### 編集

このメニューを使用して、標準の切り取り、コピー、および貼り付けの機能を実行します。このメニューの中のいくつかの機能は、「編集」ツールバーの中のアイコンをクリックしても実行できます。詳しくは、『編集メニュー・オプション』を参照してください。

### 選択

このメニューを使用して、「プロジェクト」ペインまたは「フォーム」ペインで選択したオブジェクトで有効なアクションを表示したり、選択したりします。詳しくは、『選択メニュー・オプション』を参照してください。

## フォーマット

このメニューを使用して、フォーム上で制御を位置合わせしたり、順序付けたりします。このメニューの中のいくつかの機能は、「制御配置」ツールバーおよび「制御順序」ツールバーの中のアイコンをクリックしても実行できます。詳しくは、『フォーマット・メニュー・オプション』を参照してください。

**ビルド** このメニューを使用して、アプリケーションのビルドおよびテストを行います。このメニューの中の機能は、「ビルド」ツールバーの中のアイコンをクリックしても実行できます。詳しくは、『ビルド・メニュー・オプション』を参照してください。

## ウィンドウ

このメニューを使用して、オープンしているウィンドウの間での切り替えを行います。

**ヘルプ** このメニューを使用して、最新のログ・ファイルを表示したり、製品のドキュメンテーションを表示したり、あるいはご使用の Mobile Application Builder のバージョンに関する情報を表示したりします。詳しくは、『ヘルプ・メニュー・オプション』を参照してください。

## ファイル・メニュー・オプション

「ファイル」メニューを使用して、Mobile Application Builder プロジェクトで作業を行ったり、Mobile Application Builder を終了したりします。このメニューの中のいくつかの機能は、「プロジェクト」ツールバーの中のアイコンをクリックしても実行できます。下記にリストするメニュー項目に加えて、「ファイル」メニューから、プロジェクト名でリストされている最近使用したプロジェクトをオープンすることができます。

このメニューから、次のメニュー項目を選択することができます。

### 新規プロジェクト

新規の Mobile Application Builder プロジェクトをオープンします。

### プロジェクトのオープン

既存の Mobile Application Builder プロジェクトをオープンします。

### プロジェクトのクローズ

現行の Mobile Application Builder プロジェクトをクローズします。

### プロジェクトの保管

現行の Mobile Application Builder プロジェクトをクローズせずに保管します。

### プロジェクトの別名保管

現行の Mobile Application Builder プロジェクトを新しい名前で保管します。

### プリファレンス

プロジェクトおよびアプリケーションのプリファレンスを設定するためのウィンドウを開きます。

**終了** Mobile Application Builder をクローズします。

## 選択メニュー・オプション

「**選択**」メニューを使用して、「プロジェクト」ペインまたは「フォーム」ペインで選択したオブジェクトで有効なアクションを表示したり、選択したりします。

「プロジェクト」ペインまたは「フォーム」ペインでオブジェクトが選択されている場合、そのオブジェクトに関するアクションが「**選択**」メニューに含まれています。

オブジェクトを選択するには、オブジェクトで左マウス・ボタンをクリックします。

オブジェクト上で直接右マウス・ボタンをクリックしても、オブジェクトに関するアクションを表示できます。有効なアクションのリストを含むポップアップ・メニューがオープンします。

## 編集メニュー・オプション

「**編集**」メニューを使用して、選択したオブジェクトで汎用編集コマンド (切り取り、コピー、貼り付け) を実行します。

このメニューから、次のメニュー項目を選択することができます。

### 切り取り

選択したオブジェクトを切り取り、それをクリップボードに入れます。

**コピー** クリップボードの中のオブジェクトを、選択した場所にコピーします。

### 貼り付け

クリップボードの中のオブジェクトを、選択した場所に貼り付けます。

**削除** 選択したオブジェクトを削除します。

これらの機能の中のいくつかは、「編集」ツールバーからも実行できます。

## ビルド・メニュー・オプション

このメニューを使用して、アプリケーションのビルドおよびテストを行います。

このメニューから、次のメニュー項目を選択することができます。

**ビルド** アプリケーションをビルドします。

**テスト** アプリケーションをテストします。アプリケーションをテストする前に、そのアプリケーションを正常にビルドしておく必要があります。

これらの機能は、「ビルド」ツールバーからも実行できます。

## フォーマット・メニュー・オプション

このメニューを使用して、フォーム上で制御を位置合わせしたり、順序付けたりします。

このメニューから、次のメニュー項目を選択することができます。

### 位置合わせ

選択した制御を位置合わせします。

**順序** 選択した制御をフォームの前または後ろへ移動します。

これらの機能は、「ビルド」ツールバーからも実行できます。

### ヘルプ・メニュー・オプション

「ヘルプ」メニューを使用して、メッセージ・ログを表示したり、オンライン・ヘルプを表示したり、DB2 に関する情報を表示したりします。

#### ログ・ファイルの表示

現行セッションの間に実行された各ビルドに関する情報を含むメッセージ・ログをオープンします。

#### ドキュメンテーション

Mobile Application Builder のオンライン・ヘルプを表示します。

#### 製品情報

製品に関する情報を表示します。

## Mobile Application Builder ツールバー

Mobile Application Builder インターフェースには、以下のツールバーがあります。

### プロジェクト

「プロジェクト」ペインの上にある「プロジェクト」ツールバーのアイコンを使用して、新規プロジェクトを作成したり、既存のプロジェクトをオープンしたり、現行のプロジェクトを保管したりします。詳しくは、『プロジェクト・ツールバー・オプション』を参照してください。

### 編集

「プロジェクト」ペインの上にある「編集」ツールバーのアイコンを使用して、標準の切り取り、コピー、および貼り付けのアクションを実行します。詳しくは、8 ページの『編集ツールバー・オプション』を参照してください。

### 制御順序

「フォーム」ペインの上にある「制御順序」ツールバーのアイコンを使用して、選択した制御をフォームの前または後ろへ移動します。詳しくは、8 ページの『制御順序ツールバーのオプション』を参照してください。

### 制御配置

「フォーム」ペインの上にある「制御配置」ツールバーのアイコンを使用して、選択した制御を位置合わせします。詳しくは、9 ページの『制御配置ツールバー・オプション』を参照してください。

### ビルド

「ビルド」ツールバーを使用してアプリケーションのビルドおよびテストを行います。詳しくは、9 ページの『ビルド・ツールバー』を参照してください。

## プロジェクト・ツールバー・オプション



「プロジェクト」ペインの上にある「プロジェクト」ツールバーのアイコンを使用して、新規プロジェクトを作成したり、既存のプロジェクトをオープンしたり、現行のプロジェクトを保管したりします。



### 新規プロジェクト

新規プロジェクトを開始するためのオプションを指定できる、「新規プロジェクトの作成」ウィンドウを開きます。



### プロジェクトのオープン

既存のプロジェクトを選択してオープンできる、「プロジェクトのオープン」ウィンドウを開きます。



### プロジェクトの保管

現行のプロジェクトをクローズせずに保管します。

## 編集ツールバー・オプション



「プロジェクト」ペインの上にある「編集」ツールバーのアイコンを使用して、標準の切り取り、コピー、および貼り付けのアクションを実行します。

ツールバーには次のようなアイコンがあります。



### 切り取り

選択したオブジェクトを切り取り、それをクリップボードに入れます。



### コピー

クリップボードの中のオブジェクトを、選択した場所にコピーします。



### 貼り付け

クリップボードの中のオブジェクトを、選択した場所に貼り付けます。

## 制御順序ツールバーのオプション



「フォーム」ペインの上にある「制御順序」ツールバーのアイコンを使用して、選択した制御をフォームの前または後ろへ移動します。

ツールバーには次のようなアイコンがあります。



### 前へ移動

選択した制御をフォームの前へ移動します。



### 後ろへ移動

選択した制御をフォームの後ろへ移動します。

## 制御配置ツールバー・オプション



「フォーム」ペインの上にある「制御配置」ツールバーのアイコンを使用して、選択した制御を位置合わせします。

ツールバーには次のようなアイコンがあります。



**左方に位置合わせ**

選択された制御を、選択されたアンカー制御の左端に位置合わせします。



**中央に位置合わせ**

選択された制御を、選択されたアンカー制御の垂直軸に沿って位置合わせします。



**右方に位置合わせ**

選択された制御を、選択されたアンカー制御の右端に位置合わせします。



**トップに位置合わせ**

選択された制御を、選択されたアンカー制御の上端に位置合わせします。



**中間に位置合わせ**

選択された制御を、選択されたアンカー制御の水平軸に沿って位置合わせします。



**ボトムに位置合わせ**

選択された制御を、選択されたアンカー制御の下端に位置合わせします。

## ビルド・ツールバー



このツールバーを使用して、アプリケーションのビルドおよびテストを行います。

ツールバーには次のようなアイコンがあります。



**ビルド**

アプリケーションをビルドします。



**テスト**

アプリケーションをテストします。

## Mobile Application Builder インターフェース・ペイン

Mobile Application Builder インターフェースには、以下のペインがあります。

### プロジェクト

「プロジェクト」ペインを使用して、プロジェクト・リソースを表示したり、プロジェクト・リソースで作業を行ったりします。詳しくは、『「プロジェクト」ペイン・オプション』を参照してください。

### プロパティおよびイベント

「プロパティおよびイベント」ペインを使用して、選択した制御のプロパティやイベントを表示したり、編集したりします。詳しくは、11 ページの『「プロパティおよびイベント」ペイン・オプション』を参照してください。

### フォーム

「フォーム」ペインでは、作成されたフォームを表示します。パレットから制御を選択し、その制御を置く場所をフォーム上でクリックすることができます。「フォーム」ペインでオブジェクトを選択し、それらに関するアクションを呼び出すことができます。詳しくは、11 ページの『「フォーム」ペイン・オプション』を参照してください。

### 制御パレット

制御パレットでは、指定されたデバイスで使用可能なすべてのフォーム制御を表示します。詳しくは、12 ページの『制御パレット・オプション』を参照してください。

## 「プロジェクト」ペイン・オプション

「プロジェクト」ペインを使用して、プロジェクト・リソースを表示したり、プロジェクト・リソースで作業を行ったりします。

「プロジェクト」ペインでは、階層内のオブジェクト間の関係を表示します。特定のオブジェクトを拡張すると、そのオブジェクトの中に存在する、あるいは含まれるオブジェクトが下に表示されます。

オブジェクトの中には、他のオブジェクトを含まないものもあります。そのようなオブジェクトのフォルダーは、「プロジェクト」ペインの一番下に表示されます。

「プロジェクト」ペインを縮小して、現在作業を行いたいオブジェクトのみを表示することができます。特定のオブジェクトの下の「プロジェクト」ペインを縮小するには、オブジェクトの隣の負符号 (-) をクリックします。

「プロジェクト」ペイン内のオブジェクトでアクションを呼び出すには、そのオブジェクトを右マウス・ボタンでクリックして、有効なアクションのポップアップ・メニューをオープンします。そして、メニュー項目を選択します。ウィンドウまたはノートブックがオープンして、そのアクションを実行するために必要なステップをガイドします。

「プロジェクト」ペイン内のオブジェクトに関するアクションを選択する別の方法としては、そのオブジェクトを選択してから、有効なアクションのリストを含む「**選択**」メニューを選択する方法もあります。そして、希望のアクションのメニュー項目を選択します。

## 「プロパティおよびイベント」 ペイン・オプション

「プロパティおよびイベント」 ペインを使用して、選択した制御のプロパティやイベントを表示したり、編集したりします。「プロパティ」 ページで、制御の概観、および制御の振る舞いを決定します。「イベント」 ページで、制御が特定のアクションにどのように反応するかを決定します。各制御に、異なるアクションを取らせることができます。

「プロジェクト」 ペインまたは「フォーム」 ペインでフォームまたは制御を選択し、そのフォームまたは制御に関連したプロパティまたはイベントを、「プロパティおよびイベント」 ペインで表示または編集します。

## 「フォーム」 ペイン・オプション

「フォーム」 ペインでは、作成されたフォームを表示します。パレットから制御を選択し、その制御を置く場所をフォーム上でクリックすることができます。「フォーム」 ペインでオブジェクトを選択し、それらに関するアクションを呼び出すことができます。

「フォーム」 ペイン内のオブジェクトに関するアクションを呼び出すには、オブジェクトを選択します。そして、選択したオブジェクトを右マウス・ボタンでクリックして、有効なアクションのポップアップ・メニューをオープンし、メニュー項目を選択します。ウィンドウまたはノートブックがオープンして、そのアクションを実行するために必要なステップをガイドします。

「フォーム」 ペイン内のオブジェクトに関するアクションを選択する別の方法としては、そのオブジェクトを選択してから、有効なアクションのリストを含む「**選択**」メニューを選択する方法もあります。そして、希望のアクションのメニュー項目を選択します。

## 制御パレット・オプション



制御パレットでは、指定されたデバイスで使用可能なすべてのフォーム制御を表示します。

制御を左マウス・ボタンでクリックして、選択します。カーソルが十字線に変わります。カーソルをフォーム上の希望の場所へ移動し、左マウス・ボタンでフォームをクリックして制御をドロップします。左マウス・ボタンを押したままにしてカーソルを移動し、制御のサイズを変更します。

---

## キーボード・ショートカットおよびアクセラレーターの使用

マウスを使用して行うことができる操作を、キーまたはキーの組み合わせを使用して実行することができます。

### メニュー・アクション

メニュー・アクションは、次の方法でキーボードから開始できます。

- Alt キーを押したまま、選択するメインメニュー・オプションの名前の後にある括弧内の文字を押します。たとえば、「ドキュメンテーション」メニュー・オプションを選択するには、Alt+H を押して、次に D を押してください。
- アクセラレーターを持っているメニュー項目もあります。アクセラレーターを使用すると、メニューを展開せずにメニュー・オプションを呼び出すことができます。たとえば、CTRL+S を押してプロジェクトを保管することができます。
- オプションを選択せずにメインメニューを終了するには、Esc キーを押します。

### ドラッグ・アンド・ドロップ・オプション

フォームへの制御の追加などのドラッグ・アンド・ドロップ・アクションは、テンキー・アクセシビリティ・オプションを使用可能することで実行できます。テンキー・アクセシビリティ・オプションを使用可能にするには、次のようにします。

- 「スタート」→「設定」→「コントロール パネル」に移動して、「ユーザー補助のオプション」をオープンします。
- 「マウス」ページで、「マウス キー機能を使う」チェック・ボックスを選択して、「適用」、次に「OK」をクリックします。
- キーボードの **NumLock** を使用可能にします。

テンキーを使用可能にした後で、テンキーを使用して、下の表で説明されているようにアクションを実行します。

表1. テンキー・アクション

アクション	ショートカット
カーソルを左または右に移動する	左または右矢印キー
カーソルを上または下に移動する	上または下矢印キー
制御パレットまたはフォームから制御を選択する	中央キー (別の方法: Ctrl+Tab で「選択部分」ドロップダウン・リストにフォーカスを置いて、次に Shift+下矢印でドロップダウン・リストからリソースまたは制御を選択します)
制御をフォームにドロップする	<ol style="list-style-type: none"> <li>1. カーソルをフォームに移動して、そこに置きます。</li> <li>2. Insert を押して制御をドロップします。</li> <li>3. Insert + 矢印キーを押して制御のサイズを変更します。</li> <li>4. 中央キーを押してサイズ変更の設定を受け入れます。</li> </ol>
フォーム上の制御のサイズを変更する	<ol style="list-style-type: none"> <li>1. 制御を選択します。</li> <li>2. カーソルがドラッグ・ハンドルに変わるまで、カーソルを制御の端に移動します。</li> <li>3. Insert + 矢印キーを押して制御のサイズを変更します。</li> <li>4. 中央キーを押してサイズ変更の設定を受け入れます。</li> </ol>

表2. 一般的なキーボード・ショートカットおよびアクセラレーター

アクション	ショートカット
メニュー・バーのアクセス	Alt+メニュー・ショートカット文字
次のメニュー項目に移動	矢印キー、またはメニュー・オプション内の括弧内の文字
ウィンドウ内の次のフィールドに移動	Tab
ウィンドウ内の前のフィールドに戻る	Shift+Tab
新規プロジェクトを作成	Ctrl+N
既存のプロジェクトをオープン	Ctrl+O
プロジェクトの保管	Ctrl+S
プロジェクトの別名保管...	Ctrl+A
終了	Alt+F4

表 3. 表アクションのためのキーボード・ショートカット

アクション	ショートカット
次の編集可能なセルを編集、または次のコンポーネントにフォーカスを移動する	Tab
前の編集可能なセルを編集、または前のコンポーネントにフォーカスを移動する	Shift+Tab
次のフィールドにフォーカスを移動する	Ctrl+Tab
前のフィールドにフォーカスを移動する	Ctrl+Shift+Tab
編集可能であれば現行セルを編集する	F2
セルのプルダウン・メニューから項目を選択する	F2、次に Ctrl+Alt+下矢印でプルダウン・メニューをオープンし、下矢印と上矢印で項目を選択して Enter キーを押す
上または下のセルに移動する	上矢印または下矢印
左または右のセルに移動する	左矢印または右矢印

表 4. ツリー・ナビゲーションのためのキーボード・ショートカット

アクション	ショートカット
プロジェクト・ツリーにフォーカスを置く	フォーカスがメインメニューに移動するまで Shift+Tab を押し、次に下矢印キーでフォーカスをプロジェクト・ツリーに移動します
前方にナビゲートする	Tab
後方にナビゲートする	Shift+Tab
エントリーを展開する	右矢印
エントリーを縮小する	左矢印
エントリーの展開/縮小を切り替える	Enter
エントリーを上/下に 1 つ移動する	上矢印または下矢印
先頭のエントリーに移動する	Home
表示されている最後のエントリーに移動する	End

表 5. 一般的な編集アクション

アクション	ショートカット
コピー	Ctrl+C
切り取り	Ctrl+X
貼り付け	Ctrl+V
全選択	Ctrl+A
削除	Delete キー

表 6. スクリプト・エディター・アクション

アクション	ショートカット
コピー	Ctrl+C
切り取り	Ctrl+X
貼り付け	Ctrl+V
全選択	Ctrl+A
削除	Delete キー
検索	Ctrl+F
置換	Ctrl+H

---

## MAB バージョン 8.1.4 のセットアップおよび構成

ターゲット・モバイル・デバイスのプラットフォーム用に、ビルドとセットアップ・テストの作業のみが必要になります。

---

### 関連タスク:

108 ページの『Mobile Application Builder アプリケーションのビルド』

113 ページの『Mobile Application Builder アプリケーションのテスト』

## MAB バージョン 8.1.4 Palm のセットアップおよび構成

Palm アプリケーションをビルドおよびテストするための Mobile Application Builder 環境をセットアップするには、以下を行います。

---

### Mobile Application Builder バージョン 8.1.4 Palm アプリケーション開発のセットアップおよび構成タスク:

- 『Palm C コード・ビルド環境のセットアップ』
  - 18 ページの『テストのための Palm エミュレーターのセットアップ』
  - 20 ページの『テストのための Palm OS 5 Simulator のセットアップ』
  - 22 ページの『テストのための Palm モバイル・デバイスのセットアップ』
- 

### 関連タスク:

108 ページの『Mobile Application Builder アプリケーションのビルド』

113 ページの『Mobile Application Builder アプリケーションのテスト』

### Palm C コード・ビルド環境のセットアップ

以下のツール (ただし、Palm OS SDK を除く) は、IBM の DB2 Everyplace Web サイトからダウンロードできます。これらのツールは、DB2 Everyplace SDK CD にも収められています。DB2 Everyplace SDK のインストールで「Mobile Application Builder Toolkits」オプションを選択した場合、必要なツールが

```
...\SDK\ApplicationBuilder\Toolkits\Palm Development\MAB_prereqs
```

ディレクトリーにインストールされます。PalmOS 用の C コード・アプリケーションを作成するために必要なツールの多くが組み込まれます。ただし、PalmOS SDK は Palm Computing から入手しなければなりません。以下の各ツールの説明を読んでから、各ツールのリンクをたどって IBM ダウンロード・サイトへ進んでください。IBM サイトからダウンロードする場合は、ユーザー ID とパスワードの入力が必要になります。まだ登録していない場合は、登録を済ませてから先へ進む必要があります。

前提ツールをインストールしたら、ツールのインストール場所を示すように Mobile Application Builder のプリファレンスを設定します。

本節では、以下のような必須の Palm C コード・ビルド・ツールのインストールについて説明します。

- Cygwin v1.3.22-1 (GNU PRC-tools および PiIRC リソース・コンパイラーが含まれています)
- Palm OS SDK 5

---

Cygwin v1.3.22-1 には、Palm OS 用の PRC-tools および PiIRC リソース・コンパイラーが含まれています。DB2 Everyplace SDK から Cygwin を取得する場合、これらはインストールされることが事前に選択されています。Palm Web サイトから Cygwin をダウンロードする場合、インストールで以下の Palm OS 関連パッケージを「Devel」カテゴリーから選択する必要があります。

- PRC-Tools
- PiIRC
- 作成コンポーネント

#### **Cygwin をインストールするには:**

1. 前バージョンの Cygnus または Cygwin をアンインストールします。
2. ... \SDK\ApplicationBuilder\Toolkits\Palm Development\MAB\_prereqs\Cygwin\Setup.exe を実行します。
3. 「インストール・タイプの選択」ページで「ローカル・ディレクトリーからインストール」を選択して、「次へ」をクリックします。
4. 「インストール・ディレクトリーの選択」ページで、デフォルト・ディレクトリー (c:\cygwin) とその他すべてのデフォルト設定をそのままにして、「次へ」をクリックします。
5. 「ローカル・パッケージ・ディレクトリーの選択」で、デフォルト・ディレクトリーをそのまま使用して「次へ」をクリックします。
6. 「パッケージの選択」ページで、デフォルトのまま「次へ」をクリックします。
7. 製品のインストール後、デスクトップまたは「スタート」メニューにアイコンを作成することを選択できます。次に「完了」をクリックしてセットアップを終了します。

#### **Palm OS SDK 5 for PRC-Tools をインストールするには:**

SDK には、Palm OS C ライブラリーとヘッダー・ファイルが組み込まれています。これらの SDK は、GNU コンパイラーで機能するように変更されています。Windows の場合は、Palm SDK 5 を使用してアプリケーションをコンパイルしてください。Windows の場合、Palm SDK 5 は Palm Computing から直接ダウンロードしなければなりません。以下のダウンロード・リンクを使用して SDK を入手してください。

1. SDK を保持するディレクトリーを選択または作成します。このディレクトリーは、先に PRC-Tools 2.0 に対して指定したディレクトリーと一致していなければなりません。

2. 「Palm OS SDK version 5」ページに移動し、「同意 (I Agree)」をクリックしてライセンス情報に同意します。「Palm 5 SDK」ページが表示されます。
3. 「Palm OS SDK 5 for PRC-Tools」セクションで、「Tar」をクリックします。ZIP アーカイブには、MAB を使用した Palm の開発に必要なヘッダー・ファイルが含まれていないため、「Zip」をクリックしないでください。
4. ZIP ファイルからファイルを抽出できるプログラム (WinZip など) を使用して、palmos-sdk-5.0-1.tar.gz を抽出し任意のディレクトリーに入れます。この種のプログラムを持っていない場合は、<http://www.winzip.com> から無料の体験版を入手することができます。

#### オプション: Palm SG SDK をインストールするには:

Palm デバイスで 5 ウェイ・ナビゲーションを使用可能にするには、SDK をインストールしなければなりません。この SDK を取得するには、Palm PluggedIn developer program に加入する必要があります。

1. Palm SG SDK v1.0.zip ファイルをダウンロードします。
2. zip ファイルを、前節で Palm SDK 5 をインストールしたディレクトリーの下の \include ディレクトリーに解凍します。たとえば、c:\PalmDev\sdk-5\include のようにします。
3. 新しいヘッダー・ファイルを MAB で認識可能にするためには、次節で説明されている Palm ビルド・ツール・プリファレンス・ウィンドウを開いて「OK」をクリックします。

#### Palm ビルド・ツール・プリファレンスを設定するには:

1. Mobile Application Builder を開始します。初めて Mobile Application Builder 8.1.4 を開始した場合は、ウィンドウが開き、ツール・プリファレンスの設定を促す指示が出されます。
2. 新規または既存の Palm プロジェクトをオープンします。
3. 「ファイル」→「プリファレンス」の順にクリックします。「プリファレンス」ウィンドウが開きます。
4. 「Palm ツール」をクリックして Palm ツールのプリファレンスを表示します。
5. Cygwin、PRC tools、および PilRC の場合、Cygwin がインストールされているルート・ディレクトリーの下の \bin サブディレクトリーのパスを指定します。たとえば、Cygwin が c:\cygwin ディレクトリーにインストールされている場合は c:\cygwin\bin と指定します。
6. Palm SDK の場合、ツールがインストールされているルート・ディレクトリーのパスを指定します。たとえば、Palm SDK 5 ファイルが C:\PalmDev にインストールされている場合は C:\PalmDev と入力します。
7. 「OK」をクリックします。「プリファレンス」ウィンドウが閉じます。
8. Mobile Application Builder を再始動すると、変更が有効になります。

『Palm セットアップ』に戻ります。

## テストのための Palm エミュレーターのセットアップ

Palm OS Emulator (POSE) を使用すると、実際の Palm デバイスにダウンロードする前に Palm アプリケーションをテストすることができます。POSE 3.5 以上の使用をお勧めします。このエミュレーターを使用して、Palm OS 4.0 およびそれ以前用のアプリケーションをテストできます。

特定の Palm OS デバイスを表示させない限りは、エミュレーターの「スキン」は必要ありません。デフォルトのスキンは汎用 Palm OS デバイスです。

エミュレーターを実行するには、作成するアプリケーションの対象となるデバイスで使用する PalmOS レベルの ROM イメージをダウンロードする必要があります。そのダウンロードを行うには、以下のようないくつかの方法があります。

- 米国内では、Palm Alliance Program に加入して、ROM イメージを Palm ダウンロード・サイトから取得する必要があります。通常、ROM イメージは 1 日か 2 日で取得できます。
- 米国以外の国では、Palm Developer Seeding Program に加入して ROM イメージを取得する必要があります。このプロセスには、多少時間がかかります (2 から 4 週間)。
- Palm 製ではないが PalmOS ベースのデバイスの場合は、デバイス・ライセンス所有者のデベロッパー・プログラムから ROM を取得します。デバイスによっては、Palm ダウンロード・サイトからダウンロードできるものもあります。

ROM イメージのロード手順とエミュレーターの使用方法について詳しくは、Palm Web サイトから入手できる「*Palm OS Development Tools Guide*」を参照してください。

Mobile Application Builder アプリケーションをテストするには、DB2 Everyplace データベース・エンジンをエミュレーターにインストールする必要があります。Mobile Application Builder に含まれているサンプル・アプリケーション (NurseInit など) もインストールできます。このアプリケーションは、DB2 Everyplace エンジンと Mobile Application Builder サンプルで使用するいくつかの表を使用して DB2 Everyplace データベースを初期化します。

---

### テストのために Palm エミュレーターをセットアップするには:

1. Palm OS エミュレーターをダウンロードします。
2. ZIP ファイルを Mobile Application Builder インストール・ディレクトリー (たとえば、C:\DB2EveryplaceSDK\SDK\ApplicationBuilder\bin) に保管します。
3. このファイルをこのディレクトリーで unzip します。emulator.exe ファイルを Mobile Application Builder インストール・ディレクトリーに入れておかないと、「テスト」ツールバーまたはメニュー項目を使用してアプリケーションをテストするときに、ユーザー・インターフェースはそのファイルを見つけることができません。
4. DB2 Everyplace データベース・エンジンを Palm OS Emulator にインストールします。

- emulator.exe をダブルクリックしてエミュレーターを始動します。エミュレーターのダウンロード時にスキン・ファイルを取得しない場合は、エミュレーターを始動するときその旨の警告メッセージが出ます。このメッセージを無視してデフォルトの汎用スキンを使用することもできますし、特定の Palm デバイスを表示したい場合は、このメッセージの指示に従ってスキン・ファイルをダウンロードし、インストールすることもできます。
- エミュレーター画面の任意の場所で右マウス・ボタン・クリックします。「アプリケーション/データベースのインストール」をクリックします。
- 以下のファイルをインストールします。
  - ¥Clients¥PalmOS¥database¥DB2eCAT.prc
  - ¥Clients¥PalmOS¥database¥DB2eCLI.prc
  - ¥Clients¥PalmOS¥database¥DB2eComp.prc
  - ¥Clients¥PalmOS¥database¥DB2eRunTime.prc
  - \Clients\PalmOS\database\DB2eDMS.prc
  - \Clients\PalmOS\database\PBSPkcs11.prc
  - \Clients\PalmOS\database\CryptoPlugin.prc

**ヒント:** これらの .prc ファイルをインストールするには、Windows エクスプローラーを使用して、それらをエミュレーター画面にドラッグ・アンド・ドロップします。

5. ファイルをエミュレーターにインストールしたら、エミュレーター画面上の任意の場所を右クリックし、「リセット」をクリックしてデバイスをリセットします。

**オプション: NurseInit および CLP サンプル・アプリケーションをインストールするには:**

コマンド行プロセッサ (CLP) 機能を使用して、表に対して SQL ステートメントを発行できます。

1. エミュレーター画面の任意の場所で右マウス・ボタン・クリックします。「アプリケーション/データベースのインストール」をクリックします。
2. 177 ページの『DB2 Everyplace のサンプル・アプリケーション』での説明に従って、サンプル・アプリケーションをインストールします。
3. ファイルをエミュレーターにインストールしたら、エミュレーター画面上の任意の場所を右クリックし、「リセット」をクリックしてデバイスをリセットします。
4. NurseInit アプリケーションをエミュレーターで実行します。NurseInit アプリケーションはレコードを表に挿入します。
5. 「初期化」をタップして、レコードを作成し、それらを VNSCHEDULE、VNPERSO、VNMEDICALRECORD、VNCONTACT、および VNSIGNATURE 表に挿入します。

**オプション: 同期サーバーを使用した同期化のためにエミュレーターを構成するには:**

アプリケーションに同期機能が備わっている場合は、同期サーバーの構成も行う必要があります。

- 「IBM DB2 Everyplace インストールおよびユーザズ・ガイド」の第 3 章『モバイル・デバイスを同期するためのセットアップ』に記載されている指示に従います。

**オプション: デフォルトのエミュレーター・デバッグ設定を変更するには、次のようにします:**

デフォルトでは、Palm エミュレーターはデバッグ用の警告を表示します。警告の中には重大でないものもあり、それらは無視してもかまいません。たとえば、バージョン 3.5x のエミュレーターは、アプリケーションをクローズするたびに警告を表示します。

1. エミュレーター画面の任意の場所で右マウス・ボタン・クリックします。「設定」→「デバッグ」の順にクリックします。
2. 「MemMgr リーク」チェック・ボックスをクリアします。

『Palm セットアップ』に戻ります。

---

## テストのための Palm OS 5 Simulator のセットアップ

Palm OS Simulator を使用して、実際の Palm デバイスにダウンロードする前に Palm 5.x アプリケーションをテストすることができます。このシミュレーターは、Palm OS 5.x 用のアプリケーションのみをテストするため使用できます。

シミュレーターの使用方法について詳しくは、Palm Web サイトから入手できる「*Palm OS Development Tools Guide*」を参照してください。

Mobile Application Builder アプリケーションをテストするには、DB2 Everyplace データベース・エンジンをシミュレーターにインストールする必要があります。

Mobile Application Builder に含まれているサンプル・アプリケーション (NurseInit など) もインストールできます。このアプリケーションは、DB2 Everyplace エンジンと Mobile Application Builder サンプルで使用するいくつかの表を使用して DB2 Everyplace データベースを初期化します。

以下のステップに従ってこれらのファイルをインストールした後に、シミュレーター・セッションを手動で保管する必要があります。

---

テストのために **Palm OS 5 Simulator** をセットアップするには:

1. Palm OS 5 Simulator をダウンロードします。
2. ZIP ファイルを任意のディレクトリーに保管します。
3. 次のようにして DB2 Everyplace データベース・エンジンを Palm OS Simulator にインストールします。
  - PalmSim.exe をダブルクリックしてシミュレーターを開始します。
  - シミュレーター画面の任意の場所を右マウス・ボタン・クリックし、ポップアップ・メニューの「インストール」→「データベース」をクリックします。
  - 以下のファイルをインストールします。
    - %Clients%\PalmOS\database\DB2eCAT.prc

- ¥Clients¥PalmOS¥database¥DB2eCLI.prc
- ¥Clients¥PalmOS¥database¥DB2eComp.prc
- ¥Clients¥PalmOS¥database¥DB2eRunTime.prc
- \Clients\PalmOS\database\DB2eDMS.prc
- \Clients\PalmOS\database\PBSPkcs11.prc
- \Clients\PalmOS\database\CryptoPlugin.prc

**ヒント:** これらの .prc ファイルは、Windows エクスプローラーからシミュレーター画面にドラッグ・アンド・ドロップしてインストールすることもできます。

4. ファイルをシミュレーターにインストールしたら、シミュレーター画面上の任意の場所を右マウス・ボタン・クリックし、「リセット」をクリックしてデバイスをリセットします。
5. シミュレーター画面の任意の場所を右マウス・ボタン・クリックし、「ストレージ」→「保管」をクリックしてシミュレーター・セッションのスナップショットを保管します。

**オプション: NurseInit および CLP サンプル・アプリケーションをインストールするには:**

コマンド行プロセッサ (CLP) 機能を使用して、表に対して SQL ステートメントを発行できます。

1. シミュレーター画面の任意の場所で右マウス・ボタン・クリックします。「インストール」→「データベース」をクリックします。
2. 177 ページの『DB2 Everyplace のサンプル・アプリケーション』での説明に従って、サンプル・アプリケーションをインストールします。
3. ファイルをシミュレーターにインストールしたら、シミュレーター画面上の任意の場所を右マウス・ボタン・クリックし、「リセット」をクリックしてデバイスをリセットします。
4. NurseInit アプリケーションをシミュレーターで実行します。NurseInit アプリケーションはレコードを表に挿入します。
5. 「初期化」をタップして、レコードを作成し、それらを VNSCHEDULE、VNPERSO、VNMEDICALRECORD、VNCONTACT、および VNSIGNATURE 表に挿入します。

**オプション: 同期サーバーを使用した同期化のためにシミュレーターを構成するには:**

アプリケーションに同期機能が備わっている場合は、同期サーバーの構成も行う必要があります。

- 「IBM DB2 Everyplace インストールおよびユーザズ・ガイド」の第 3 章『モバイル・デバイスを同期するためのセットアップ』に記載されている指示に従います。

『Palm セットアップ』に戻ります。

## テストのための Palm モバイル・デバイスのセットアップ

---

テストのために **Palm** モバイル・デバイスをセットアップするには:

1. モバイル・デバイスに添付されている手順に従って、接続ソフトウェアを (Palm OS デバイス用の HotSync など) をインストールします。 Mobile Application Builder は Palm OS 3.5 以上をサポートします。
2. DB2 Everyplace エンジンと一緒に提供された「*IBM DB2 Everyplace* インストールおよびユーザーズ・ガイド」の手順に従って、DB2 Everyplace エンジンをモバイル・デバイスにインストールします。以下のファイルをインストールします。
  - %Clients%\PalmOS\database\DB2eCAT.prc
  - %Clients%\PalmOS\database\DB2eCLI.prc
  - %Clients%\PalmOS\database\DB2eComp.prc
  - %Clients%\PalmOS\database\DB2eRunTime.prc
  - \Clients\PalmOS\database\PBSPkcs11.prc
  - \Clients\PalmOS\database\CryptoPlugin.prc

**オプション: NurseInit および CLP サンプル・アプリケーションをインストールするには:**

コマンド行プロセッサ (CLP) 機能を使用して、表に対して SQL ステートメントを発行できます。

1. 177 ページの『DB2 Everyplace のサンプル・アプリケーション』での説明に従って、サンプル・アプリケーションをインストールします。
2. NurseInit アプリケーションをモバイル・デバイスで実行します。NurseInit アプリケーションはレコードを表に挿入します。
3. 「初期化」をタップして、レコードを作成し、それらを VNSCHEDULE、VNPERSO、VNMEDICALRECORD、VNCONTACT、および VNSIGNATURE 表に挿入します。

**オプション: 同期サーバーを使用した同期化のためにデバイスを構成するには:**

アプリケーションに同期機能が備わっている場合は、同期サーバーの構成も行う必要があります。

- 「*IBM DB2 Everyplace* インストールおよびユーザーズ・ガイド」の第 3 章『モバイル・デバイスを同期するためのセットアップ』に記載されている指示に従います。

『Palm セットアップ』に戻ります。

---

## MAB バージョン 8.1.4 Symbian Crystal のセットアップおよび構成

Symbian Crystal アプリケーションをビルドおよびテストするための Mobile Application Builder 環境をセットアップするには、以下を行います。

---

### Mobile Application Builder バージョン 8.1.4 のセットアップおよび構成タスク:

- 『Symbian Crystal Java コード・ビルド環境のセットアップ』
- 24 ページの『テストのための Symbian Crystal エミュレーターのセットアップ』
- 26 ページの『テストのための Symbian Crystal モバイル・デバイスのセットアップ』

---

### 関連タスク:

108 ページの『Mobile Application Builder アプリケーションのビルド』

113 ページの『Mobile Application Builder アプリケーションのテスト』

### Symbian Crystal Java コード・ビルド環境のセットアップ

前提ツールをインストールしたら、ツールのインストール場所を示すように Mobile Application Builder のプリファレンスを設定します。

本節では、次の Symbian ツールのインストールについて説明します。このツールは、Nokia Communicator 92xx などの Crystal デバイスでアプリケーションのビルドおよびテストを行うために必要です。

- Nokia 9200 Series SDK for Symbian OS

Windowsの場合、Nokia 9200 Series SDK for Symbian OS は、Forum Nokia から直接ダウンロードしなければなりません。以下のダウンロード・リンクを使用して SDK を入手してください。SDK CD を自分あてにメール送信するよう登録するか、または SDK をダウンロードする必要があります。SDK には Windows デスクトップ・エミュレーターが組み込まれています。

SDK には JRE が組み込まれています。これを使用することもできますし、IBM Java Technology Zone (<http://www.ibm.com/developerworks/java>) または Sun Microsystems (<http://java.sun.com/products>) から JDK を入手することもできます。別の JDK を使用する場合は、JDK %bin ディレクトリーが組み込まれるように Windows Environment PATH 変数を設定します。

SDK には Perl が組み込まれています。他のバージョンの Perl がインストールされている場合は、PATH 環境変数で、他方の Perl バージョンが Symbian SDK の Perl より前に現れると、アプリケーション・ビルド・プロセスが失敗することがあります。C:%Perl%bin が他のどの Perl バージョンよりも前に現れていることを確認してください。

---

### Nokia 9200 Series SDK for Symbian OS をインストールするには:

- SDK を Forum Nokia からダウンロードします。Nokia 9200 Series SDK for Symbian OS ZIP ファイルを任意の一時ディレクトリーに保管します。

- Nokia 9200 Series SDK for Symbian OS の実行可能 (.exe) ファイルを実行します。インストール・ウィザードがオープンします。デフォルトのインストール・ディレクトリーは C:\Symbian\6.x\NokiaJava です。
- すべてのコンポーネントを選択し、それらのコンポーネントのデフォルト・ディレクトリーを保存します。これらのコンポーネントを SDK と異なるドライブにインストールすると、ビルド・エラーになることがあります。JDK 1.2 以上がすでに Windows デスクトップにインストールされている場合は、それを使用し、JRE をインストールしないことを選択できます。

#### Symbian ビルド・ツール・プリファレンスを設定するには:

Crystal と UIQ SDK の両方を同時に使用する場合、これらが同じルート・ディレクトリー (例: c:\Symbian) にインストールされていることを確認し、これらがインストールされているディレクトリーを指すように次のようにしてプリファレンスを設定します。

1. Mobile Application Builder を開始します。初めて Mobile Application Builder 8.1.4 を開始した場合は、ウィンドウが開き、ツール・プリファレンスの設定を促す指示が出されます。プロジェクトをオープンまたは開始する必要はありません。
2. 「ファイル」→「プリファレンス」の順にクリックします。「プリファレンス」ウィンドウが開きます。
3. 「Symbian ツール」をクリックして Symbian ツールのプリファレンスを表示します。SDK がインストールされているディレクトリーのパス (ドライブを含む) を指定します。たとえば、c:\temp のようにします。
4. 「OK」をクリックします。「プリファレンス」ウィンドウが閉じます。

『Crystal セットアップ』に戻ります。

### テストのための Symbian Crystal エミュレーターのセットアップ

Nokia 9200 Series SDK for Symbian OS は、Windows デスクトップ・エミュレーターを提供します。Symbian Crystal Java コード・ビルド環境をセットアップしておかないと、Symbian Crystal エミュレーターをテスト用にセットアップすることはできません。

#### テストのために Symbian Crystal エミュレーターをセットアップするには:

- DB2 Everyplace データベース・エンジンと JDBC ドライバーをエミュレーターにインストールします。
  - 以下のファイルを  
 %Symbian\6.x\NokiaJava\Epoc32\Release\wins\ure1 にコピーします。  
 \Clients\Symbian6\database\wins\db2e.d11  
 \Clients\Symbian6\database\wins\db2ejdbc.d11
  - 以下のファイルを  
 %Symbian\6.x\NokiaJava\erj\ext にコピーします。

\\Clients\Symbian6\database\wins\db2ejdbc.jar

**オプション: NurseInit アプリケーションをインストールおよび実行するには:**

DB2 Everyplace SDK とともに提供される NurseInit アプリケーションは、SDK に入っている Symbian チュートリアルを修了するために必要な表を作成したり、表にデータを追加するように設計されたアプリケーションです。

1. NurseInit.SIS ファイルを \\Symbian\6.x\NokiaJava\erj ディレクトリーにコピーします。このファイルの場所は 177 ページの『DB2 Everyplace のサンプル・アプリケーション』に記されています。
2. Symbian エミュレーターをオープンします。
3. Extras ディレクトリーから「制御パネル」をオープンします。
4. インストール・プログラムをオープンします。
5. 「**新規インストール**」を選択します。
6. J: ドライブまでブラウズして、NurseInit.sis ファイルをインストールします。
7. インストール・プログラムをクローズし、**Extras** から NurseInit を実行します。

**オプション: Symbian Crystal エミュレーターの起動時間を短縮するには:**

以下のステップを実行して、Nokia エミュレーターの起動時間を短縮します。以下のステップを完了しないと、エミュレーターのオープンに 1 分くらいかかることがあります。

- Windows Explorer を使用して、次を表示します。  
\\Symbian\6.x\NokiaJava\Epoc32\Release\wins\ure1\z\system\LIBS
- このロケーションに hidden という名前のディレクトリーを作成します。
- ファイル LINDASTART.DLL を  
– \\Symbian\6.x\NokiaJava\Epoc32\Release\wins\ure1\z\system\LIBS  
から \\hidden ディレクトリーに移動します。
- ファイル phonet.dll を  
\\Symbian\6.x\NokiaJava\Epoc32\Release\wins\ure1\  
から \\hidden ディレクトリーに移動します。

**オプション: 同期サーバーを使用した同期化のために Symbian エミュレーターを構成するには:**

アプリケーションに同期機能が備わっている場合は、同期サーバーの構成も行う必要があります。

- 「*IBM DB2 Everyplace* インストールおよびユーザーズ・ガイド」の第 3 章『モバイル・デバイスを同期するためのセットアップ』に記載されている指示に従います。

『Crystal セットアップ』に戻ります。

## テストのための Symbian Crystal モバイル・デバイスのセットアップ

---

テストのために **Symbian Crystal** デバイスをセットアップするには:

- DB2 Everyplace データベース・エンジンと JDBC ドライバーをデバイスにインストールします。
  - 以下のファイルをインストールします。

```
\Clients\Symbian6\database\armi\DB2e.SIS  
\Clients\Symbian6\database\armi\db2ejdbc.SIS
```

**オプション: 同期サーバーを使用した同期化のためにデバイスを構成するには:**

アプリケーションに同期機能が備わっている場合は、同期サーバーの構成も行う必要があります。

- 「*IBM DB2 Everyplace* インストールおよびユーザズ・ガイド」の第 3 章『モバイル・デバイスを同期するためのセットアップ』に記載されている指示に従います。

**オプション: NurseInit アプリケーションをインストールおよび実行するには:**

DB2 Everyplace SDK とともに提供される NurseInit アプリケーションは、SDK に入っている Symbian チュートリアルを修了するために必要な表を作成したり、表にデータを追加するように設計されたアプリケーションです。

- NurseInit.sis ファイルをデバイスにインストールします。このファイルの場所は 177 ページの『DB2 Everyplace のサンプル・アプリケーション』に記されています。
- デバイスでアプリケーションを実行します。

『Crystal セットアップ』に戻ります。

---

## MAB バージョン 8.1.4 Symbian UIQ のセットアップおよび構成

以下のタスクを行って、Symbian UIQ アプリケーションをビルドおよびテストするための Mobile Application Builder 環境をセットアップします。

---

**Mobile Application Builder バージョン 8.1.4 のセットアップおよび構成タスク:**

- 27 ページの『Symbian UIQ Java コード・ビルド環境のセットアップ』
  - 28 ページの『テストのための Symbian UIQ エミュレーターのセットアップ』
  - 29 ページの『テストのための Symbian UIQ デバイスのセットアップ』
- 

関連タスク:

108 ページの『Mobile Application Builder アプリケーションのビルド』

113 ページの『Mobile Application Builder アプリケーションのテスト』

## Symbian UIQ Java コード・ビルド環境のセットアップ

本節では、次の Symbian ツールのインストールについて説明します。このツールは、Sony Ericsson P800 などの UIQ デバイスでアプリケーションのビルドおよびテストを行うために必要です。ツールのインストール後に、ツールがインストールされている場所を指すように Mobile Application Builder でビルド・プリファレンスを設定する必要があります。

- Symbian OS v7.0 SDK for UIQ

Windows の場合は、Symbian OS v7.0 SDK for UIQ を Ericsson Mobility World から直接ダウンロードしなければなりません。以下のダウンロード・リンクを使用して SDK を入手してください。SDK をダウンロードするためには登録を行う必要があります。Windows デスクトップ・エミュレーターもダウンロードで入手できます。

SDK には JRE が組み込まれています。これを使用することもできますし、IBM Java Technology Zone (<http://www.ibm.com/developerworks/java>) または Sun Microsystems (<http://java.sun.com/products>) から JDK を入手することもできます。別の JDK を使用する場合は、JDK %bin ディレクトリーが組み込まれるように Windows Environment PATH 変数を設定します。

SDK には Perl が組み込まれています。他のバージョンの Perl がインストールされている場合は、PATH 環境変数で、他方の Perl バージョンが Symbian SDK の Perl より前に現れると、アプリケーション・ビルド・プロセスが失敗することがあります。C:%Perl%bin が他のどの Perl バージョンよりも前に現れていることを確認してください。

---

### Symbian OS v7.0 SDK for UIQ をインストールするには:

1. SDK を Ericsson Mobility World からダウンロードします。Symbian OS v7.0 SDK for UIQ ZIP ファイルを任意の一時ディレクトリーに保管します。SDK の README の指示に従ってください。
2. SDK 実行可能 (.exe) ファイルを実行します。インストール・ウィザードがオープンします。デフォルトのインストール・ディレクトリーは X:\Symbian\UIQ\_70 です。
3. すべてのコンポーネントを選択し、それらのコンポーネントのデフォルト・ディレクトリーを保存します。すでに Active Perl と JDK がインストールされている場合、これらをインストールしないことを選択できます。
4. 現行バージョンの SDK では、ビルド・エラーを回避するために、aiftool.rh ファイルを Symbian\UIQ\_70\epoc32\include ディレクトリーに手動で移動する必要があります。このファイルは com.symbian.api.GT-shared\_0\_0\_70.sdkpkg IIRC から抽出できます。これは、SDK インストール・ファイルの 1 つです。このファイルは、通常の ZIP ファイルです。これを unzip して aiftool.rh ファイルを抽出します。

## Symbian ビルド・ツール・プリファレンスを設定するには:

Crystal と UIQ SDK の両方を同時に使用する場合は、これらが同じルート・ディレクトリー (例: X:\Symbian) にインストールされていることを確認し、これらがインストールされているディレクトリーを指すように次のようにしてプリファレンスを設定します。

1. Mobile Application Builder を開始します。初めて Mobile Application Builder 8.1.4 を開始した場合は、ウィンドウが開き、ツール・プリファレンスの設定を促す指示が出されます。プロジェクトをオープンまたは開始する必要はありません。
2. 「ファイル」→「プリファレンス」の順にクリックします。「プリファレンス」ウィンドウが開きます。
3. 「Symbian ツール」をクリックして Symbian ツールのプリファレンスを表示します。SDK がインストールされているディレクトリーのパス (ドライブを含む) を指定します。たとえば、c:\temp のようにします。
4. 「OK」をクリックします。「プリファレンス」ウィンドウが閉じます。

『Crystal セットアップ』に戻ります。

---

## テストのための Symbian UIQ エミュレーターのセットアップ

Symbian OS v7.0 SDK for UIQ には、Windows デスクトップ・エミュレーターが備わっています。Symbian UIQ Java コード・ビルド環境をセットアップしておかないと、Symbian UIQ エミュレーターをテスト用にセットアップすることはできません。

---

## テストのために Symbian UIQ エミュレーターをセットアップするには:

- DB2 Everyplace データベース・エンジンと JDBC ドライバーをエミュレーターにインストールします。

1. 以下のファイルを

\Symbian\UIQ\_70\Epoc32\Release\wincsw\ure1 にコピーします。

```
\Clients\Symbian7\database\wins\db2e.d11
\Clients\Symbian7\database\wins\ECPKCS11.d11
\Clients\Symbian7\database\wins\db2ejdbc.d11
\Clients\Symbian7\database\wins\cryptoplugin.d11
```

2. 次のファイルを

\Symbian\UIQ\_70\erj\ext にコピーします。

```
\Clients\Symbian7\database\wins\db2ejdbc.jar
```

**オプション: Nurselnit や他のサンプル・アプリケーションをインストールするには、次のようにします:**

Symbian UIQ サンプル・ファイルの場所は、177 ページの『DB2 Everyplace のサンプル・アプリケーション』に記されています。2 つの方法で Symbian UIQ エミュレーターにサンプル・アプリケーションをインストールできます。

**サンプル・アプリケーションをインストールするための第 1 の方法:**

1. 各サンプルの .SIS ファイルを \Symbian\UIQ\_70\erj\examples ディレクトリーにコピーします。
2. これらのファイルをエミュレーターにインストールします。ファイルのインストール方法について詳しくは、エミュレーターに付属する資料を参照してください。

#### サンプル・アプリケーションをインストールするためのもう 1 つの方法:

1. Symbian\UIQ\_70\epoc32\wincw\c\system\Apps ディレクトリー下にプロジェクト・ディレクトリーを作成します。たとえば、PersonList アプリケーション用に Symbian\UIQ\_70\epoc32\wincw\c\system\Apps\PersonList ディレクトリーを作成します。
2. 作成したディレクトリーに、次の場所から .SIS ファイルを除く すべてのファイルをコピーします。

Client\platform\database\Samples\Language

サンプル・アプリケーションは、自動的にエミュレーターにインストールされません。

『Crystal セットアップ』に戻ります。

---

## テストのための Symbian UIQ デバイスのセットアップ

本節では、DB2 Everyplace のライブラリーおよびサンプル・アプリケーションを Symbian OS バージョン 7 UIQ モバイル・デバイスにインストールする方法について説明します。インストールを行う前に、以下のことを行っておく必要があります。

- Windows ワークステーションで、Symbian UIQ モバイル・デバイスに付属している接続ソフトウェアのインストールおよび構成を行います。
- Symbian UIQ モバイル・デバイスをワークステーションに接続します。モバイル・デバイスの資料を参照して、デバイスが正しく接続されているかを確認してください。

---

#### テストのために Symbian UIQ デバイスをセットアップするには:

DB2 Everyplace ライブラリーおよびサンプル・アプリケーション・ファイルを手動でインストールするには、Symbian OS UIQ モバイル・デバイスに付属している接続ソフトウェアのインストール・ツールを使用します。

- DB2 Everyplace データベース・エンジンと JDBC ドライバーをデバイスにインストールします。

– \Clients\Symbian7\database\armi\DB2e.sis

– \Clients\Symbian7\database\armi\DB2eJDBC.sis

**オプション: Nurselnit およびその他のサンプル UIQ アプリケーションをインストールするには:**

Symbian UIQ サンプル・ファイルは \Clients\Symbian7\database にあります。以下のファイルをデバイスにインストールします。これらファイルの場所は 177 ページの『DB2 Everyplace のサンプル・アプリケーション』に記されています。

表 7. デバイス用の Symbian UIQ サンプル・アプリケーション

ファイル名	説明
DB2eCLP.sis	コマンド行プロセッサ (CLP) アプリケーション
PersonList.sis	PersonList サンプル・アプリケーション
NurseInit.sis	NurseInit サンプル・アプリケーション
DB2e_Symbian7.sis	DB2 Everyplace パッケージ。このパッケージには、以下が含まれています。 <ul style="list-style-type: none"> <li>• DB2e.sis</li> <li>• DB2eJDBC.sis</li> <li>• SampleCLP.sis</li> <li>• NurseInit.sis</li> <li>• PersonList.sis</li> </ul>

**オプション:** 同期サーバーを使用した同期化のために **Symbian** デバイスを構成するには:

アプリケーションに同期機能が備わっている場合は、同期サーバーの構成も行う必要があります。以下のファイルをデバイスにインストールします。

```
\Clients\Symbian7\sync\

```

デバイスで同期のテストを行うには、デバイスが接続ソフトウェアを使用してワークステーションに接続されていることを確認してください。アプリケーション・リストから ISync を選択します。「ファイル」→「設定値」ダイアログで、IP、ポート、ユーザー、およびパスワードを入力します。「ファイル」→「同期化」に移動して同期のテストを行います。

『Crystal セットアップ』に戻ります。

## MAB バージョン 8.1.4 WinCE のセットアップおよび構成

以下のタスクを行って、WinCE アプリケーションをビルドおよびテストするための Mobile Application Builder 環境をセットアップします。

**Mobile Application Builder バージョン 8.1.4 のセットアップおよび構成タスク:**

- 31 ページの『WinCE JVM プリファレンスのセットアップ』
- 33 ページの『デスクトップ上で WinCE または Sharp Zaurus アプリケーションをテストするためのセットアップ』
- 34 ページの『テストのための WinCE モバイル・デバイスのセットアップ』

---

## 関連タスク:

- 108 ページの『Mobile Application Builder アプリケーションのビルド』
- 113 ページの『Mobile Application Builder アプリケーションのテスト』

## WinCE JVM プリファレンスのセットアップ

Java 仮想マシン (JVM) が組み込まれておらず、デバイスまたはデバイス ROM にインストールされている WinCE/PocketPC デバイスの場合、MAB ビルド・アプリケーションは、以下のベンダーの Java 仮想マシンを使用してテストされています。

- IBM J9 (DB2 Everyplace SDK または WebSphere Studio Device Developer (WSDD) に組み込まれている)
- Insignia Corporation の Jeode PDA Edition バージョン 1.9
- Sun Microsystem の Personal Java Runtime Environment 1.0
- CrEme Plus JVM バージョン 3.21 (バーコード・スキャン・アプリケーションで必要とされる)

Mobile Application Builder は、JVM を持つ任意のデバイスにインストール可能なアプリケーションを生成します。アプリケーションをビルドする前に、使用する JVM に合わせて WinCE JVM プリファレンスを設定する必要があります。

Insignia、Sun、および CrEme Plus JVM のプリファレンスは MAB に組み込まれます。次の節に、以下の JVM の入手およびインストールについての情報が記載されています。

- IBM J9 Java 仮想マシン
- CrEme Plus JVM

WSDD から J9 を入手して、MAB で使用方法について、「*IBM DB2 Everyplace* インストールおよびユーザーズ・ガイド」を参照するか、または以下のステップを行ってください。

---

## IBM J9 Java 仮想マシンを WSDD から入手して、デバイスにインストールするには:

1. WSDD と、その関連製品を WSDD Web サイト からダウンロードします。
2. 以下のファイルまたはディレクトリーを WSDD インストールからデバイスにコピーします。下に記述されているディレクトリー構造を維持しないと、J9 実行可能ファイルは正しく機能しません。
  - <WSDD>\wsdd5.0\ive\lib\jclMax から取得した以下のファイルをデバイス上の \wsdd\lib\JCLMax ディレクトリーにインストールします。
    - classes.zip
    - profile.jar
  - <WSDD>\wsdd5.0\eclipse\plugins\com.ibm.ive.tomcatsupport\_5.0.0 から取得した以下のファイルをデバイス上の \wsdd\lib\JCLMax ディレクトリーにインストールします。
    - iverel20.dll

- j9.exe
- j9dyn20.dll
- j9max20.dll
- j9midp20.dll
- j9prt20.dll
- j9thr20.dll
- j9vm20.dll
- j9w.exe
- j9zlib20.dll
- swt-win32-2104.dll
- 英語以外のプロジェクトの場合、下に示されたディレクトリーから取得した以下のファイルをデバイス上の \wsdd\lib ディレクトリーにインストールします。
  - <WSDD>\wsdd5.0\ive\lib\charconv.zip
  - <WSDD>\wsdd5.0\ive\lib\jclMax\locale.zip

#### IBM J9 を使用するように Mobile Application Builder を構成するには:

IBM J9 を使用するには、プロジェクトの「WinCE JVM プリファレンス」および「データベース・ロケーション」プロパティを設定する必要があります。

1. Mobile Application Builder を開始します。初めて Mobile Application Builder 8.1.4 を開始した場合は、ウィンドウが開き、ツール・プリファレンスの設定を促す指示が出されます。プロジェクトをオープンまたは開始する必要はありません。
2. 「ファイル」→「プリファレンス」の順にクリックします。「プリファレンス」ウィンドウが開きます。
3. 「アプリケーション」→「WinCE JVM プリファレンス」をクリックします。
4. JVM のリストから「IBM J9 JVM」を選択します。
5. デフォルト設定を次の設定値に置き換えます。
  - 「JVM パス」フィールドに、\WSDD\bin\j9.exe を入力します。
  - 「引き数」フィールドに、次を入力します。

```
"-cp:\WSDD\lib\jclMax\charconv.zip;
\WSDD\lib\jclMax\classes.zip;
\WSDD\lib\jclMax\prsnlwin.jar;
\WSDD\lib\jclMax\database_enabler.jar;\X\X.jar;
\windows\db2ejdbc.jar;
" X.MABAppFrame
```

ここで、

- X はプロジェクト名です。
  - \WSDD は、J9 がインストールされているデバイス上のファイル・システムのロケーションです。
6. 「OK」をクリックします。「プリファレンス」ウィンドウが閉じます。
  7. 「プロジェクト」ペインで「WinCE」をクリックします。

8. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「データベース・ロケーション」プロパティを \ から \\ に変更します。

#### CrEme Plus JVM を入手するには:

バーコード・スキャンを使用する WinCE アプリケーションを開発する場合は、CrEme Plus JVM を使用します。

1. NSIcom Web サイトから CrEme Plus JVM をダウンロードします。
2. ダウンロード・ページの「**NSIcom Software**」フィールドで「CrEme Plus」を選択し、その他のオプションをターゲット・デバイスに応じて指定します。
3. すべてのダウンロード・ステップが完了したら、接続されたデバイスで setup.exe を実行してそのデバイスに JVM をインストールします。

#### CrEme Plus JVM を使用するように Mobile Application Builder を構成するには:

CrEme Plus JVM を使用するには、WinCE JVM プリファレンスを設定する必要があります。

1. Mobile Application Builder を開始します。初めて Mobile Application Builder 8.1.4 を開始した場合は、ウィンドウが開き、ツール・プリファレンスの設定を促す指示が出されます。
2. 新規の WinCE プロジェクトをオープンします。
3. 「ファイル」→「プリファレンス」の順にクリックします。「プリファレンス」ウィンドウが開きます。
4. 「アプリケーション」→「**WinCE JVM プリファレンス**」をクリックします。
5. JVM のリストから「**NSIcom — CrEme Plus JVM**」を選択します。
6. 「**JVM パスの入力**」フィールドのデフォルトを受け入れます。
7. 「**JVM 引き数の入力**」フィールドで、現行 WinCE プロジェクト用に設定されたデータベース・ロケーション・パス (デバイス上のデータベース表のロケーション) を、引き数のリストにスペースで区切って追加する必要があります。「プロジェクト」ペイン内の WinCE をクリックして、「プロパティおよびイベント」ペインの「プロパティ」ページにある「**データベース・ロケーション**」プロパティを表示することで、データベース・ロケーションを参照できます。  
たとえば、「データベース・ロケーション」プロパティが \windows の場合は、「**JVM 引き数の入力**」フィールドの最後のエントリーの後にスペースを入力し、次に \windows を入力します。
8. 「**OK**」をクリックします。「プリファレンス」ウィンドウが閉じます。

『WinCE セットアップ』に戻ります。

---

## デスクトップ上で WinCE または Sharp Zaurus アプリケーションをテストするためのセットアップ

WinCE/PocketPC アプリケーションは、Microsoft の WinCE エミュレーター (eMbedded Visual Tools 3.0 製品) ではテストできません。実際のデバイスを使用する必要があります。または、アプリケーションを Windows デスクトップ (Win32)

で実行することができます。また、以下の説明に従ってデスクトップに Sharp Zaurus アプリケーションをセットアップすることも可能です。

---

デスクトップ上での **WinCE** または **Sharp Zaurus** アプリケーション・テストをセットアップするには:

1. Mobile Application Builder の WinCE エミュレーション・プリファレンスを設定します。
  - a. Mobile Application Builder を開始します。初めて Mobile Application Builder 8.1.4 を開始した場合は、ウィンドウが開き、ツール・プリファレンスの設定を促す指示が出されます。プロジェクトをオープンまたは開始する必要はありません。
  - b. 「ファイル」 → 「プリファレンス」の順にクリックします。「プリファレンス」ウィンドウが開きます。
  - c. 「Win32 エミュレーション要件」をクリックします。
  - d. 設定が正しく、かつ DB2 Everyplace データベース・エンジンと表がインストールされている場所と一致していることを確認します。データベースのデフォルトの場所は次のとおりです。  
c:\DB2EveryplaceSDK 表のデフォルトの場所は次のとおりです。  
C:\DB2EveryplaceSDK\SDK\ApplicationBuilder  
\Projects\Samples\Win32\Tables
  - e. 「OK」をクリックします。「プリファレンス」ウィンドウが閉じます。

**オプション: WinCE サンプル・アプリケーションで使用する表を作成し、表にデータを追加するには:**

¥Projects¥Samples¥Win32 ディレクトリーの NurseInit.bat は、提供された SchemaInit.java プログラムを使用して、ASCII ファイルの行内容に基づく Win32 のデータベース表を作成し、移植するサンプル・スクリプトです。

1. NurseInit.bat を変更して、DB2 Everyplace がインストールされている場所へのパス情報に調整します。
2. NurseInit.bat を実行します。このファイルは、デスクトップで PersonList および VNApp サンプルが使用するスキーマをユーザー選択のディレクトリーに作成します。SchemaInit は、表がすでに存在していれば、それらをデータベースから除去し、ターゲット・ディレクトリーが存在していなければ、それも作成しません。

『Crystal セットアップ』に戻ります。

---

## テストのための WinCE モバイル・デバイスのセットアップ

WinCE デバイスでテストを行うには、DB2 Everyplace データベース・エンジン、JDBC ドライバー、および Java 仮想計算機 (JVM) をデバイスにインストールする必要があります。

---

テストのために **WinCE** デバイスをセットアップするには:

1. 使用するデバイスのプロセッサ・タイプと WinCE バージョンに適合する DB2 Everyplace データベースおよび JDBC ファイルを  
¥Clients¥Win32¥database¥x86 ディレクトリーからインストールします。必要なファイルの説明については、「*IBM DB2 Everyplace インストールおよびユーザーズ・ガイド*」を参照してください。
2. 適切な JVM for WinCE デバイスを取得して、インストールします。MAB 生成アプリケーションは、以下のソースからの JVM を使用してテストされています。
  - iPAQ 3630 などの ARM プロセッサの場合は、Insignia solutions (<http://www.insignia.com>)
  - Compaq Aero や HP Jornada などの MIPS および SH3 プロセッサの場合は、Sun Microsystems (<http://www.java.sun.com>)
3. JVM の手順に従って、JVM をデバイスにインストールします。インストールの実行中にパスの選択を求められた場合は、デフォルト・パスを選択してください。

『Crystal セットアップ』に戻ります。

---

## MAB バージョン 8.1.4 Sharp Zaurus のセットアップおよび構成

以下のタスクを行って、Sharp Zaurus アプリケーションをビルドおよびテストするための Mobile Application Builder 環境をセットアップします。

---

**Mobile Application Builder バージョン 8.1.4 Zaurus のセットアップおよび構成タスク:**

- 『Sharp Zaurus ビルド環境のセットアップ』
  - 36 ページの『デスクトップ上で Sharp Zaurus アプリケーションをテストするためのセットアップ』
  - 37 ページの『テストのための Sharp Zaurus デバイスのセットアップ』
- 

**関連タスク:**

108 ページの『Mobile Application Builder アプリケーションのビルド』

113 ページの『Mobile Application Builder アプリケーションのテスト』

### Sharp Zaurus ビルド環境のセットアップ

Sharp Zaurus 用のアプリケーションをビルドする前に、Cygwin をインストールして、Cygwin \bin ディレクトリーを Windows 環境ユーザー変数 PATH に追加す

る必要があります。Sharp Zaurus 用の Cygwin インストールは、Palm 用の Cygwin インストールと同じです。ただし Zaurus の場合、Palm の場合のように Mobile Application Builder を使用してビルド・プリファレンスを設定するのではなく、\bin ディレクトリーを手動で PATH 変数に追加する必要があります。

Sharp Zaurus アプリケーション・ビルドで使用される Cygwin ユーティリティーは tar.exe と gzip.exe です。これらのユーティリティーの他のバージョンも使用可能ですが、Cygwin バージョンのみテスト済みです。これらのユーティリティーの他のバージョンを使用する場合、下で説明されているように、これらを PATH 変数に追加する必要があります。

---

### Sharp Zaurus ビルド環境をセットアップするには:

1. 『Palm C コード・ビルド環境のセットアップ』の節で説明されているとおりに、Cygwin をインストールします。
2. <cygwin installation directory>\cygwin-b20\h-i586-cygwin32\bin を、下記の方法で Windows 2000 環境ユーザー変数 PATH の先頭に追加します。他の Windows オペレーティング・システムでも、同様の方法での追加が必要となります。Windows 2000 を使用しない場合、特定の説明については Windows ヘルプを参照してください。
  - a. 「スタート」→「設定」→「制御 パネル」に移動します。
  - b. 「システム」アイコンをダブルクリックして、システムのプロパティーを変更します。
  - c. 「詳細」タブをクリックします。
  - d. 「環境変数」ボタンをクリックします。
  - e. 現在ログインしているユーザーでビルダーを使用するか、または任意のユーザーでビルダーを使用するかに応じて、ユーザー環境変数またはシステム環境変数の下の PATH 変数を編集します。ユーザー環境変数については、PATH がまだ定義されていなければ「新規」ボタンを使用して PATH 変数を追加する必要があります。

35 ページの『MAB バージョン 8.1.4 Sharp Zaurus のセットアップおよび構成』

---

### デスクトップ上で Sharp Zaurus アプリケーションをテストするためのセットアップ

Sharp Zaurus アプリケーションは、エミュレーターでテストすることはできません。実際のデバイスを使用する必要があります。または、アプリケーションを Windows デスクトップ (Win32) で実行することができます。『デスクトップ上で WinCE アプリケーションをテストするためのセットアップ』の節の説明に従って (ただし、WinCE サンプル・アプリケーションに関連したオプション・ステップは除く)、Win32 デスクトップ・シミュレーションを使用して Zaurus アプリケーションをテストします。

35 ページの『MAB バージョン 8.1.4 Sharp Zaurus のセットアップおよび構成』

---

## テストのための Sharp Zaurus デバイスのセットアップ

本節では、DB2 Everyplace のライブラリーおよびサンプル・アプリケーションを Sharp Zaurus モバイル・デバイスにインストールする方法について説明します。インストールを行う前に、以下のことを行っておく必要があります。

- Windows ワークステーションで、Sharp Zaurus モバイル・デバイスに付属している接続ソフトウェアのインストールと構成を行います。
- Sharp Zaurus モバイル・デバイスをワークステーションに接続します。モバイル・デバイスの資料を参照して、デバイスが正しく接続されているかを確認してください。

---

### テストのために Sharp Zaurus デバイスをセットアップするには:

DB2 Everyplace ライブラリーおよびサンプル・アプリケーション・ファイルを手動でインストールするには、Sharp Zaurus モバイル・デバイスに付属している接続ソフトウェアのインストール・ツールを使用します。ファイル

`\Clients\embeddedLinux\database\install\db2e-libs_8.1.2_arm.ipk` をインストールします。このファイルによって、以下のアイテムがデバイス上の `/usr/bin` ディレクトリーにインストールされます。

- すべての必須 DB2 Everyplace データベース・エンジン・バイナリー
- すべての必須同期バイナリー
- `db2ejdbc.jar`
- `isync4j.jar`

### オプション: NurseInit およびその他のサンプル Zaurus アプリケーションをインストールするには:

Zaurus サンプル・ファイルの場所は、177 ページの『DB2 Everyplace のサンプル・アプリケーション』に記されています。

1. 以下のファイルをデバイスにインストールします。

表 8. デバイス用の Sharp Zaurus サンプル・アプリケーション

ファイル名	説明
PersonList_1.0_arm.ipk	PersonList サンプル・アプリケーション
NurseInit_1.0_arm.ipk	NurseInit サンプル・アプリケーション

2. NurseInit サンプル・アプリケーションをインストールした後に、これをデバイス上で実行します。このアプリケーションは、VNSCHEDULE、VNPERSO、VNMEDICALRECORD、VNCONTACT、および VNSIGNATURE 表を作成して、これらの表にレコードを挿入し、デバイス上の `/home/db2e/database` に置きます。
3. Mobile Application Builder に戻って、「ファイル」→「プリファレンス」をクリックし、「プリファレンス」ウィンドウをオープンします。

## Mobile Application Builder のアンインストール

Mobile Application Builder をインストールする前に、以前の版をアンインストールします。

---

Mobile Application Builder をアンインストールするには:

1. Windows デスクトップから、「スタート」->「設定」->「制御 パネル」->「アプリケーションの追加と削除」に移動します。
  2. インストールされているプログラムのリストから **IBM DB2 Everyplace Software Development Kit** を選択します。
  3. 「変更/削除」をクリックして製品をアンインストールします。
  4. 製品を除去したいかどうかを尋ねてきたら、「はい」をクリックします。
  5. アンインストールが完了したら「OK」をクリックし、「アプリケーションの追加と削除」ウィンドウを閉じます。
- 

関連タスク:

15 ページの『MAB バージョン 8.1.4 のセットアップおよび構成』

---

## MAB プロジェクト

MAB のプロジェクトには、アプリケーションをビルドするための制御およびリソースをすべて保管して、開発に合わせて編成するために使用できるフレームワークが含まれています。

---

関連タスク:

50 ページの『アプリケーション ID』

## ウェルカム・ダイアログからのプロジェクトのオープン

Mobile Application Builder を始動するたびに、ウェルカム・ダイアログが表示されます。このウェルカム・ダイアログを開始点として、新しいプロジェクトを作成したり、既存のプロジェクトを使用したり、最後にオープンしたプロジェクトの作業を続けたりすることができます。

---

ウェルカム・ダイアログからプロジェクトをオープンするには:

ウェルカム・ダイアログで、以下を行います。

1. オープンしたいプロジェクトのタイプを指定します。
  - 新規プロジェクトを作成できるウィンドウをオープンする場合は、「**新規プロジェクトを開始**」を選択します。
  - 前のセッションで作成したプロジェクトを処理する場合は、「**既存のプロジェクトをオープン**」を選択します。
  - 前回のプロジェクトをオープンするには、「**最後に作業したプロジェクトを続行 (プロジェクト名)**」を選択します。このオプションを使用できるのは、Mobile Application Builder が最後のプロジェクトを検出した場合のみです。
2. オプション: 次回 Mobile Application Builder を始動したときにこのウィンドウをオープンしたくない場合は、「このウィンドウを始動時に表示」チェック・ボックスからチェックマークを外します。Mobile Application Builder 内からウィンドウをリストアするには、「**ファイル**」→「**設定**」の順にクリックし、「**一般**」をクリックし、「**始動時にウェルカム・ダイアログを表示**」チェック・ボックスにチェックマークを付けます。
3. 「**OK**」をクリックします。ウェルカム・ダイアログがクローズし、選択が適用されます。

---

#### 関連タスク:

『新規プロジェクトの作成』

40 ページの『既存のプロジェクトのオープン』

## 新規プロジェクトの作成

Mobile Application Builder のプロジェクトには、アプリケーションをビルドするための制御とリソースをすべて保管して、開発に合わせて編成することができるフレームワークが含まれています。

プロジェクトを作成する場合は、初期プロジェクト情報を入力しなければなりません。

---

#### 新規プロジェクトを作成するには:

1. 次のいずれかの方法に従って、「新規プロジェクト」ウィンドウを開きます。
  - 「**ファイル**」メニューから  「**新規プロジェクト**」をクリックします。
  - ツールバーから  をクリックします。
2. 「**プロジェクト名**」フィールドに、プロジェクトの名前を入力します。命名規則と制限については、41 ページの『プロジェクトとアプリケーションの命名規則』を参照してください。この名前は、プロジェクト・ディレクトリーの一部として使用されます。プロジェクト名は後で編集できます。
3. 「**プロジェクト・ディレクトリー**」フィールドに、プロジェクトを保管するディレクトリーを指定するか、デフォルトの `..%ApplicationBuilder%Projects` ディ

レクトリーを使用します。このフィールドに指定したディレクトリーの下に、プロジェクトと同じ名前のディレクトリーが作成されます。これが、プロジェクト・ファイルの保管場所になります。命名規則と制限については、41 ページの『プロジェクトとアプリケーションの命名規則』を参照してください。

4. 「**アプリケーション名**」フィールドに、アプリケーションの名前を入力します。命名規則と制限については、41 ページの『プロジェクトとアプリケーションの命名規則』を参照してください。この名前は、ターゲット・デバイスのアプリケーション・ピッカー画面に、アプリケーションの名前として表示されます。
5. 「**ターゲット・デバイス**」フィールドで、アプリケーションを実行するモバイル・デバイスを選択します。「**ターゲット・デバイス**」フィールドの下のフィールドのオプションは、選択したモバイル・デバイスによって異なります。
6. 「**アプリケーション ID**」フィールドに、アプリケーションの ID を入力します。「**アプリケーション ID**」フィールドがない場合は、ターゲットのプラットフォームにアプリケーション ID は必要ありません。詳細については、50 ページの『アプリケーション ID』を参照してください。
7. 「**フォーム・サイズ**」フィールドで、アプリケーションのデフォルトのフォーム・サイズを指定します。このプロジェクトの新規フォームはすべて、このデフォルト・サイズで作成されます。デフォルトのフォーム・サイズは、プロジェクト内の個々のフォームごとに変更できます。このフィールドを編集できるのは、「**ターゲット・デバイス**」フィールドで「汎用 Java」を選択した場合のみです。汎用 Java アプリケーションの場合は、デフォルト・サイズ (1/4 VGA) を選択するか、フォームの幅と高さを指定できます。
8. 「**完了**」をクリックします。「プロジェクト」ペインにプロジェクトが表示されます。「完了」が使用できない場合は、プロジェクト名の先頭または末尾にスペースが入っていないこと、または「**名前**」および「**アプリケーション名**」フィールドが他の命名要件に違反していないことを確認してください。

---

#### 関連タスク:

『既存のプロジェクトのオープン』

38 ページの『ウェルカム・ダイアログからのプロジェクトのオープン』

42 ページの『プロジェクトまたはアプリケーションの名前変更』

## 既存のプロジェクトのオープン

Mobile Application Builder のプロジェクトには、アプリケーションをビルドするための制御とリソースをすべて保管して、開発に合わせて編成することができるフレームワークが含まれています。

---

既存のプロジェクトをオープンするには、次の方法のいずれかを使用します:

-

1. 「ファイル」 →  「プロジェクトのオープン」の順にクリックするか、ツールバーから  をクリックします。
  2. オープンしたいプロジェクトを探して、「オープン」をクリックします。
- メインメニューから「ファイル」 →  「プロジェクト名」の順にクリックします。「ファイル」メニューには、最後にオープンしたプロジェクトの項目が入ります。

#### 関連タスク:

39 ページの『新規プロジェクトの作成』

38 ページの『ウェルカム・ダイアログからのプロジェクトのオープン』

42 ページの『プロジェクトまたはアプリケーションの名前変更』

## プロジェクトとアプリケーションの命名規則

プロジェクト、プロジェクト・ディレクトリー、およびアプリケーションの名前には、いくつかの制限があります。

この表は、プロジェクトの命名規則と制限を示したものです。

表 9. プロジェクトの命名規則および制限

すべてのプラットフォーム	プロジェクト名には SBBCS/ASCII 文字のみを使用します。プロジェクト名には DBCS 文字を使用しないでください。プロジェクト名を NULL にすることはできません。プロジェクト名をスペースやピリオドで開始または終了することはできません。
Java コード・プラットフォーム	プロジェクト名にスペースを含めることはできません。

この表は、プロジェクト・ディレクトリーの命名規則と制限を示したものです。

表 10. プロジェクト・ディレクトリーの命名規則および制限

すべてのプラットフォーム	プロジェクト・ディレクトリー名を NULL にすることはできません。プロジェクト・ディレクトリーをスラッシュ ( ¥ ) で終わらせることはできません。
Symbian プラットフォーム	すべてのプラットフォームに対する制限に加えて: プロジェクト・ディレクトリーに DBCS 文字またはスペースを含めることはできません。

この表は、アプリケーションの命名規則と制限を示したものです。

表 11. アプリケーションの命名規則および制限

すべてのプラットフォーム	アプリケーション名を NULL にすることはできません。アプリケーション名をスペースやピリオドで開始または終了することはできません。
Symbian プラットフォーム	すべてのプラットフォームに対する制限に加えて: アプリケーション名に DBCS 文字を使用しないでください。他のプラットフォームのアプリケーション名には DBCS 文字を使用することができます。アプリケーション名にスペースまたはピリオドを含めることはできません。また、アプリケーション名を単一の文字で構成することはできません。

---

#### 関連タスク:

39 ページの『新規プロジェクトの作成』  
『プロジェクトまたはアプリケーションの名前変更』

#### 関連参照:

170 ページの『トラブルシューティング』

## プロジェクトまたはアプリケーションの名前変更

現在処理している Mobile Application Builder プロジェクトのプロジェクト名またはアプリケーション名を変更することができます。

---

#### プロジェクトまたはアプリケーションを名前変更するには:

1. 「プロジェクト」ペインで「プロジェクト名」をクリックします。
  2. 「プロパティおよびイベント」ペインの「名前」フィールドまたは「アプリケーション名」フィールドをクリックします。ウィンドウがオープンし、そこにプロジェクトまたはアプリケーションの新規名を指定することができます。
  3. 「新規名」フィールドに、現行ターゲット・プラットフォームの命名要件に準拠する、プロジェクトまたはアプリケーションの新規名を入力します。命名要件の詳細については、41 ページの『プロジェクトとアプリケーションの命名規則』を参照してください。
  4. 「OK」をクリックします。ウィンドウがクローズし、新規名が「名前」フィールドまたは「アプリケーション名」フィールドに表示されます。「OK」が使用可能になっていない場合は、入力した名前が現行ターゲット・プラットフォームの要件を満たしているかを確認してください。
-

#### 関連タスク:

39 ページの『新規プロジェクトの作成』

38 ページの『ウェルカム・ダイアログからのプロジェクトのオープン』

## プロジェクト・プロパティの構成

プロジェクト・プロパティを構成するには、プロジェクトをオープンする必要があります。プロジェクト・プロパティは、ターゲット・プラットフォームによって異なります。

多くのプロジェクト・プロパティには、あらかじめデフォルト値が設定されています。

---

#### プロジェクト・プロパティを構成するには:

1. 「プロジェクト」ペインで <ターゲット OS> をクリックします。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページでプロジェクト・プロパティを変更します。それぞれのプロジェクト・プロパティの詳細い説明については、128 ページの『プロジェクト・プロパティ』を参照してください。

---

#### 関連参照情報:

128 ページの『プロジェクト・プロパティ』

## プロジェクトへのソース・ファイルの追加

ソース・ファイル (.c および .h ファイル) を使用して、Palm プロジェクトの一部としてコンパイルします。ソース・ファイルは、どの Palm プロジェクトにも追加できます。使用するソース・ファイルを追加、編集、除去、再配列するには、「ソースおよびヘッダー・ファイル」ウィンドウを使用します。

ソース・ファイルのサポートは、ターゲット・プラットフォームによって異なります。

---

#### プロジェクトにソース・ファイルを追加するには:

1. 「プロジェクト」ペインで「**グローバル定義**」をクリックします。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「ソース・ファイル」の隣にある  をクリックし、「ソースおよびヘッダー・ファイル」ウィンドウを開きます。
3. 「**追加**」をクリックして、組み込むソース・ファイルを選択します。
4. オープンしたウィンドウの「**ファイル**」フィールドで、ソース・ファイル名を入力するか、または探して、「**OK**」をクリックします。

5. 「OK」をクリックします。「ソースおよびヘッダー・ファイル」ウィンドウが閉じます。

---

#### 関連タスク:

『プロジェクトへのライブラリー・ファイルの追加』

## プロジェクトへのライブラリー・ファイルの追加

ライブラリー・ファイルを使用して、共用ライブラリーをアプリケーションに追加します。共用ライブラリーによって、アプリケーションにより多くの機構や機能を提供できます。「共用ライブラリー・ファイル」ウィンドウまたは「Java ライブラリー・ファイル」ウィンドウを使用して、使用するライブラリー・ファイルを追加、編集、除去、または再配列します。ライブラリー・ファイルと共に提供される文書で、共用ライブラリーを定義するための情報を探してください。

ライブラリー・ファイルのサポートは、ターゲット・プラットフォームによって異なります。

---

#### ライブラリー・ファイルを Palm C コード・プロジェクトに追加するには:

1. 「プロジェクト」ペインで「グローバル定義」をクリックします。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「ライブラリー・ファイル」の隣にある  をクリックし、「共用ライブラリー・ファイル」ウィンドウを開きます。
3. 「追加」をクリックして、組み込む共用ライブラリー・ファイルを定義します。
4. オープンしたウィンドウに、次の情報を入力します。
  - ライブラリー名に対するポインター (CharPtr)
  - ライブラリー参照を保管するためのグローバル変数 (UIntPtr)
  - ライブラリーのタイプ (DWord)
  - ライブラリーの作成者 (DWord)
5. 「OK」をクリックします。
6. 「OK」をもう 1 度クリックして、「共用ライブラリー・ファイル」ウィンドウを閉じます。

#### ライブラリー・ファイルを Java コード・プラットフォーム・プロジェクトに追加するには:

jar ファイルを Java プラットフォーム・プロジェクトに追加することができます。また、これらのファイルをビルドに組み込み、アプリケーションと一緒にデバイスにインストールするように指定することもできます。ただし、これらの jar ファイルによって使用される他のファイル (たとえば、.dll ファイル) は、デバイスまたはエミュレーターに別にインストールする必要があります。

1. 「プロジェクト」ペインで「グローバル定義」をクリックします。

2. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「ライブラリー・ファイル」の隣りにある  をクリックし、「Java ライブラリー・ファイル」ウィンドウを開きます。
3. 「追加」をクリックして、組み込む共用ライブラリー・ファイルを定義します。
4. オープンしたウィンドウの「ファイル」フィールドで、ファイル名を入力するか、またはブラウズします。jar ファイルのみを追加できます。
5. オプション: 「インストール可能ファイルにファイルを組み込むための確認」チェック・ボックスにチェックマークを付けて、アプリケーションのデバイスへのインストール時にこのファイルがデバイスにインストールされるようにします。
6. 「OK」をクリックします。
7. 「OK」をもう 1 度クリックして、「Java ライブラリー・ファイル」ウィンドウを閉じます。

---

#### 関連タスク:

44 ページの『プロジェクトへのライブラリー・ファイルの追加』

## プロジェクトの保管とクローズ

終了後プロジェクトを保管してクローズします。クローズしなくても、プロジェクトを保管することもできます。

---

#### プロジェクトを保管してクローズするには:

1. 「プロジェクト」ペインで「プロジェクト名」をクリックします。
2. 「ファイル」→「プロジェクトのクローズ」の順にクリックします。プロジェクトに変更を加えた場合は、ウィンドウがオープンし、変更の保管を求めるプロンプトが表示されます。
3. 「はい」をクリックして、変更を保管し、プロジェクトをクローズします。

#### プロジェクトをクローズしないで保管するには:

1. ツールバーの  をクリックするか、メインメニューから「ファイル」→  「プロジェクトの保管」の順にクリックします。

---

#### 関連タスク:

39 ページの『新規プロジェクトの作成』

40 ページの『既存のプロジェクトのオープン』

---

## プリファレンスの設定

Mobile Application Builder のコピーのための、アプリケーション設定をカスタマイズすることができます。

プリファレンス・オプションは、ターゲット・プラットフォームによって異なります。

---

### 関連概念:

157 ページの『スクリプト記述サポート』

### 関連タスク:

108 ページの『Mobile Application Builder アプリケーションのビルド』

101 ページの『アプリケーションへの印刷機能の追加』

38 ページの『MAB プロジェクト』

87 ページの『Mobile Application Builder 中のスクリプト』

15 ページの『MAB バージョン 8.1.4 のセットアップおよび構成』

## 一般プリファレンスの設定

ご使用の Mobile Application Builder の、一般設定をカスタマイズすることができます。

---

### 一般プリファレンスを設定するには:

1. 「ファイル」 → 「プリファレンス」の順にクリックします。「プリファレンス」ウィンドウが開きます。
2. 「その他」ノードをクリックして、一般プリファレンスを表示します。
3. オプション: Mobile Application Builder の始動時に「新規アプリケーション」ウィザードを自動的に立ち上げるには、「始動時にウェルカム・ダイアログを表示」を選択します。
4. オプション: 「ブラウザーの位置を指定してオンライン・ヘルプを表示します。(全パスおよびファイル名)」フィールドで、表示するデフォルト・ブラウザーを選択します。ご使用のブラウザーの実行可能ファイルを見つけるには、「ブラウズ」ボタンをクリックします。
5. 「OK」をクリックします。「プリファレンス」ウィンドウが閉じます。

---

### 関連タスク:

『プリファレンスの設定』

## ビルド・プリファレンスの設定

ご使用の Mobile Application Builder の、ビルド・プリファレンスをカスタマイズすることができます。

アプリケーションにデバッグ・シンボルを追加することで、アプリケーションをテストし、GDB を使用してデバッグすることができます。

Palm OS には、セグメントごとに 32K のコード・セグメント制限があります。単一コード・セグメントを使用しているアプリケーションが 32K を超えた場合、オーバーフロー・エラーが返され、アプリケーションはコンパイルされません。デフォルトでは、複数コード・セグメントが使用され、この制限を回避します。複数コード・セグメントでは、各セグメントが 32K 未満になるようにコードを全体的にフォームごとに分離します。

小規模のアプリケーションを作成してより効率的にしたい場合は、複数コード・セグメントを使用不可にしてください。CodeWarrior を使用するアプリケーションを作成する前に、複数コード・セグメントを使用不可にすることをお勧めします。

---

### ビルド・プリファレンスを設定するには:

1. 「ファイル」→「プリファレンス」の順にクリックします。「プリファレンス」ウィンドウが開きます。
2. 「ビルド」ノードをクリックすると、ビルド・プリファレンスが表示されます。
3. オプション: 「ビルド・モード」の下のオプションを選択します。
  - ビルド時に、アプリケーションにデバッグ・シンボルを追加するためには、「デバッグ」を選択します。このオプションを選択すると、ビルド時に中間ファイルがプロジェクト・ディレクトリーに保管され、アプリケーションをデバッグする際に役立ちます。
  - デバッグ・シンボルを組み込まないで、アプリケーションのパフォーマンスを最適化するためには、「リリース」を選択します。このオプションを選択すると、ビルドが正常に終了した後は、中間ファイルは保管されません。
4. オプション: 「複数コード・セグメントでコンパイル」チェック・ボックスをクリックすると、小さなアプリケーションの効率を若干上げることができます。
5. オプション: Palm 5 ウェイ・ナビゲーション・ボタンのコード生成を使用可能にするには、「5 ウェイ・ナビゲーションを使用可能にする (Enable Five Way Navigation)」チェック・ボックスを選択します。詳細については、67 ページの『Palm 5 ウェイ・ナビゲーションを使用可能にする』を参照してください。
6. オプション: 「ビルドのための印刷ライブラリーの選択」リストから印刷ソリューションを選択します。
7. 「OK」をクリックします。「プリファレンス」ウィンドウが閉じます。

---

### 関連タスク:

46 ページの『プリファレンスの設定』

## アプリケーション・プリファレンスの設定

Mobile Application Builder のコピーのための、アプリケーション設定をカスタマイズすることができます。

---

アプリケーション・プリファレンスを設定するには:

1. 「ファイル」→「プリファレンス」の順にクリックします。「プリファレンス」ウィンドウが開きます。
2. 「アプリケーション」ノードを拡張すると、アプリケーション・プリファレンスが表示されます。
3. オプション: 「メッセージ」をクリックして、データベース・アクションの前または後に確認または応答のメッセージ・ダイアログを組み込むかどうかを選択します。デフォルトのメッセージを選択するか、ユーザー独自のメッセージをカスタマイズしてください。
4. オプション: 「UI カラー」をクリックして、オブジェクト (ボタン、チェック・ボックスなど)、メニュー、フィールド、フォーム、ダイアログ、およびアラートなどの UI エLEMENTのカラーを選択し、アプリケーションをカスタマイズします。UI カラー・オプションのサポートは、ターゲット・プラットフォームによって異なります。
5. オプション: 「フォント」をクリックし、フォント・オプションを選択してアプリケーションをカスタマイズします。フォント・オプションのサポートは、ターゲット・プラットフォームによって異なります。
6. オプション: 「WinCE JVM プリファレンス」をクリックして、WinCE アプリケーションをビルドするためのプリファレンスを設定します。
7. 「OK」をクリックします。「プリファレンス」ウィンドウが閉じます。

---

関連タスク:

46 ページの『プリファレンスの設定』

## スクリプト・プリファレンスの設定

「スクリプト」ページでは、スクリプト・ウィンドウをカスタマイズし、読み取りおよび書き込みをより容易にすることができます。ここでの変更は、スクリプト・ウィンドウで表示されるテキストにのみ影響し、モバイル・デバイスには影響しません。

---

スクリプト・プリファレンスを設定するには:

1. 「ファイル」→「プリファレンス」の順にクリックします。「プリファレンス」ウィンドウが開きます。
2. 「スクリプト」ノードを拡張すると、スクリプト・プリファレンスが表示されます。

3. 「スクリプト」ノードをクリックして、スクリプト・エディター・エレメントの動作を設定します。
4. オプション: 「字下げ」の下のオプションを選択して、スクリプト・ウィンドウ内で使用される字下げのタイプをカスタマイズします。
5. オプション: 「検索および置換」の下のオプションを選択して、検索および置換の動作をカスタマイズします。
6. オプション: 「**Mobile Application Builder** の生成済み関数をプロジェクト・ビュー内に組み込む」を選択して、「プロジェクト」ペイン内に MAB 関数と一緒に、ユーザーが生成した関数も表示します。
7. 「テキスト・スタイル」ノードをクリックして、スクリプト・ウィンドウ内のテキストのフォントの外観およびサイズをカスタマイズします。
8. オプション: 「フォント」リストからフォント・タイプを選択します。
9. オプション: 「サイズ」リストからフォント・サイズを選択します。
10. オプション: 異なるタイプのテキストを異なる色で表示するには、「**構文にカラー強調表示を使用**」を選択してから、「基本テキスト」、「定数」、「キーワード」、「コメント」、および「エラー」の色を選択します。

---

#### 関連概念:

157 ページの『スクリプト記述サポート』

#### 関連タスク:

46 ページの『プリファレンスの設定』

## 印刷プリファレンスの設定

ご使用の Mobile Application Builder の、印刷プリファレンスをカスタマイズすることができます。

印刷機能のサポートは、ターゲット・プラットフォームによって異なります。

---

#### 印刷プリファレンスを設定するには:

1. 「ファイル」→「プリファレンス」の順をクリックします。「プリファレンス」ウィンドウが開きます。
2. 「ビルド」ノードをクリックすると、ビルド・プリファレンスが表示されます。
3. 「ビルドのための印刷ライブラリーの選択」リストから印刷ソリューションを選択します。
4. 「OK」をクリックします。「プリファレンス」ウィンドウが閉じます。

---

#### 関連タスク:

- 46 ページの『プリファレンスの設定』
- 101 ページの『アプリケーションへの印刷機能の追加』

#### 関連参照情報:

- 158 ページの『印刷サポート』
- 159 ページの『印刷 API』

## 前提条件ツールのためのプリファレンスの設定

Mobile Application Builder を初めてセットアップする場合、前提条件ツールのパス設定を示す必要があります。パス設定を示す必要があるのは、対象となるプラットフォームのためのパス設定のみです。前提条件ツールは、任意のディレクトリーにインストールできます。

---

#### 前提条件ツール・プリファレンスを設定するには:

1. 「ファイル」→「プリファレンス」の順にクリックします。「プリファレンス」ウィンドウが開きます。
2. オプション: 「**Palm ツール**」をクリックすると、Palm ツールのプリファレンスが表示されます。各ツールごとに、そのツールをインストールしたルート・ディレクトリーへのパスを指定します。
3. オプション: 「**Symbian ツール**」をクリックすると、Symbian ツールのプリファレンスが表示されます。Symbian SDK がインストールされているディレクトリーの絶対パス (ドライブを含む) を指定します。たとえば、c:\temp のように指定します。
4. オプション: 「**WinCE エミュレーション・パス要件**」をクリックすると、WinCE のプリファレンスが表示されます。DB2 Everyplace データベース・エンジンをインストールしたルート・ディレクトリーへのパスを指定し、また、デスクトップ上の Win 32 データベース表へのパスも指定します。
5. 「**OK**」をクリックします。「プリファレンス」ウィンドウが閉じます。

---

#### 関連タスク:

- 15 ページの『MAB バージョン 8.1.4 のセットアップおよび構成』
- 46 ページの『プリファレンスの設定』

---

## アプリケーション ID

モバイル・デバイスのオペレーティング・システムの中には、システムで実行されている各アプリケーションに固有 ID を使用するものがあります。デバイス上のアプリケーション間で競合が発生することがないように、これらのモバイル・デバイスのオペレーティング・システムは、こうしたアプリケーション ID の公式リストを保持しています。

この要件が適用されるデバイスをターゲットとしている場合は、登録済みのアプリケーション ID を入手する必要があります。こうすれば、同じ ID を持つ同じアプリケーションを、問題なくデバイスまたはエミュレーターにダウンロードすることができます。問題が発生した場合は、デバイスまたはエミュレーター上のアプリケーションを削除してから、再度ダウンロードしてください。

---

#### 関連タスク:

39 ページの『新規プロジェクトの作成』

## アプリケーション ID の登録

モバイル・デバイスのオペレーティング・システムの中には、システムで実行されている各アプリケーションに固有 ID を使用するものがあります。デバイス上のアプリケーション間で競合が発生することがないように、これらのモバイル・デバイスのオペレーティング・システムは、こうしたアプリケーション ID の公式リストを保持しています。

---

#### Palm アプリケーション ID を登録するには:

1. Palm OS アプリケーション ID データベースを検索し、使用したい ID がまだ取得されていないことを確認します。
2. オンライン登録フォームを使用して、アプリケーション ID を予約します。

#### SymbianOS UID 番号を登録するには:

- Symbian で 1 つ以上の UID を要求する場合は E メールを送信しなければなりません。UID の割り当ては Symbian が行っています。
  - 詳細については、Symbian の Web サイトを参照してください。
- 

#### 関連タスク:

『アプリケーション ID の設定または変更』

## アプリケーション ID の設定または変更

このタスクは、登録済みのアプリケーション ID が必要であるデバイスをターゲットとしている場合のみ必要となります。この要件が適用されるデバイスをターゲットとしている場合には、登録済みのアプリケーション ID を入手しなければ、アプリケーションを導入することはできません。

Mobile Application Builder はアプリケーション ID を自動的に生成します。ターゲット・プラットフォームが、登録済みのアプリケーション ID を必要としている場合は、次のステップを実行して、アプリケーション ID を登録済みの ID に変更してください。

---

アプリケーション ID を設定または変更するには:

1. 「プロジェクト」ペインで <モバイル・デバイス OS> をクリックします。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページ上の「アプリケーション ID」フィールドを編集します。「プロパティ」ページに「アプリケーション ID」フィールドがない場合は、ターゲット・デバイスに、登録済みのアプリケーション ID は必要ありません。

---

関連タスク:

51 ページの『アプリケーション ID の登録』

---

## 表およびデータ・ソース

アプリケーションを開発する場合は、表定義ファイル (\*.ddl) が必要です。表定義ファイルには、CREATE TABLE ステートメントが入っていないとなりません。CREATE TABLE ステートメントの詳細については、DB2 Everyplace エンジンと一緒にインストールされる、「*IBM DB2 Everyplace アプリケーション開発ガイド*」を参照してください。

---

関連例:

179 ページの『CREATE TABLE ステートメントの例』

179 ページの『データベース照会の定義』

## 表定義ファイルの作成およびインポート

アプリケーションに収集した情報を、正しく表という形式にするためには、表定義をプロジェクトにインポートしなければなりません。表定義 (\*.ddl) をインポートすると、プロジェクトとそのリソースに、表作成のベースとして使用できる列とデータ・タイプが提供されます。

表定義ファイル (\*.ddl) をプロジェクトにインポートした後で変更した場合は、その変更をプロジェクトで有効にするために、表定義ファイルをいったん除去してから再びインポートする必要があります。表を除去してから再びインポートしたら、その表に対する既存のすべての参照をリセットしなければなりません。

DB2 Universal Database (UDB) などの別のデータベース上にある表を使用している場合も、定義を生成してファイルにすることができる場合があります。DB2 UDB の場合は、Control Center で表に対し「**DDL の生成**」メニュー機能を使用します。ファイルを編集して小さい変更を加えなければならないこともありますが、.ddl を最初から作り直すよりもこの方が正確です。

表定義ファイルを作成してインポートするには:

1. テキスト・エディターを使用して、CREATE TABLE ステートメントとフィールド定義を含む \*.ddl ファイルを作成します。CREATE TABLE ステートメントの詳細については、DB2 Everyplace エンジンと一緒にインストールされる、「*IBM DB2 Everyplace アプリケーション開発ガイド*」を参照してください。
2. 「プロジェクト」ペインで、 「表」を右マウス・ボタン・クリックします。
3. ポップアップ・メニューから「**表のインポート**」をクリックします。
4. 適切な .ddl ファイルを探して、「**オープン**」をクリックします。

---

関連タスク:

- 『表定義ファイルの更新』
- 54 ページの『表定義ファイルの削除』
- 57 ページの『新規フォームの作成』

関連例:

- 179 ページの『CREATE TABLE ステートメントの例』
- 179 ページの『データベース照会の定義』

## 表定義ファイルの更新

必要に応じて、表定義ファイルを更新します。変更を加えたら、表定義ファイルを再度インポートし、プロジェクトでその変更を有効にします。

表を除去してから再びインポートしたら、その表に対する既存のすべての参照をリセットしなければなりません。

---

プロジェクトにすでにインポート済みの表定義ファイルを更新するには:

1. 「プロジェクト」ペインで表の名前を右マウス・ボタン・クリックして、「**削除**」をクリックします。
2. 「プロジェクト」ペインで、 「表」を右マウス・ボタン・クリックします。
3. ポップアップ・メニューから「**表のインポート**」をクリックします。
4. 適切な .ddl ファイルを探して、「**オープン**」をクリックします。
- 5.

---

関連タスク:

- 52 ページの『表定義ファイルの作成およびインポート』
- 54 ページの『表定義ファイルの削除』

#### 関連例:

179 ページの『CREATE TABLE ステートメントの例』

179 ページの『データベース照会の定義』

## 表定義ファイルの削除

プロジェクトから表構造を除去するには、表定義ファイルを削除します。

---

#### 表定義ファイルを削除するには:

1. 「プロジェクト」ペインで、削除したい表を右マウス・ボタン・クリックします。
2. ポップアップ・メニューから「削除」をクリックします。

---

#### 関連タスク:

52 ページの『表定義ファイルの作成およびインポート』

53 ページの『表定義ファイルの更新』

#### 関連例:

179 ページの『CREATE TABLE ステートメントの例』

179 ページの『データベース照会の定義』

## リスト制御のデータ・ソースの定義

「データ・ソースのリスト」ウィンドウを使用して、リストに表示する表定義ファイルおよびフィールドを定義します。

リスト制御のデータ・ソースを定義するには、事前にプロジェクトに表をインポートしておかなければなりません。

「データ・ソース」および「データ・フィールド」は、フィールドに関連するデータベース表と列を識別します。フォーム上の「データ・ソース」と「データ・フィールド」の関連付けはすべて、そのフォームの SQL 照会のビルドに使用されます。アプリケーションを実行すると、これらの照会が実行され、検索されたデータがフォーム上に表示されます。データベース更新のイベント (レコードの作成、変更、削除) とフォームを関連付けた場合、イベントが起動された際に現在のフォームの内容でデータベースを更新する SQL ステートメントが実行されます。

「リンク・ソース」および「リンク・フィールド」は、2 つのデータベース表からの情報を結合する SQL 照会をビルドするために使用されるデータベース表と列を示します。

たとえば、Visiting Nurses アプリケーションの最初のフォームに、予約時刻と患者 ID のリストがあります。この情報は、「巡回予定」表と「巡回個人情報」表を結合することで取得されます。表の結合方法を定義する表列は、「巡回予定.患者ID」お

よび「巡回個人情報.ID」です。次のオプションを選択すると、「巡回予定.患者ID」と「巡回個人情報.ID」が等しい、「巡回予定」表および「巡回個人情報」表の内容を結合する照会がフォームに作成されます。

- データ・ソース: 巡回予定
- データ・フィールド: 患者ID
- リンク・ソース: 巡回個人情報
- リンク・フィールド: ID

---

### リスト制御のデータ・ソースを定義するには:

1. リスト制御を選択します。
2. 「プロパティおよびイベント」ペインの「プロパティ」タブをクリックします。
3. 「データ・ソース」の隣にあるフィールドで、 をクリックします。「データ・ソースのリスト」ウィンドウが開きます。
4. オプション: 「データ・ソース」フィールドで、プロジェクトにインポートした表の中から表定義を選択します。「データ・ソース」プルダウン・メニューには、プロジェクトにインポートした表のみが表示されます。表がない場合は、表がプロジェクト内にインポートされていることを確認してください。
5. 「データ・フィールド」フィールドで、選択したデータ・ソース表から列を選択します。このプルダウン・メニューに表示される列は、選択したデータ・ソースからのものです。何も表示されない場合は、データ・ソースが選択されていることを確認してください。
6. オプション: 「リンク・ソース」フィールドで、プロジェクトにインポートした表の中から表定義を選択します。「リンク・ソース」プルダウン・メニューには、プロジェクトにインポートした表のみが表示されます。表がない場合は、表がプロジェクトにインポートされていることを確認してください。
7. オプション: 「リンク・フィールド」フィールドで、選択したリンク・ソース表からフィールドを選択します。このプルダウン・メニューに表示されるフィールドは、選択したリンク・ソースからのものです。何も表示されない場合は、リンク・ソースが選択されていることを確認してください。
8. オプション: 「使用可能」フィールドで、データを表示するには「使用可能」、データを非表示にするには「使用不可能」を選択します。
9. オプション: 「幅」フィールドで、リスト内の列の幅を文字数で指定します。デフォルトの幅は、表に設定されている幅に等しくなります。「幅」フィールドが空白の場合、列のサイズは内容に合わせて自動的に変更されます。
10. オプション: 「区切り文字」フィールドに、各列の情報を分離する記号およびスペースを入力します。表示させる区切り文字をそのまま入力します。区切り文字は、その文字を後ろに表示したい列の行に入力します。
11. オプション: 「位置合わせ」フィールドで、その列におけるテキストの位置合わせ方法を選択します。プルダウン・リストから「左そろえ」、「右そろえ」、または「中央」を選択します。
12. オプション: 行順序を変更します。

13. 「OK」をクリックします。

---

**関連タスク:**

52 ページの『表定義ファイルの作成およびインポート』

**関連例:**

179 ページの『CREATE TABLE ステートメントの例』

179 ページの『データベース照会の定義』

**行順序の変更**

行順序を変更するには、事前にリスト制御のデータ・ソースを定義しておかなければなりません。

以下のステップは、リスト制御のデータ・ソースを定義する、より大きなタスクの一部です。行順序を変更するステップを完了したら、『リスト制御のデータ・ソースの定義』に戻ります。

---

**行順序を変更するには:**

1. 「データ・ソースのリスト」ウィンドウで、行を選択し、行番号を右マウス・ボタン・クリックしてメニューを表示します。
2. 実行するアクションを選択します。
  - 「新規行の挿入」 - 選択された行の上に空白行を挿入します。
  - 「行の削除」 - 選択された行を削除します。
  - 「行の上移動」 - 選択された行を 1 行上に移動します。
  - 「行の下移動」 - 選択された行を 1 行下に移動します。

『リスト制御のデータ・ソースの定義』に戻ります。

---

---

## フォーム

アプリケーションにフォームを追加します。

---

**関連タスク:**

60 ページの『制御』

52 ページの『表およびデータ・ソース』

## 新規フォームの作成

新規プロジェクトを作成する場合、ブランク・フォームが自動的に作成されます。追加のフォームを作成する方法がいくつかあります。

フォーム作成ウィザードでは、選択されている各データベース列のラベルとフィールドを含むフォームを作成します。また、特定のイベントが関連付けられたボタンやメニュー項目を追加することもできます。ターゲット・デバイスがサポートしていれば、アクションとコマンド・ボタン配列 (CBA) とを関連付けるフォームも作成できます。

ウィザードで作成した制御は、ウィザードを使用せずに変更できます。

---

### ウィザードを使用して新規フォームを作成するには:

1. 「プロジェクト」ペインで、新規フォームに表示させる列を含む表をクリックします。
2. 表の上で右マウス・ボタン・クリックし、「**フォーム作成ウィザードの立ち上げ**」をクリックしてウィザードを開始します。 デフォルトでは、すべての列とデータベース操作が選択されています。
3. オプション: 「データベース列」 ページで、以下のステップを実行します。
  - 「**列**」列で、フォームに表示したいフィールドにチェックマークを付けます。デフォルトでは、すべての列が選択されています。「**タイプ**」列には、そのフィールドおよびフォームで受け入れることができるデータのタイプが表示されます。
  - 「**ラベル**」列に、各フィールドのラベルとして表示したいテキストを入力します。
4. 「**次へ**」をクリックします。
5. オプション: 「データベース操作」 ページで、次のステップを実行します。
  - 「**アクション**」列で、有効なアクションのリストから、フォームに追加したいアクションを選択します。
  - 「**テキスト**」列に、ボタン・ラベル上に表示するテキストを入力します。
  - 「**コマンド・ボタン配列**」列で、デバイスのコマンド・ボタン配列と関連付けるデータベース操作を最大 4 つまで選択します。操作を CBA に関連付けない場合、操作はメニューに割り当てられます。「**コマンド・ボタン配列**」列がない場合、このオプションは現行のターゲット・デバイスでは使用できません。
6. 「**完了**」をクリックして、フォームを作成します。

### 新規フォームを手動で作成するには:

1. 「プロジェクト」ペインで、 「**フォーム**」を右マウス・ボタン・クリックします。
2.  「**新規フォームの追加**」をクリックします。「**フォーム**」ペインに、新しいフォームが表示されます。現行のターゲット・プラットフォームがコマンド・ボ

タン配列 (CBA) をサポートしている場合、この制御は新しいフォームの固定位置に追加されます。CBA は移動することも、削除することもできません。

---

#### 関連タスク:

- 52 ページの『表およびデータ・ソース』
- 60 ページの『フォームへの制御の追加』
- 『アプリケーション初期フォームの設定』
- 59 ページの『フォームのサイズ変更』
- 『フォームのズーム率の変更』
- 59 ページの『フォームの削除』

## アプリケーション初期フォームの設定

アプリケーションを立ち上げると、必ず初期フォームが表示されます。初期フォームはアプリケーションの開始点です。残りのフォームの順序は、アプリケーションで使用されるフォーム・ナビゲーションに基づいて決定されます。

---

#### アプリケーション初期フォームを設定するには:

次のいずれかを行います。

- ターゲット OS のプロパティを変更します。
    1. 「プロジェクト」ペインで <ターゲット OS> をクリックします。
    2. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「始動フォーム」フィールドからフォームを選択します。
  - または、フォームを右マウス・ボタン・クリックして、「初期フォームとして設定」をクリックします。
- 

#### 関連タスク:

- 52 ページの『表およびデータ・ソース』
- 57 ページの『新規フォームの作成』
- 59 ページの『フォームのサイズ変更』
- 『フォームのズーム率の変更』
- 59 ページの『フォームの削除』

## フォームのズーム率の変更

プロジェクトを最初にオープンしたときのデフォルト・フォーム・ズームは、ターゲットのプラットフォームによって異なります。各フォームは、個別にサイズ変更することができます。

---

フォームのズーム率を変更するには:

1. 「フォーム」ペインでフォームを選択します。
2. フォームの上部にある下矢印をクリックします。
3. リストから希望するフォーム・ズーム率を選択します。

---

関連タスク:

『フォームのサイズ変更』

## フォームのサイズ変更

プロジェクトを最初にオープンしたときのデフォルト・フォーム・サイズは、ターゲットのプラットフォームによって異なります。Mobile Application Builder は、フォームにとって最適なデフォルト・サイズを選択しますが、各フォームは個別にサイズ変更することもできます。

---

フォームをサイズ変更するには:

1. 「フォーム」ペインでフォームを選択します。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページで、フォームの「幅」プロパティと「高さ」プロパティを指定します。

---

関連タスク:

58 ページの『フォームのズーム率の変更』

## フォームの削除

プロジェクトからフォームを削除することができます。

---

フォームを削除するには:

1. 「フォーム」ペインまたは「プロジェクト」ペインでフォームを右マウス・ボタン・クリックします。
2. ポップアップ・メニューから「削除」をクリックします。確認ウィンドウが表示されます。
3. 「OK」をクリックしてウィンドウをクローズし、フォームを削除します。

---

関連タスク:

57 ページの『新規フォームの作成』

---

## 制御

パレットには、アプリケーションを組み立てるのに使用される制御が含まれます。Mobile Application Builder には、制御の標準パレットが付いています。

それぞれの制御ごとに、ご使用のアプリケーションの要求に合うようカスタマイズできる属性がたくさんあります。ほとんどの制御に、「名前」および「表題」プロパティがあります。「名前」プロパティは、システムにより生成される、制御のための固有の名前であり、編集することはできません。「名前」はユーザーには表示されません。制御のラベル用には、「表題」プロパティを使用してください。「表題」は、フォーム・ビットマップ制御の場合を除き、ユーザーに表示されます。

パレットで使用できる制御は、ターゲット・プラットフォームによって異なります。

---

### 関連タスク:

57 ページの『新規フォームの作成』

### 関連参照情報:

128 ページの『プロパティの制御』

## フォームへの制御の追加

パレットには、アプリケーションを組み立てるのに使用される制御が含まれます。Mobile Application Builder には、制御の標準パレットが付いています。

---

### フォームに制御を追加するには:

1. パレット上で制御アイコンをクリックします。これにより、制御と一緒にカーソルがロードされ、カーソルが十字線に変わります。
2. カーソルをフォーム上の任意の場所に移動します。
3. フォームをクリックして制御をドロップします。選択された制御を使用しない場合は、別の制御をクリックするか、あるいはその制御を再度クリックして選択解除してください。

---

### 関連タスク:

64 ページの『制御プロパティの構成』  
『制御』

### 関連参照情報:

128 ページの『プロパティの制御』

## フォームからの制御の削除

フォームから制御を削除することができます。

フォームから制御を削除するには:

1. 「フォーム」ペインで制御を右マウス・ボタン・クリックします。
2. 「削除」をクリックするか、またはメインメニューから「編集」->「削除」の順にクリックします。

関連タスク:

60 ページの『フォームへの制御の追加』

関連参照情報:

128 ページの『プロパティの制御』

## 制御の位置合わせ

位置合わせ機能を使用して、別の制御の位置に基づいて 1 つ以上の制御を移動します。最初に選択した制御は、アンカーとして使用され、その他の選択した制御は、そのアンカーの位置に基づいて位置合わせされます。アンカー制御のハンドルは単一色で他の選択済み制御のハンドルは白色です。それぞれの位置合わせ機能の動作に関する説明については、以下の図表を参照してください。

制御を位置合わせするには:

1. まず、アンカーとして提供する制御をクリックします。
2. Ctrl キーを押したままで別の制御をクリックして、アンカー制御に位置合わせします。複数の制御を選択するときは、Ctrl キーを押しながら、選択する制御をクリックします。
3. メインメニューから「フォーマット」->「位置合わせ」->「方向」の順にクリックし、ツールバーのアイコンを使用するか、または表示されたリストから位置合わせを行う選択項目を右マウス・ボタン・クリックして選択します。

表 12. 位置合わせオプション: 説明

左そろえ		選択された制御を、選択されたアンカー制御の左端に位置合わせします。
中央		選択された制御を、選択されたアンカー制御の垂直軸に沿って位置合わせします。
右方		選択された制御を、選択されたアンカー制御の右端に位置合わせします。
トップ		選択された制御を、選択されたアンカー制御の上端に位置合わせします。

表 12. 位置合わせオプション (続き) : 説明

中間		選択された制御を、選択されたアンカー制御の水平軸に沿って位置合わせします。
ボトム		選択された制御を、選択されたアンカー制御の下端に位置合わせします。

---

**関連タスク:**

『制御の移動』  
63 ページの『制御の再配列』

**関連参照情報:**

128 ページの『プロパティの制御』

## 制御の移動

ナッジ位置決め機能を使用して、フォーム上の制御を個別または複数移動することができます。また、「プロパティおよびイベント」ペインの「プロパティ」ページで制御のプロパティを変更して、制御を移動することもできます。

---

**ナッジ位置決め機能を使用して制御を移動するには:**

1. フォーム上で 1 つまたは複数の制御を選択します。
2. キーボードの矢印キーを使用して、選択された 1 つ以上の制御を一度に 1 ピクセル移動します。

**プロパティを変更して制御を移動するには:**

1. 制御を選択します。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「左方」プロパティを調整して、制御の左端から画面の端までの距離を指定します。  
「トップ」プロパティを調整して、制御の上端から画面の端までの距離を指定します。

---

**関連タスク:**

61 ページの『制御の位置合わせ』  
63 ページの『制御の再配列』  
63 ページの『制御のサイズ変更』

## 制御のサイズ変更

「フォーム」ペインで制御を選択すると、サイズ変更ハンドルが制御の角および辺に沿って表示されます。制御をサイズ変更するには、このサイズ変更ハンドルをドラッグします。より厳密にサイズ変更する場合は、制御の「高さ」プロパティと「幅」プロパティを指定します。

---

**サイズ変更ハンドルを使用して制御をサイズ変更するには:**

1. 「フォーム」ペインでサイズ変更したい制御をクリックします。サイズ変更ハンドルが制御の角および辺に表示されます。
2. サイズ変更ハンドルをドラッグし、制御のサイズを変更します。

**プロパティを調整して制御をサイズ変更するには:**

1. サイズ変更したい制御をクリックします。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「幅」フィールドに制御の幅を指定し、「高さ」フィールドに制御の高さを指定します。

---

**関連タスク:**

62 ページの『制御の移動』  
『制御の再配列』

**関連参照情報:**

128 ページの『プロパティの制御』

## 制御の再配列

順序機能を使用して、1 つの制御をフォームの前または後ろに移動させます。制御が階層化されている場合、順序はより明確になります。それぞれの順序機能の振る舞いに関する説明については、以下の図表を参照してください。

---

**制御を再配列するには:**

1. 移動したい制御をクリックします。この機能は制御が 1 つ選択されているときにのみ機能します。
2. メインメニューから「フォーマット」→「順序」→「方向」の順にクリックし、ツールバーのアイコンを使用するか、表示されたリストから順序づけを行う選択項目を右マウス・ボタン・クリックして選択します。

表 13. 順序オプション: 説明

前へ移動		制御を、フォームの前方に移動します。
後ろへ移動		制御を、フォームの後方に移動します。

---

**関連タスク:**

- 62 ページの『制御の移動』
- 63 ページの『制御の再配列』

**関連参照情報:**

- 128 ページの『プロパティの制御』

## 制御プロパティの構成

制御プロパティを構成するには、事前にフォームに制御を追加しておかなければなりません。

ほとんどの制御プロパティにはあらかじめデフォルト値が設定されています。特定の表の読み取り、書き込みのため、チェック・ボックス、フィールド、リスト、データ・ソース、データ・フィールド・プロパティを定義する必要があります。フォーム・ビットマップ制御の場合、ビットマップ・リソース名を定義する必要があります。

---

**制御プロパティを構成するには:**

1. 「フォーム」ペインまたは「プロジェクト」ペインで制御を選択します。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページで制御プロパティを変更します。それぞれの制御プロパティの詳しい説明については、128 ページの『プロパティの制御』を参照してください。

---

**関連タスク:**

- 82 ページの『イベント、アクション、およびターゲット』  
『デバイス制御プロパティの構成』

**関連参照情報:**

- 128 ページの『プロパティの制御』

## デバイス制御プロパティの構成

Mobile Application Builder は、多くのアプリケーション制御をサポートします。それぞれの制御には、プロパティが関連付けられています。デバイス制御構成を使用すれば、各制御のプロパティに対するデフォルト値を変更することができます。これらの変更は、Mobile Application Builder のすべての制御に影響を与えるもので、プロジェクトに限定されません。

---

デバイスの制御プロパティを構成するには:

1. 「プロジェクト」ペインの <ターゲット OS> を右クリックし、「デバイス構成のオープン」をクリックします。
2. Palettes ノードを展開して、使用可能な制御パレット・オプションを調べます。
3. 制御パレット・ノードを展開して、使用可能な制御を調べます。
4. 制御の名前をクリックし、右側パネル上のデフォルト・プロパティを変更します。
5. 「ファイル」→「クローズ」の順にクリックします。変更が適用されます。

---

関連タスク:

64 ページの『制御プロパティの構成』

関連参照情報:

128 ページの『プロパティの制御』

## 制御のフォントの変更

アプリケーションで制御の表題を表示する場合に使用するフォントを変更することができます。フォントは、アプリケーション・レベルで、また個々の制御ごとに設定できます。別のフォントを指定しない限り、デフォルトのフォントが使用されます。

フォント設定の変更は、以下の方法で有効になります。

- 制御のプロパティ設定を使用して個々の制御を変更する場合、変更はその制御にのみ適用されます。
- アプリケーション・プリファレンスでデフォルト設定を変更する場合、制御のプロパティを使用して以前に個別に変更された制御を除くすべての制御に、変更が適用されます。
- 個別に変更した制御をデフォルト設定にリセットできます。

フォントのサポートは、ターゲット・プラットフォームによって異なります。

---

アプリケーション・レベルでデフォルトのフォント設定を変更するには:

1. 「ファイル」→「プリファレンス」の順にクリックします。「プリファレンス」ウィンドウが開きます。
2. 「アプリケーション」ノードを拡張すると、アプリケーション・プリファレンスが表示されます。

3. 「フォント」ノードをクリックします。右側のペインでフォント設定を指定するか、または「**デフォルトのリストア**」をクリックしてデフォルトの Mobile Application Builder 設定に戻します。設定オプションは、ターゲット・プラットフォームによって異なります。
4. 「**OK**」をクリックします。行った変更は、制御のプロパティを使用して以前に変更した制御を除くすべての制御に適用されます。

#### 個々の制御のフォントを変更するには:

1. フォントを変更したい制御をクリックします。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページで、次のいずれかを行います。使用できるオプションは、ターゲット・プラットフォームによって異なります。
  - 「**フォント**」フィールドでオプションを選択します。このオプションを使用できるのは、ターゲット・プラットフォームが Palm OS の場合です。
  - または、
    -  をクリックして、「フォント・オプション」ウィンドウを開きます。

「フォント」フィールドに  がない場合、このオプションは現行のターゲット・プラットフォームでは使用できません。
    - オプション: 「**フォント**」フィールドでフォントを指定します。
    - オプション: 「**スタイル**」フィールドでスタイルを指定します。
    - オプション: 「**サイズ**」フィールドでサイズを指定します。
    - 「**OK**」をクリックします。「フォント・オプション」ウィンドウが閉じま  
すし、ユーザーの選択が制御に適用されます。

---

#### 関連タスク:

64 ページの『制御プロパティの構成』

## 制御の色の変更

アプリケーション内の制御を表示するために使用される色を変更することができます。色は、アプリケーション・レベルで、また個々の制御ごとに設定できます。別の色を指定しない場合は、デフォルトの色が使用されます。カラー・オプションのサポートは、ターゲット・プラットフォームによって異なります。

カラー設定の変更は、以下の方法で有効になります。

- 制御のプロパティ設定を使用して個々の制御を変更する場合、変更はその制御にのみ適用されます。
- アプリケーション・プリファレンスでデフォルト設定を変更する場合、制御のプロパティを使用して以前に個別に変更された制御を除くすべての制御に、変更が適用されます。
- 個別に変更した制御をデフォルト設定にリセットできます。

カラー・サポートは、ターゲット・プラットフォームによって異なります。

---

アプリケーション・レベルでデフォルトのカラー設定を変更するには:

1. 「ファイル」→「プリファレンス」の順にクリックします。「プリファレンス」ウィンドウが開きます。
2. 「アプリケーション」ノードを拡張すると、アプリケーション・プリファレンスが表示されます。
3. 「UI カラー」ノードを展開し、カラー・オプションを設定したい制御またはリソースをクリックします。
4. 右側のペインでカラー設定を指定するか、または「デフォルトのリストア」をクリックしてデフォルトの Mobile Application Builder 設定に戻します。設定オプションは、ターゲット・プラットフォームによって異なります。
5. 「OK」をクリックします。行った変更は、制御のプロパティを使用して以前に変更した制御を除くすべての制御に適用されます。

個々の制御の色を変更するには:

1. 色を変更したい制御をクリックします。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページで以下を行います。使用できるオプションは、ターゲット・プラットフォームによって異なります。
  - 「背景」または「前景」プロパティの横にある  をクリックして、「カラー・オプション」ウィンドウを開きます。「背景」または「前景」プロパティがない場合、現行ターゲット・プラットフォームではこのオプションは使用できません。
  - 「カラー・オプション」ウィンドウで、色を指定するか、または「デフォルトを使用 (Use Default)」をクリックしてプロジェクトのデフォルト・カラー設定に戻します。「OK」をクリックします。

---

関連タスク:

64 ページの『制御プロパティの構成』

## Palm 5 ウェイ・ナビゲーションを使用可能にする

Palm アプリケーション用に 5 ウェイ・ナビゲーションを使用可能にすることができます。これによりユーザーは、Palm デバイスで 5 ウェイ・ナビゲーション・ボタンを使用して、フォーム上の制御間を移動することが可能になります。

MAB 生成アプリケーションでは、ナビゲーションは以下のようにマップされます。

ボタン方向	アクション
右	次のフィールドに移動します。
左	直前のフィールドに移動します。
上	リストが現行フォーム上にある場合に、直前のリスト項目へ移動します。

ボタン方向	アクション
下	リストが現行フォーム上にある場合に、次のリスト項目へ移動します。
中央	リスト内の強調表示された項目を選択します。これはリストでの「タップ」イベントと同等です。

フォーム上に複数のリストがある場合、フォームのオープン時に最初のリストでナビゲーション・アクションが使用可能になります。別のリスト上でユーザーが「タップ」すると、そのリストでナビゲーション・アクションが使用可能になります。

#### 前提条件:

この機能を使用可能にするには、15 ページの『Palm C コード・ビルド環境のセットアップ』で説明されているように、Palm SG SDK をインストールする必要があります。Palm SG SDK が適切にインストールされていないと、アプリケーションのビルド時にインストール関連エラーを受け取ります。

---

#### Palm アプリケーションで 5 ウェイ・ナビゲーションを使用可能にするには:

1. Palm プロジェクトをオープンします。
2. 「ファイル」 → 「プリファレンス」の順にクリックします。「プリファレンス」ウィンドウが開きます。
3. 「ビルド」ノードをクリックすると、ビルド・プリファレンスが表示されます。
4. 「5 ウェイ・ナビゲーションを使用可能にする (Enable Five Way Navigation)」チェック・ボックスを選択します。
5. 「OK」をクリックします。

---

#### 関連タスク:

47 ページの『ビルド・プリファレンスの設定』

## ポップアップ・トリガーの使用

ポップアップ・トリガーを使用すれば、制御をタップしたときに、ユーザーはドロップダウン・リストの多肢選択から選択できるようになります。リストの内容は、ポップアップ・トリガーを持つリスト制御にリンクすることで定義されます。ポップアップ・トリガーがクリックされると、アクションの命令が開始されます。次にリストが表示され、ユーザーは選択を行います。ポップアップ・トリガーのラベルが変わり選択したものが表示される前に、制御に関連するすべてのイベントが実行されます。

この制御のサポートは、ターゲット・プラットフォームによって異なります。

---

#### ポップアップ・トリガーを使用するには:

1. ポップアップ・トリガーをオンにしたいフォームにリスト制御を追加します。
2. 次のリスト・プロパティを設定します。
  - データ・ソースまたはリスト項目を定義します。これらは、ポップアップ・トリガーをクリックしたときに使用可能になる選択項目です。
  - 「可視項目」プロパティでは、ポップアップに表示する項目の数を設定しておく必要があります。
  - 「使用可能」プロパティを「使用不可能」に設定します。このリストは、ポップアップ・トリガーによって呼び出されるまで表示されません。
3. 同じフォームにポップアップ・トリガーを追加します。通常、ポップアップ・トリガーはリストの上部に位置づけます。ポップアップ・トリガーがクリックされると、リストが表示されます。
4. 次の「ポップアップ・トリガー」プロパティを設定します。
  - ポップアップ・トリガーにデフォルトで表示する表題を指定します。
  - 上記で作成したリストの名前を「ListID」プロパティの隣りに指定します。

---

#### 関連タスク:

64 ページの『制御プロパティの構成』

#### 関連参照情報:

142 ページの『ポップアップ・トリガー・プロパティ』

## ラジオ・ボタンの作成

Java プラットフォーム用のラジオ・ボタンは、複数のチェック・ボックス制御の「グループ ID」および「チェック済み」プロパティを構成することで作成できます。

この制御のサポートは、ターゲット・プラットフォームによって異なります。

---

#### ラジオ・ボタンを作成するには:

1. 複数のチェック・ボックス制御を、ラジオ・ボタンを作成するフォームに追加します。
2. すべてのチェック・ボックス制御の「グループ ID」プロパティを、以下の制約事項に従って同じ値に設定します。
  - ブランク以外の値を設定する必要があります。
  - グループ ID は、スペースなしの Java グループ名でなければなりません。
  - 先頭文字は英数字でなければなりません。

3. デフォルトで選択されている、表示する制御の「**チェック済み**」プロパティを「真」に設定して、その他のすべての制御の「**チェック済み**」プロパティを「偽」に設定します。アプリケーションをビルドおよびテストすると、これらの制御は、指定したデフォルト制御が選択された状態でラジオ・ボタンとしてレンダリングされます。

---

#### 関連タスク:

64 ページの『制御プロパティの構成』

#### 関連参照情報:

133 ページの『チェック・ボックス・プロパティ』

## リスト内でのフィールドの表示順序の定義

「順序づけ」ウィンドウを使用して、リスト内でのフィールドの表示順序を定義します。

リスト内でのフィールドの表示順序を定義するには、事前にリスト制御のデータ・ソースを定義しておかなければなりません。

---

#### リスト内でのフィールドの表示順序を定義するには:

1. リスト制御を選択します。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「**順序づけ**」フィールドの隣にある  をクリックし、「順序づけ」ウィンドウを開きます。「順序づけ」ウィンドウをオープンするには、事前にリスト制御のデータ・ソースを定義しておかなければなりません。
3. 「**データ・ソース**」列で、リスト制御のデータ・ソースとして定義した表の中から表定義を選択します。表がない場合は、リストのデータ・ソースが定義されていることを確認してください。
4. 「**データ・フィールド**」列で、選択したデータ・ソース表から列を選択します。表示される列は、前述のステップで選択したデータ・ソースからのものです。何も表示されない場合は、データ・ソースが選択されていることを確認してください。
5. 「**順序**」フィールドで、データの表示順序を選択します。「昇順」または「降順」の順序を選択できます。
6. 「**OK**」をクリックします。「順序づけ」ウィンドウが閉じます。

---

#### 関連タスク:

60 ページの『フォームへの制御の追加』

## 関連参照情報:

140 ページの『リスト・プロパティー』

## 固有値とフィールド制御との関連付け

Mobile Application Builder は、必要に応じて、CustomerID や MedicalRecordID などの ID フィールドと関連付けることができる固有 ID を生成することができます。この ID は、データベースがレコード追跡の目的にのみ使用する固有の数値として生成されるもので、必ずしもエンド・ユーザーが使用する必要はありません。

固有 ID には少なくともタイム・スタンプが入り、26 文字を必要とします。ニーズに応じて、およびフィールドに有効な文字数に応じて、接頭部ストリング、現行ユーザー ID の一部、または接尾部ストリングを含めるように固有 ID をカスタマイズすることもできます。

固有値にユーザー ID を追加する場合は、固有 ID を入れるフィールド制御があるフォームをオープンする前に、まずユーザーが情報を入力することを認めることにより、ユーザー ID 情報を使用可能にしておかなければなりません。次に、Mobile Application Builder がこのために自動的に生成するグローバル変数とユーザー ID を関連付けなければなりません。

Palm C コード・アプリケーションでは、Mobile Application Builder は MAB\_USERID というグローバル変数を生成します。Java コード・プラットフォームではすべて、このグローバル変数の名前は MABGlobalScripts.MAB\_USERID です。

固有 ID には、次の要件を満たすフィールド制御を関連付けることができます。

- 制御のデータ・フィールドとして指定されている列のデータ・タイプが VARCHAR または CHAR である。
- フィールドに有効な最大文字数が 26 文字以上である。

---

### 固有値とフィールド制御との関連付け:

1. フィールド制御を選択します。
2. 「プロパティーおよびイベント」ペインの「プロパティー」ページで、「データ・ソース」リストから表を選択します。表がない場合は、表定義が作成されてプロジェクトにインポートされていることを確認してください。
3. 「データ・フィールド」リストから列を選択します。選択した列のデータ・タイプが「データ・フィールド・タイプ」フィールドに表示されます。列のデータ・タイプは VARCHAR または CHAR でなければなりません。有効な最大文字数が「最大文字数」フィールドに表示されます。この値は 26 以上でなければなりません。
4. 「固有値の使用」フィールドの隣にある  をクリックし、固有値の特性を指定できるウィンドウを開きます。
5. 「このフィールドに固有値を設定します」オプションを選択します。
6. オプション: 「接頭部ストリングを入力してください」フィールドに、固有値の先頭に追加するストリングを入力します。

7. オプション: 隣のフィールドに数値を指定し、現行のユーザー ID の一部またはすべてを固有 ID に追加します。ここでは、固有値に追加する現行のユーザー ID の文字数を指定します。
8. オプション: 「接尾部ストリングを入力してください」フィールドに、固有値の最後に追加するストリングを入力します。
9. 「OK」をクリックします。固有値のパラメーターが有効であると、ウィンドウがクローズして、変更が適用されます。パラメーターが無効であると、メッセージ・ウィンドウが表示されて、設定を調整する必要があることが通知されます。

---

#### 関連タスク:

- 52 ページの『表定義ファイルの作成およびインポート』
- 90 ページの『グローバル変数の定義』

#### 関連参照情報:

- 136 ページの『フィールド・プロパティ』

---

## プロジェクト・リソース

Mobile Application Builder は、アプリケーションで使用する多くのリソースを提供します。リソースのプロパティおよびイベントを設定し、それらをスクリプトで使用します。

アプリケーションに使用できるリソースは、ターゲット・プラットフォームによって異なります。

---

#### 関連タスク:

- 38 ページの『MAB プロジェクト』

#### 関連参照情報:

- 148 ページの『リソース・プロパティ』

## 新規アラートの作成

アラートは、ユーザーの応答を要求するプロンプトを出すのに使用されます。ユーザーは処理を行う前に、アラートに応答する必要があります。スクリプトを使用してアラートを参照します。

アラートおよびアラート・プロパティのサポートは、ターゲット・プラットフォームによって異なります。

---

新規 Palm アラートを作成するには:

1. 「プロジェクト」ペインで  「アラート」を右マウス・ボタン・クリックします。「プロジェクト」ペインに「アラート」ノードがない場合、現行のターゲット・プラットフォームでアラートはサポートされていません。
2. 「新規アラートの追加」をクリックします。「プロジェクト」ペインで、「アラート」の下に新しいアラートが表示されます。
3. 「プロジェクト」ペインで新しいアラートをクリックします。
4. 「プロパティおよびイベント」ペインの「プロパティ」ページで、表題、アラート・タイプ (Palm アプリケーションのみ)、メッセージ、ボタン・テキスト、およびデフォルト・ボタン ID (Palm アプリケーションのみ) を入力します。
5. アラートを関連付けるフォームまたは制御をクリックし、「選択」->「新規スクリプトの作成」の順にクリックします。スクリプト・ウィンドウが開きます。
6. スクリプト・ウィンドウに次のスクリプトを入力します。 `alertName` には、システム生成の実際のアラート名を指定します。

```
FrmAlert(alertName);
```

7. 「ファイル」->  「スクリプトの保管」の順にクリックします。
8. 「ファイル」->「クローズ」の順にクリックします。スクリプト・ウィンドウがクローズし、スクリプトを作成したフォームまたは制御に、スクリプトがイベントとして追加されます。

#### Java プラットフォームで新規アラートを作成するには:

カスタム・アラートを作成したり、標準アラートを使用したりすることができます。

- カスタム・アラートを作成するには:

1. 「プロジェクト」ペインで  「アラート」を右マウス・ボタン・クリックします。「プロジェクト」ペインに「アラート」ノードがない場合、現行のターゲット・プラットフォームでアラートはサポートされていません。
2. 「新規アラートの追加」をクリックします。「プロジェクト」ペインで、「アラート」の下に新しいアラートが表示されます。
3. 「プロジェクト」ペインで新しいアラートをクリックします。
4. 「プロパティおよびイベント」ペインの「プロパティ」ページで、表題、メッセージ、およびボタン・テキストを入力します。
5. アラートを関連付けるフォームまたは制御をクリックし、「選択」->「新規スクリプトの作成」の順にクリックします。スクリプト・ウィンドウが開きます。
6. スクリプト・ウィンドウに次のスクリプトを入力します。 `alertName` には、システム生成の実際のアラート名を指定します。 `showAlert` メソッドからの戻り値は、クリックされたアラート上のボタンの索引です (1 つ目のボタンの索引は 0 になります)。

```
Alerts.showAlert("alertName");
```

7. 「ファイル」->  「スクリプトの保管」の順にクリックします。
  8. 「ファイル」->「クローズ」の順にクリックします。スクリプト・ウィンドウがクローズし、スクリプトを作成したフォームまたは制御に、スクリプトがイベントとして追加されます。
- 標準アラートを使用するには:
    1. アラートを関連付けるフォームまたは制御をクリックし、「選択」->「新規スクリプトの作成」の順にクリックします。スクリプト・ウィンドウが開きます。
    2. スクリプト・ウィンドウに次のスクリプトを入力し、太字イタリックのテキストを次のようにカスタマイズします。

```
int showAlert(alertType,buttonOption,title,message)
```

ここで、

- *alertType* と *buttonOption* は整数です。
- *title* と *message* はストリングです。

alertType	次のアラート・タイプのいずれかを指定します。これを使用して、アラートに表示するイメージが決定されます。 <ul style="list-style-type: none"> <li>• CONFIRMATION_ALERT</li> <li>• INFORMATION_ALERT</li> <li>• ERROR_ALERT</li> <li>• WARNING_ALERT</li> </ul>
buttonOption	アラートに表示するボタンおよびテキストについて次のオプションのいずれかを指定します。 <ul style="list-style-type: none"> <li>• DEFAULT_OPTION</li> <li>• OK_CANCEL_OPTION</li> <li>• YES_NO_CANCEL_OPTION</li> <li>• YES_NO_OPTION</li> </ul>
String title	アラートの表題を指定します。
String message	アラートに表示するメッセージ・テキストを指定します。

#### 関連概念:

157 ページの『スクリプト記述サポート』

#### 関連タスク:

87 ページの『Mobile Application Builder 中のスクリプト』

#### 関連参照情報:

149 ページの『アラート・プロパティ』

## ビットマップの作成および表示

アプリケーションでは、フォームまたはボタン制御の上にビットマップを表示することができます。ビットマップ・イメージは、白黒または直接色形式 (Palm OS 4.0 またはそれ以上でサポートされます) でビットマップ・イメージを保管できる、任意の外部ドロー・プログラムを使用して作成することができます。ビットマップは、アプリケーションのプロジェクト・ディレクトリーに保管します。フォームにビットマップを表示するには、フォームにフォーム・ビットマップ制御を追加します。また、ボタン、プッシュボタン、繰り返しボタン上にビットマップを表示するには、制御のプロパティを構成します。

ビットマップの表示特性は、ターゲット・デバイスの画面解像度によって異なります。たとえば、16 ビットのビットマップを提供し、ターゲット・デバイスで最大 4 ビットのビットマップまでしかサポートされない場合、イメージは 4 ビットのビットマップとして表示されます。これは、ビットマップの外観に好ましくない影響を及ぼします。常に、エミュレーターまたはモバイル・デバイスでビットマップをテストしてください。

ビットマップのサポートは、ターゲット・プラットフォームによって異なります。

---

#### Mobile Application Builder ビットマップ・タスク:

『新規ビットマップの作成』

76 ページの『フォーム上でビットマップの表示』

77 ページの『制御上でのビットマップの表示』

---

#### 関連参照情報:

153 ページの『ビットマップ・プロパティ』

129 ページの『フォーム・ビットマップ・プロパティ』

### 新規ビットマップの作成

フォームまたは制御の上にビットマップを表示するには、事前にビットマップを作成し、プロジェクト・ディレクトリーに保管してから、ビットマップ・リソースをプロジェクトに追加しておかなければなりません。

ビットマップのサポートは、ターゲット・プラットフォームによって異なります。

---

新規ビットマップを作成するには:

1. 「プロジェクト」 ペインで「ビットマップ」を右マウス・ボタン・クリックします。「プロジェクト」ペインに「ビットマップ」ノードがない場合、現行のターゲット・プラットフォームでビットマップはサポートされていません。
2. 「新規ビットマップの追加」をクリックします。「ビットマップ」の下に新しいビットマップが表示されます。
3. 「プロジェクト」ペインで新しいビットマップを選択します。
4. 「プロパティおよびイベント」ペインの「プロパティ」タブで、「ビットマップ・ファイル名」フィールドにビットマップ・ファイルの名前を入力します。ビットマップ・ファイルは、現行のプロジェクト・ディレクトリーに入っていない必要があります。

---

#### 関連タスク:

『フォーム上でビットマップの表示』  
77 ページの『制御上でのビットマップの表示』

#### 関連参照情報:

153 ページの『ビットマップ・プロパティ』  
129 ページの『フォーム・ビットマップ・プロパティ』

### フォーム上でビットマップの表示

フォームの上にビットマップを表示するには、事前にビットマップを作成し、プロジェクトにリソースとして追加しておかなければなりません。

ビットマップのサポートは、ターゲット・プラットフォームによって異なります。

---

#### フォーム上でビットマップを表示するには:

1. フォームを選択します。
2. 制御パレットからフォーム・ビットマップ制御をフォームに追加します。
3. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「ビットマップ・リソース名」リストからビットマップを選択します。「ビットマップ・リソース名」リストに使用可能なビットマップがない場合は、ビットマップが作成されていること、およびプロジェクトにリソースとしてビットマップが追加されていることを確認してください。

---

#### 関連タスク:

75 ページの『新規ビットマップの作成』  
77 ページの『制御上でのビットマップの表示』

#### 関連参照情報:

153 ページの『ビットマップ・プロパティ』

129 ページの『フォーム・ビットマップ・プロパティ』

## 制御上でのビットマップの表示

ボタン、プッシュボタン、または繰り返しボタンの上にビットマップを表示することができます。制御の上にビットマップを表示するには、事前にビットマップを作成し、プロジェクト・ディレクトリーに保管しておかなければなりません。制御の選択されている状態および選択されていない状態の両方に対してビットマップを指定しなければなりません。そうしないと、正しく表示されません。これらのビットマップは、同じでも、異なっても構いません。

ビットマップのサポートは、ターゲット・プラットフォームによって異なります。

---

### 制御上でビットマップを表示するには:

1. ボタン、プッシュボタン、または繰り返しボタンの制御を選択します。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「**ビットマップ**」リストからビットマップを選択します。「**ビットマップ**」リストに使用可能なビットマップがない場合は、ビットマップが作成されていること、およびプロジェクト・ディレクトリーにビットマップが保管されていることを確認してください。このプロパティは、制御が選択されていない状態の場合にその制御にどのビットマップを表示するかを指定します。
3. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「**選択されたビットマップ**」リストからビットマップを選択します。「**ビットマップ**」リストに使用可能なビットマップがない場合は、ビットマップが作成されていること、およびプロジェクト・ディレクトリーにビットマップが保管されていることを確認してください。このプロパティは、制御が選択されている状態の場合にその制御にどのビットマップを表示するかを指定します。「**ビットマップ**」プロパティに指定したビットマップと同じビットマップを指定することもできます。

---

### 関連タスク:

76 ページの『フォーム上でビットマップの表示』

75 ページの『新規ビットマップの作成』

### 関連参照情報:

153 ページの『ビットマップ・プロパティ』

129 ページの『フォーム・ビットマップ・プロパティ』

## 新規アイコンの作成

各 Palm アプリケーションには、「Palm アプリケーション」ペインに表示されるアイコンが含まれます。

Mobile Application Builder はデフォルトのアイコンを提供しますが、任意の外部アイコン・エディターを使用して独自に作成することもできます。ターゲット・プラットフォームの中には、アイコンとマスクが必要なものもあります。

アイコンのサポートは、ターゲット・プラットフォームによって異なります。

アイコンの表示特性は、ターゲット・デバイスの画面解像度によって異なります。常に、エミュレーターまたはモバイル・デバイスでアイコンをテストしてください。

アイコンを作成する場合は、次の指針に従ってください。

**Palm アプリケーションの場合 (アイコンのみ):**

- このアイコン・サイズは、.bmp 形式で、32x32、32x22、または 22x22 ピクセルでなければなりません。それ以外のサイズでは正しく表示されません。
- アイコンは、カラー形式か、白黒の形式にできます。

**SymbianOS/Crystal アプリケーションの場合 (アイコンおよびマスク):**

- このアイコン・サイズは、.bmp 形式で、64x50 または 25x20 ピクセルのいずれかでなければなりません。カラーの濃さは 16 ビットです。
- アイコンは、カラー形式か、白黒の形式にできます。
- マスクは、アイコンの背景が透けて見えなければならない透過域では白、その他の領域では黒でなければなりません。

**SymbianOS/Quartz アプリケーションの場合 (アイコンおよびマスク):**

- このアイコン・サイズは、.bmp 形式で、32x32 または 20x16 ピクセルのいずれかでなければなりません。カラーの濃さは 16 ビットです。
- アイコンは、カラー形式か、白黒の形式にできます。
- マスクは、アイコンの背景が透けて見えなければならない透過域では白、その他の領域では黒でなければなりません。

---

#### アイコンを作成するには:

1. 「プロジェクト」ペインで、「アイコン」をクリックします。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページ上の「アイコン・ファイル名」に、アイコンのパスおよびファイル名を入力します。Symbian プラットフォームの場合は、アイコンとマスクに対する複数のファイル名をスペースで区切って入力します。

---

#### 関連概念:

157 ページの『スクリプト記述サポート』

#### 関連タスク:

87 ページの『Mobile Application Builder 中のスクリプト』

## 関連参照情報:

152 ページの『アイコン・プロパティ』

## 新規メニュー項目の作成

メニューまたはメニュー項目は、メニュー・バーからアクセス可能な個々の機能で、画面上の他の場所にはありません。

メニューは、メニュー・バー・トピックのドロップダウン・リストに表示されません。メニュー項目は、メニュー・バーと関連付けられている場合のみ使用可能です。メニューを作成したら、メニュー・バーを作成してメニューを関連付けます。

---

### 新規メニュー項目を作成するには:

1. 「プロジェクト」ペインで、「メニュー」を右マウス・ボタン・クリックします。
2. 「新規メニューの追加」をクリックします。「メニュー」の下に新しいメニューが表示されます。
3. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「メニュー項目テキスト」フィールドと「アクセラレーター文字」フィールドに、それぞれメニュー項目のテキストとショートカット文字を入力します。
4. メニュー項目がアプリケーションで選択されたときに実行するイベントとアクションを指定します。
  - a. 「プロパティおよびイベント」ペインで「イベント」タブをクリックします。
  - b. 左方でイベント・タイプを選択します。詳細については、『イベントの説明』を参照してください。
  - c. イベントの説明の右にある  をクリックします。「イベント・アクションおよびターゲットの選択」ウィンドウが開きます。
  - d. 「イベント・アクションおよびターゲットの選択」ウィンドウで、「アクション」リストから必要なアクションを選択します。詳細については、『アクションの説明』を参照してください。アクションは、制御のイベントが実行されたときに実行されます。「アクション」リストには、選択されたイベントと関連付けられている有効なアクションが表示されます。
  - e. 「ターゲット」リストから、アクションの対応するターゲットを選択します。ターゲットは、アクションを実行したい表、フォーム、または制御です。「ターゲット」リストには、選択されたアクションに基づいて、選択できる有効なターゲットが表示されます。
5. メニュー項目を表示するメニュー・バーを作成します。

---

### 関連タスク:

80 ページの『新規メニュー・バーの作成』

## 関連参照情報:

151 ページの『メニュー・プロパティ』

151 ページの『メニュー・バー・プロパティ』

## 新規メニュー・バーの作成

メニュー・バーは、画面上の他の場所で使用できない追加機能 (メニュー項目) へのアクセスを提供します。メニュー・バーは、Palm シルク・スクリーン・メニュー・ボタンがタップされると使用可能になります。アプリケーション内のフォームごとに複数のメニュー・バーを持つことができます。

メニュー・バーには、リストをドロップダウンし、メニュー項目を表示するトピックが含まれます。まずメニュー項目を作成し、メニュー・バーと関連付ける必要があります。メニュー・バーを作成したら、特定のフォームと関連付けて、アプリケーションで使用できるようにする必要があります。

---

### 新規メニュー・バーを作成するには:

1. 「プロジェクト」ペインで、「メニュー・バー」を右マウス・ボタン・クリックします。
  2. 「新規メニュー・バーの追加」をクリックします。「メニュー・バー」の下に新しいメニュー・バーが表示されます。
  3. 新しいメニュー・バーのプロパティに移動し、「プルダウン」フィールドにメニュー・バー・ドロップダウンのテキストを入力します。
  4. 「メニュー項目 ID」プロパティの隣りの「メニュー・バー」ドロップダウンの下に表示させる「メニュー項目」の名前を入力します。
  5. メニュー項目 ID を追加または削除するには、メニュー項目 ID を右マウス・ボタン・クリックして、「メニュー項目の追加」をクリックするか、ポップアップ・リストから「削除」をクリックします。
  6. ドロップダウン・メニューを追加するには:
    - a. 「プルダウン」を右マウス・ボタン・クリックし、「プルダウンの追加」をクリックします。追加のプルダウン項目が表示されます。
    - b. 新規ドロップダウン・メニューのテキストを入力します。
    - c. このドロップダウン・メニューにメニュー項目を追加するには、既存の最後のプルダウン項目を右マウス・ボタン・クリックし、「メニュー項目の追加」をクリックします。
    - d. 「メニュー項目 ID」プロパティの隣りの新しいメニュー・バー・ドロップダウンの下に表示させるメニュー項目の名前を入力します。
  7. メニュー・バーをフォームに関連付けて、アクティブにします。
    - フォームを選択します。
    - 「プロパティおよびイベント」ペインの「プロパティ」ページで、「メニュー ID」プロパティのリストからメニュー・バーを選択します。
-

## 関連タスク:

79 ページの『新規メニュー項目の作成』

## 関連参照情報:

151 ページの『メニュー・プロパティ』

151 ページの『メニュー・バー・プロパティ』

## 新規ストリングの作成

文字のストリングを表示するには、ストリングを使用します。ストリングを使用して、バージョン番号などのアプリケーションに関するメッセージを表示します。ストリングの参照には、スクリプトを使用します。

ストリングのサポートは、ターゲット・プラットフォームによって異なります。

### 新規ストリングを作成するには:

1. 「プロジェクト」ペインで  「ストリング」を右マウス・ボタン・クリックします。「プロジェクト」ペインに「ストリング」ノードがない場合、現行のターゲット・プラットフォームでストリングはサポートされていません。
2. 「新規ストリングの追加」をクリックします。「プロジェクト」ペインで、「ストリング」の下に新しいストリングが表示されます。
3. 「プロジェクト」ペインで新しいストリングをクリックします。
4. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「**ストリング**」フィールドにストリング・テキストを入力するか、テキストが入っているファイルの名前を入力します。
5. フォームに、ストリングを表示するフィールドを追加します。
6. フィールドを選択し、「**選択**」->「**新規スクリプトの作成**」の順にクリックします。スクリプト・ウィンドウが開きます。
7. 以下のサンプル・スクリプトを入力します。アプリケーションに合わせて、太字イタリックのテキストをカスタマイズします。フィールドに表示するテキストを設定するには、「**部品の貼り付け**」 が必要です。以下のコードのコメントを参照してください。

```
CharPtr startString;
VoidHand rsch;
/* ストリング・リソースの検索 */
rsch = DmGetResource (strRsc, stringName);
startString = MemHandleLock (rsch);
/* 部品の貼り付けの使用 */
/* 表示フィールド・ストリングの選択およびテキストの設定 */
/* CharPtr を上記で定義した値と同じ値に変更 */
PAB_setFieldDrawPtr(fieldName, startString);
/* ストリング・リソースの解放 */
MemHandleUnlock(rsch);
DmReleaseResource(rsch);
```

8. 「**ファイル**」->  「**スクリプトの保管**」の順にクリックします。

9. 「ファイル」->「クローズ」の順にクリックします。スクリプト・ウィンドウがクローズし、スクリプトを作成したフォームまたは制御に、スクリプトがタップ・イベントとして追加されます。

---

**関連概念:**

157 ページの『スクリプト記述サポート』

**関連タスク:**

87 ページの『Mobile Application Builder 中のスクリプト』

**関連参照情報:**

150 ページの『ストリング・プロパティ』

---

## イベント、アクション、およびターゲット

イベントとは、ご使用のアプリケーションでのペン入力や、キーを押すことによって引き起こされるアクションを指します。フォーム、制御、およびリソースには、一緒に使用できるイベントがたくさんあります。たとえば、ボタン制御が押されたときにスクリプトを実行し、別のフォームに移動する場合は、イベントは 2 つです。イベントは、リストされた順に発生します。

アクションは、制御のイベントが実行されたときに実行されます。「イベント・アクションおよびターゲットの選択」ウィンドウの「アクション」フィールドには、選択されたイベントに関連付けることができるアクションのリストが表示されます。リストから、イベントが実行されたときに実行するアクションを選択します。たとえば、ボタン制御の「タップ」イベントには、別のフォームをオープンする関連アクション (表示) を割り当てることができます。

ターゲットは、アクションを実行したい表、フォーム、ダイアログ、または制御です。「ターゲット」フィールドには、選択されたアクションに基づいて、選択できる有効なターゲットのリストが表示されます。たとえば、前述の「表示」アクションには、それに関連付けられたターゲット・フォームを割り当てる必要があります。

---

**関連タスク:**

60 ページの『制御』

**関連参照情報:**

153 ページの『イベントの説明』

155 ページの『アクションの説明』

## 制御イベントの構成

制御イベントを構成するには、事前にフォームに制御を追加しておかなければなりません。

---

制御イベントを構成するには:

1. 「フォーム」ペインまたは「プロジェクト」ペインで制御を選択します。
2. 「プロパティおよびイベント」ペインで「イベント」タブをクリックします。
3. 左方でイベント・タイプを選択します。詳細については、『イベントの説明』を参照してください。
4. イベントの説明の右にある  をクリックします。「イベント・アクションおよびターゲットの選択」ウィンドウが開きます。
5. 「アクション」リストから、必要なアクションを選択します。詳細については、『アクションの説明』を参照してください。
6. オプション: 「オプション」をクリックし、特定のアクションに対するオプションを指定します。「オプション」ボタンが使用できない場合は、選択したアクションに対する有効なオプションはありません。
7. 「ターゲット」リストから、アクションの対応するターゲットを選択します。「ターゲット」リストには、選択されたアクションに基づいて、選択できる有効なターゲットが表示されます。
8. オプション: 「レコードの作成」、「レコードの更新」、または「レコードの削除」アクションでは、イベントに確認ダイアログを追加します。
9. 「OK」をクリックします。指定したアクションおよびターゲットがイベントに追加されます。
10. オプション: 「プロパティおよびイベント」ペインの「イベント」ページを右マウス・ボタン・クリックし、ポップアップ・メニューから「新規イベント接続の追加」をクリックして、新規イベントを追加します。

---

関連タスク:

60 ページの『フォームへの制御の追加』

関連参照情報:

128 ページの『プロパティの制御』

153 ページの『イベントの説明』

155 ページの『アクションの説明』

### データベース・アクションへの確認ダイアログの追加

「レコードの作成」、「レコードの更新」、または「レコードの削除」アクションの確認ダイアログを追加するには、事前に制御イベントを構成しておかなければなりません。

デフォルトのメッセージを選択するか、ユーザー独自のメッセージをカスタマイズしてください。イベント・アクションにダイアログを追加すると、「プリファレンス」で設定した確認ダイアログの設定内容がオーバーライドされます。

以下のステップは、制御イベントを構成する、より大きなタスクの一部です。「レコードの作成」、「レコードの更新」、または「レコードの削除」アクションの確認ダイアログを追加するステップを完了したら、『制御イベントの構成』に戻ります。

---

**「レコードの作成」、「レコードの更新」、または「レコードの削除」アクションの確認ダイアログを追加するには:**

1. 「イベント・アクションおよびターゲットの選択」ウィンドウから、「レコードの作成」、「レコードの更新」、または「レコードの削除」アクションを選択します。
2. 「イベント・アクションおよびターゲットの選択」ウィンドウの下部の「メッセージの設定...」をクリックして、「確認および応答」ウィンドウを開きます。
3. 「確認および応答」ウィンドウで、選択したアクションについての確認ダイアログの特性を定義します。
  - 「メッセージ」を選択し、選択したアクションの前または後にダイアログをアクティブにします。
  - 「メッセージ」チェック・ボックスの隣のフィールドをクリックして、メッセージをリストから選択するか、カスタマイズしたメッセージ・テキストを入力します。
4. 「OK」をクリックします。

『制御イベントの構成』に戻ります。

---

## 制御イベントの再配列

制御のイベントが実行される順序を変更できます。

---

**制御イベントを再配列するには:**

1. 「フォーム」ペインまたは「プロジェクト」ペインで制御を選択します。
  2. 「プロパティおよびイベント」ペインで「イベント」タブをクリックします。
  3. イベントを右マウス・ボタン・クリックし、「イベントの再配列」をクリックします。
  4. オープンしたウィンドウで、イベントをクリックし、上または下矢印をクリックしてリスト内のイベントを上または下に移動します。
  5. 「OK」をクリックします。
-

#### 関連タスク:

60 ページの『フォームへの制御の追加』

#### 関連参照情報:

128 ページの『プロパティの制御』

153 ページの『イベントの説明』

155 ページの『アクションの説明』

## 制御へのデータベース・プリファレンス・ダイアログを表示するイベントの追加

Mobile Application Builder は、コンパクト・フラッシュ、IBM マイクロドライブ、Palm 拡張カード・スロットなどの 2 次ストレージ・デバイス上にある DB2 Everyplace データベースをサポートします。暗号化 DB2 Everyplace データベースもサポートします。Mobile Application Builder は、実行中のアプリケーションからユーザーがデータベースの位置を選択したり、暗号化されたデータにアクセスするためにユーザー ID とパスワードを入力したりできるダイアログを表示するために必要なコードを生成します。イベント・アクションを関連付けることができるアプリケーション・メニュー、ボタン、または任意の制御からこのダイアログへのアクセスを手動で選択することもできます。このダイアログは、データベースの位置だけを入れるように構成することも、データベースの位置と、ユーザー ID フィールド、パスワード・フィールドの両方を入れるように構成することもできます。2 次ストレージ・デバイス上のデータにアクセスするには、データベース位置フィールドのみのダイアログを使用します。暗号化されたデータにアクセスするには、すべてのフィールドを含むダイアログを使用します。

Palm アプリケーションの場合、ロケーション・ダイアログには、ユーザーがデバイス RAM (1 次ストレージ) または 2 次ストレージ・カードのいずれかを選択できるドロップダウン・リストがあります。ユーザーが 2 次ストレージ・カードを選択した場合、ユーザーは、どのディレクトリーにデータベース表が含まれているのかを示すディレクトリー・パスも追加フィールドに入力する必要があります。このフィールドに、ユーザーはスラッシュを使用して (先頭にはスラッシュは付けない) ディレクトリー・パスを入力します。末尾にスラッシュを使用することはできませんが、必須ではありません。以下に例を示します。

- 正しいロケーション・エントリー: db2e/database/tables
- 正しいロケーション・エントリー: db2e/database/tables/
- 誤ったロケーション・エントリー: /db2e/database/tables

このロケーション・ダイアログは Mobile Application Builder によって自動的に生成される場合もあります。たとえば、アプリケーションが暗号化されたデータにアクセスしたときに、有効なユーザー ID 情報とパスワード情報がまだ保管されていない場合は、ダイアログが自動的に立ち上がり、ユーザーは適切な情報を入力することができます。

ユーザー ID 情報とパスワード情報は、一度入力すると、アプリケーションがクローズされるまでそのアプリケーション内に保管されます。また、ユーザーは、ダイアログ上のチェック・ボックスを選択することで、パスワード情報の永久保管を指

定することもできます。このオプションが選択されていない場合、パスワード情報はアプリケーションがクローズされると破棄され、ユーザー ID はデフォルトのユーザー ID として保管されます。

ユーザーがデータベースの位置を現行の接続位置とは異なる位置に変更すると、新しい接続を確立できるように、アプリケーションの再始動を求めるプロンプトが表示されます。これにより、アプリケーションですでに取得されているデータの整合性が確保されます。

---

**データベース・プリファレンス・ダイアログを表示するイベントを制御に追加するには:**

1. ダイアログの追加先の制御を選択します。
2. 「プロパティおよびイベント」ペインで「イベント」タブをクリックします。
3. この制御に関連付けられた既存のアクションがない場合は、右にある  をクリックして、「イベント・アクションおよびターゲットの選択」ウィンドウを開きます。または、表示されている最後のイベントを右マウス・ボタン・クリックし、「新規イベント接続の追加」をクリックしてから、 をクリックします。
4. 「アクション」リストで「表示」をクリックします。
5. 「ターゲット」リストで、次のオプションのいずれかを選択します。
  - データベース位置フィールド、ユーザー ID フィールド、およびパスワード・フィールドを含むダイアログを生成するには、「データベース認証ダイアログ」をクリックします。
  - データベース位置フィールドのみを含むダイアログを生成するには、「データベース位置ダイアログ」をクリックします。

---

**関連タスク:**

60 ページの『フォームへの制御の追加』

**関連参照情報:**

128 ページの『プロパティの制御』

153 ページの『イベントの説明』

155 ページの『アクションの説明』

## 「レコードの削除」アクションへの物理削除オプションの追加

レコードの DELETE アクションに、クライアント (モバイル) データベース上でのみレコードを削除するオプションを追加することができます。このオプションは、モバイル・データベースからレコードを削除したいが、同期化する際にサーバー (エンタープライズ) データベースに削除アクションを反映したくない場合に適用できます。

---

「レコードの削除」アクションに物理削除オプションを追加するには:

1. 削除アクションを追加したい制御を選択します。
2. 「プロパティおよびイベント」ペインで「イベント」タブをクリックします。
3. この制御に関連付けられた既存のアクションがない場合は、右にある  をクリックして、「イベント・アクションおよびターゲットの選択」ウィンドウを開きます。または、表示されている最後のイベントを右マウス・ボタン・クリックし、「新規イベント接続の追加」をクリックしてから、 をクリックします。
4. 「アクション」リストで「レコードの削除」をクリックします。
5. 「オプション」をクリックします。「オプション」ウィンドウが開きます。
6. 「デバイスからのみレコードを削除します (物理削除)」チェック・ボックスを選択します。
7. 「OK」をクリックします。「オプション」ウィンドウが閉じます。

---

**関連参照情報:**

- 128 ページの『プロパティの制御』
- 153 ページの『イベントの説明』
- 155 ページの『アクションの説明』

---

## Mobile Application Builder 中のスクリプト

Mobile Application Builder は、リソースおよび制御の一部に関連した多くのコード関数を生成します。さらに、組み込みスクリプト関数を使用して、独自の関数を追加することができます。スクリプト関数は、モバイル・デバイスの API 呼び出しを使用して、ユーザー独自の C コードを書き込むことができるようにします。

---

**関連概念:**

- 157 ページの『スクリプト記述サポート』

**関連タスク:**

- 48 ページの『スクリプト・プリファレンスの設定』

## Java スクリプトを使用して MAB 生成コードにアクセスする

このトピックでは、生成された Java アプリケーションのデフォルト動作をスクリプトを使用して拡張する方法について説明します。次の情報を使用して作成されたスクリプトのロジックに加えて、構文や意味エラーも開発者の責任となります。

## 生成された MAB オブジェクト:

次の表には、MAB オブジェクトの生成方法、およびスクリプトでのアクセス方法が示されています。

表 14. 生成された MAB オブジェクト

MAB オブジェクト	生成されるオブジェクト	タイプ	スクリプトでのアクセス
ホスト変数	クラス <i>MABGlobalScripts</i> の静的メンバー	<i>java.lang.String</i>	<i>MABGlobalScripts.hv</i> としてアクセス可能です ( <i>hv</i> はホスト変数)。
グローバル・スクリプト	クラス <i>MABGlobalScripts</i> のメンバー・メソッド	メソッド・シグニチャーはユーザーによって定義されます。	<code>new MABGlobalScripts().gs()</code> としてアクセス可能です ( <i>gs()</i> はグローバル・スクリプト)。
フォーム	<ul style="list-style-type: none"> <li>MAB のフォームの「名前」プロパティの値と同じ名前のクラスとして生成されます。</li> <li>すべてのフォームは、<i>MABAppFrame</i> クラスの <i>panelTable</i> という名前のハッシュ・テーブルに作成され、追加されます。フォームを表すそれぞれのクラスには、<i>parent</i> という名前の変数があります。この変数は <i>MABAppFrame</i> を参照します。</li> </ul>	<i>java.awt.Panel</i>	<i>Form1</i> のフィールド <i>Field1Form1</i> にあるテキストは、 <code>((Form1)parent.panelTable.getField1Form1().getText());</code> として、 <i>Form2</i> (または任意のフォーム) 上のボタンについてのスクリプトからアクセスできます。
制御	MAB のフィールド制御の「名前」プロパティの値と同じ名前を持つクラスのメンバー	リスト制御: <i>java.awt.List</i> フィールド制御: <i>java.awt.TextField</i>	フォーム (またはフォーム上の任意のフィールド制御) で定義されたスクリプトは、フォームの名前を参照することでそのフォーム上のフィールド制御にアクセスできます。たとえば、ボタンのスクリプトで、 <i>Field1Form1.getText()</i> として、フィールド <i>Field1Form1</i> からテキストを取得できます。

表 14. 生成された MAB オブジェクト (続き)

MAB オブジェクト	生成されるオブジェクト	タイプ	スクリプトでのアクセス
フォームで定義されたスクリプト、またはフォーム上の任意の制御で定義されたスクリプト	フォームのクラスのメンバー関数		<i>Button1Form1</i> での最初のスクリプトは、 <code>void Button1Form1Script1 (ANTEvent ae)</code> として生成されます。

### データのアクセス:

アプリケーション実行中に、表から行が選択されると、その行にある列の値が内部バッファに保管されます。このバッファは、DB2 Everyplace からデータ・フェッチが行われるたびに更新されます。そのため、任意の列について、その列について最後に選択された値がバッファに含まれます。

このバッファは *DataStore* クラスにインプリメントされ、*getValue()* メソッドが列の現行値を戻します。例: *Object getValue(String key)* の場合、*key* は表名と列名を連結することによって取得されるストリングです。戻り値は、列について最後に選択された値か、またはその列を含む行がデータベースからまだフェッチされていない場合は *NULL* です。

そのため、*Table1* の *Column1* の現行値は、任意のフォームについてのスクリプトから、*parent.datastore.getValue("Table1Column1")* として取得できます。

アプリケーションのデータベース接続は、*MABDataAccess* クラス内の *con* という名前の *public* 静的変数として生成され、*MABDataAccess.con* としてスクリプトからアクセスできます。

### 関連概念:

157 ページの『Java コード・プラットフォームのためのスクリプト記述サポート』

### 関連タスク:

87 ページの『Mobile Application Builder 中のスクリプト』

## グローバル・スクリプトの作成

グローバル・スクリプトを使用して、他のスクリプトから呼び出される一般関数を作成できます。複数の場所でスクリプトを使用する場合にグローバル・スクリプトを推奨します。グローバル・スクリプトはイベント・スクリプトからのみ呼び出されます。

新規グローバル・スクリプトは、「プロジェクト」ペインのグローバル定義の下に追加され、プロジェクト全体で使用できます。

---

グローバル・スクリプトを作成するには:

1. 「プロジェクト」ペインで「**グローバル定義**」をクリックします。
  2. メインメニューから「**選択**」→「**新規スクリプトの作成**」の順にクリックします。スクリプト・ウィンドウが開きます。
  3. スクリプトを追加します。戻り値およびパラメーターは現状のままにします。
  4. オプション: スクリプト・ウィンドウにヘルパー機能を追加します。
  5. オプション: スクリプト・ウィンドウに制御を貼り付けます。
  6.  「**保管**」ボタンをクリックし、スクリプト・ウィンドウを閉じます。
- 

関連概念:

157 ページの『スクリプト記述サポート』

関連タスク:

48 ページの『スクリプト・プリファレンスの設定』

91 ページの『イベント・スクリプトの作成』

94 ページの『スクリプト・ウィンドウへのヘルパー機能の追加』

95 ページの『スクリプト・ウィンドウへの制御の追加』

## グローバル変数の定義

グローバル変数を使用して、すべてのスクリプトで可視にする変数を定義します。これらの変数はグローバルおよびイベント・スクリプトで呼び出されます。グローバル変数は、プロジェクトのすべてのグローバル変数が保管されている 1 つのウィンドウまたは 1 つのファイルです。

グローバル変数のサポートは、ターゲット・プラットフォームによって異なります。

---

グローバル変数を定義するには:

1. 「プロジェクト」ペインで「**グローバル定義**」をクリックします。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページにある、グローバル変数の隣りの  をクリックし、「グローバル変数」ウィンドウを開きます。
3. アプリケーションのグローバル変数を定義します。
4. 「**OK**」をクリックして「グローバル変数」ウィンドウを閉じます。

---

#### 関連概念:

157 ページの『スクリプト記述サポート』

#### 関連タスク:

48 ページの『スクリプト・プリファレンスの設定』

## グローバル・スクリプトの編集または削除

すべてのグローバル・スクリプトが、「プロジェクト」ペインの「グローバル定義」の下にあります。

---

#### グローバル・スクリプトを編集または削除するには:

1. 編集するグローバル・スクリプトを「プロジェクト」ペイン. で選択します。
2. メインメニューから「選択」→「スクリプトの編集」または「選択」→「スクリプトの削除」をクリックします。

---

#### 関連概念:

157 ページの『スクリプト記述サポート』

#### 関連タスク:

48 ページの『スクリプト・プリファレンスの設定』

89 ページの『グローバル・スクリプトの作成』

## イベント・スクリプトの作成

イベント・スクリプトを使用して、ボタン・クリックやフォーム初期化といった特定のイベントが起きたときに、特定のスクリプトを実行することができます。イベント・スクリプトを作成するには、以下のステップに従ってください。スクリプトをグローバル定義の下に定義し、これらのスクリプトをイベント・スクリプトから呼び出すことを推奨します。

すべてのイベント・スクリプトが、特定のフォームおよび制御と関連しています。すべてのフォームまたは制御で利用できるスクリプトが必要な場合は、グローバル・スクリプトを作成してください。

---

#### イベント・スクリプトを作成するには:

1. フォームまたは制御のいずれかを選択します。
2. メインメニューから「選択」->「新規スクリプトの作成」を選択します。スクリプト・ウィンドウが開きます。 C 関数スケルトンまたは標準 Java 署名のいずれ

れかが、既にウィンドウに表示されています。どちらが表示されるかは、ご使用になっているターゲット・プラットフォームによって決まります。スクリプト・ウィンドウの最上部にある C 関数定義には、スクリプトに追加している制御の名前が含まれています。この名前の文字は、GNU C コンパイラーで使用可能な文字セットにはなく、下線文字に変更されます。

3. スクリプトを追加します。C コード・スクリプトの場合は、戻り値およびパラメーターは現状のままにします。
4. オプション: C コード・スクリプトの場合、スクリプト・ウィンドウにヘルパー機能を追加します。
5. オプション: C コード・スクリプトの場合、スクリプト・ウィンドウに制御を貼り付けます。
6. オプション: イベント・スクリプトの名前を変更します。イベント・スクリプトを名前変更しないと、自動的に変更されます。
7.  「保管」 ボタンをクリックし、スクリプト・ウィンドウを閉じます。
8. スクリプトを呼び出すイベントを設定します。
  - a. フォームまたは制御を選択した状態で、プロパティおよび「プロパティおよびイベント」ペインの「イベント」タブをクリックします。イベントのリストに、新規スクリプトが表示されます。
  - b. 新規スクリプトの隣りにある、左端の列のイベントのリストからイベントを選択します。

---

#### 関連概念:

157 ページの『スクリプト記述サポート』

#### 関連タスク:

48 ページの『スクリプト・プリファレンスの設定』

89 ページの『グローバル・スクリプトの作成』

『イベント・スクリプトの編集または削除』

93 ページの『イベント・スクリプトの名前変更』

94 ページの『スクリプト・ウィンドウへのヘルパー機能の追加』

95 ページの『スクリプト・ウィンドウへの制御の追加』

#### 関連参照情報:

153 ページの『イベントの説明』

## イベント・スクリプトの編集または削除

イベント・スクリプトは、フォームおよび制御とのみ関連しています。特定のイベント・スクリプトを見つけて編集するには、以下のステップに従ってください。

---

イベント・スクリプトを編集または削除するには:

1. 関連するスクリプトと一緒にフォームまたは制御を選択します。
2. メインメニューから「選択」→「スクリプトの編集」→「スクリプト名」または「スクリプトの削除」→「スクリプト」→「スクリプト名」の順にクリックします。

---

#### 関連概念:

157 ページの『スクリプト記述サポート』

#### 関連タスク:

48 ページの『スクリプト・プリファレンスの設定』

91 ページの『イベント・スクリプトの作成』

『イベント・スクリプトの名前変更』

95 ページの『スクリプト・ウィンドウへの制御の追加』

94 ページの『スクリプト・ウィンドウへのヘルパー機能の追加』

## イベント・スクリプトの名前変更

Mobile Application Builder でのスクリプトの処理方法によっては、イベント・スクリプトの名前を変更する場合に、アクションの名前変更を行うスクリプトを実行する必要があります。イベント・スクリプトを名前変更しないと、自動的に変更されます。

---

#### イベント・スクリプトを名前変更するには:

1. スクリプト・ウィンドウで「編集」→「スクリプトの名前変更」をクリックします。
2. 新規スクリプト名を入力します。スペースまたは無効な文字を使用しないでください。
3. 「OK」をクリックします。無効な名前が入力されると、その名前は保管されません。名前を訂正して「OK」を再度クリックしてください。

---

#### 関連概念:

157 ページの『スクリプト記述サポート』

#### 関連タスク:

48 ページの『スクリプト・プリファレンスの設定』

91 ページの『イベント・スクリプトの作成』

92 ページの『イベント・スクリプトの編集または削除』

95 ページの『スクリプト・ウィンドウへの制御の追加』

94 ページの『スクリプト・ウィンドウへのヘルパー機能の追加』

## スクリプト・ウィンドウへのヘルパー機能の追加

スクリプト記述の際に使用可能な「ヘルパーの貼り付け」ウィンドウによって、スクリプトに関数を追加することができます。Mobile Application Builder と共に提供されるヘルパー機能とプロジェクト・ユーザー作成グローバル・スクリプトは、貼り付けて使用することができます。ヘルパー機能は、指定されたタスクを実行するために必要なすべての Palm API 呼び出しを実行します。

ヘルパー機能のサポートは、ターゲット・プラットフォームによって異なります。

現在、以下の 2 つのヘルパー機能が製品と共に提供されています:

### MAB\_getDateString

現在のシステム日付を取得します。巡回医療サービスのサンプル・アプリケーションで使用されるデータベースに合わせたフォーマットになっています。

### MAB\_getTimeString

現在のシステム時刻を取得します。巡回医療サービスのサンプル・アプリケーションで使用されるデータベースに合わせたフォーマットになっています。

提供されているヘルパー機能を使用したい場合は、「ヘルパーの貼り付け」ウィンドウを使って、使用したい機能呼び出しを選択し、そのコードがエディターに挿入されるようにしなければなりません。ウィンドウを使用せずに呼び出しを手作業でヘルパーに挿入すると、生成ステップで失敗します。機能呼び出しがスクリプト・ウィンドウに挿入された後は、そのコードを変更することができます。

---

### ヘルパー機能をスクリプト・ウィンドウに追加するには:

1. スクリプト・ウィンドウで、機能を始めたい場所にカーソルを持っていきます。
2. メインメニューまたはツールバーから「編集」→ 「ヘルパーの貼り付け」をクリックします。アイコンおよびメニュー項目が選択可能になっていない場合は、この機能が現在のターゲット・プラットフォームではサポートされないことを意味します。
3. オプション: 提供されているラジオ・ボタンを使用して、Mobile Application Builder ヘルパー機能とユーザー定義グローバル・スクリプトを切り替えます。
4. 使用するヘルパー機能を選択します。選択した機能の説明が、説明域で提供されます。
5. 「OK」をクリックします。選択した機能が、ヘルパーの貼り付けを呼び出した時点でカーソルがあった場所を始まりとして、スクリプト内に置かれます。

---

### 関連概念:

157 ページの『スクリプト記述サポート』

### 関連タスク:

89 ページの『グローバル・スクリプトの作成』

91 ページの『イベント・スクリプトの作成』

『スクリプト・ウィンドウへの制御の追加』

## スクリプト・ウィンドウへの制御の追加

スクリプト・ウィンドウにある「部品の貼り付け」ボタンを使用して、フォーム内にドロップされた制御の特定プロパティにアクセスする、ヘルパー機能をいくつか追加することができます。現在このウィンドウでサポートするのは、データ・ソース・プロパティおよびデータ・フィールド・プロパティの値が有効なフィールド制御を持つ、テキストの取得および設定だけです。

ヘルパー機能のサポートは、ターゲット・プラットフォームによって異なります。

---

### 制御をスクリプト・ウィンドウに貼り付けるには:

1. スクリプト・ウィンドウで、機能を始めたい場所にカーソルを持っていきます。
2. メインメニューまたはツールバーから「編集」→ 「部品の貼り付け」をクリックします。「部品の貼り付け」ウィンドウが開きます。アイコンおよびメニュー項目が選択可能になっていない場合は、この機能が現在のターゲット・プラットフォームではサポートされないことを意味します。
3. 使用する部品を選択します。現行のフォームの部品だけがリストに表示されません。
4. 部品に関連付けるアクションを選択します。各部品ごとに適用できるアクションだけが表示されます。
5. 「OK」をクリックします。スクリプト・ウィンドウに機能が追加されました。

---

### 関連概念:

157 ページの『スクリプト記述サポート』

### 関連タスク:

89 ページの『グローバル・スクリプトの作成』

91 ページの『イベント・スクリプトの作成』

94 ページの『スクリプト・ウィンドウへのヘルパー機能の追加』

## フォーム・イベントの SQL を変更

フォームはフォーム自体に関連する SQL ステートメントを持つ場合があります。フォームが、「データ・ソース」や「データ・フィールド」に定義されたプロパティを持つ制御を有している場合は、関連する SELECT ステートメントがあり、このステートメントが、フォームを取り込むためにデータを検索します。データベース・アクション (CREATE/DELETE/UPDATE レコード) がフォームに関連付けられている場合は、それに対応する SQL ステートメントがあり、このステートメントがターゲット・データベース表を変更します。この機能により、Mobile Application

Builder で使用可能な基本機能の枠を超えた SQL ステートメントのカスタマイズが可能になります。たとえば、関連する演算子を追加したり検索条件を変更したりして、Mobile Application Builder で生成した SQL ステートメントで使用することができます。

#### SQL ステートメントの手動変更についての注意:

- 「**編集 SQL の使用**」オプションを使用して SQL ステートメントを編集して保管した後は (以下のステップで説明されています)、それ以降にプロジェクトを保管してもその SQL は MAB によって再生成されません。
- 次にプロジェクトを保管したときに SQL ステートメントが生成されるように MAB にフラグを立てるには、この SQL ステートメントについて「**デフォルト生成 SQL の使用**」オプションを選択します。
- 手動で編集された SQL は、この SQL を使用する制御またはフォームが削除されてもプロジェクトに残ります。変更された SQL は手動で削除できます。

以下のステップに従ってフォームで SQL を編集します。

---

#### フォーム・イベントで SQL を変更するには:

1. 関連する SQL を持つフォームをクリックします。
2. メインメニューから「**選択**」->「**SQL の編集**」をクリックします。または、フォーム上で右マウス・ボタン・クリックしてから、ポップアップ・メニューの「**SQL の編集**」をクリックします。
3. 編集したい SQL を選択します。ウィンドウがオープンし、SQL ステートメントを表示します。
4. 「**編集 SQL の使用**」ラジオ・ボタンをクリックします。このオプションを選択しなかった場合、変更は保管されず、SQL は次回にプロジェクトを保管したときに MAB によって再生成されます。
5. 必要な変更を行います。列名や列の順序は変更しないでください。列情報の変更はエラーの原因となります。
6. 「**OK**」をクリックして変更を保管し、ウィンドウを閉じます。

---

#### 関連タスク:

『フォーム・イベントの SQL を削除』

## フォーム・イベントの SQL を削除

手動で編集された SQL は、この SQL を使用する制御またはフォームが削除されてもプロジェクトに残ります。変更された SQL をプロジェクトから削除するには、次のステップを実行します。MAB 生成 SQL を削除することはできません。

---

#### フォーム・イベントの SQL を削除するには:

1. 関連する SQL を持つフォームをクリックします。
2. メインメニューから「選択」->「SQL の削除」をクリックします。または、フォーム上で右マウス・ボタン・クリックしてから、ポップアップ・メニューの「SQL の削除」をクリックします。「SQL の削除」メニュー項目は、手動で編集された、選択されたフォームに関連付けられている SQL が存在する場合にのみ使用できます。
3. 削除する SQL を選択します。
4. 確認ダイアログで「OK」をクリックして、選択した SQL を削除します。

---

#### 関連タスク:

95 ページの『フォーム・イベントの SQL を変更』

---

## ホスト変数の使用

ホスト変数は、アプリケーション・ホスト・プログラムの組み込み SQL ステートメントによって参照される変数です。ホスト変数は、アプリケーション・プログラムのプログラミング変数であり、データベース内の表とアプリケーション・プログラム作業域との間でデータを送信するための基本メカニズムです。

ホスト変数を使用して、ユーザー入力を基にしたデータ抽出、および複数列条件検査を基にした表でのリンクが可能です。

注: 定義済みホスト変数をスクリプトで使用する場合、ホスト変数を必要データ・タイプにキャストする必要があります。

---

#### ホスト変数を使用するには:

1. プロジェクト・レベルでホスト変数を定義します。
2. フィールド、リスト制御、または選択リスト制御の「ホスト変数の更新」プロパティを設定します。
3. フォーム、リスト制御、または選択リスト制御の「選択基準」プロパティを設定します。

---

#### 関連参照情報:

128 ページの『プロパティの制御』

## ホスト変数の定義

以下のステップは、ホスト変数を使用する、より大きなタスクの一部です。ホスト変数を定義するステップを完了したら、『ホスト変数の使用』に戻ります。

---

ホスト変数を定義するには:

1. 「プロジェクト」ペインで「**グローバル定義**」をクリックします。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「**ホスト変数**」の隣りにある  をクリックし、「ホスト変数」ウィンドウを開きます。
3. 「**追加**」をクリックし、新規ホスト変数を定義します。
4. オープンしたウィンドウで、ホスト変数の名前を入力します。すべての変数が、以下の SQL データ・タイプである必要があります。
  - Palm の場合は CHAR
  - Java の場合は STRINGインプリメンテーションは非表示になります。ホスト変数名に DBCS (2 バイト) 文字を含めることはできません。
5. 「**OK**」をクリックします。新しいホスト変数がホスト変数のリストに追加されます。
6. オプション: ステップ 3 と 4 を繰り返して、その他のホスト変数を追加します。
7. 「**OK**」をクリックし、「ホスト変数」ウィンドウを閉じます。

『ホスト変数の使用』に戻ります。

---

## フィールド制御の「ホスト変数の更新」プロパティの設定

以下のステップは、ホスト変数を使用する、より大きなタスクの一部です。フィールド制御の「**ホスト変数の更新**」プロパティを設定するステップを完了したら、『ホスト変数の使用』に戻ります。

---

フィールド制御の「ホスト変数の更新」プロパティを設定するには:

1. フィールド制御を選択します。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「**ホスト変数の更新**」リストからホスト変数の名前を選択し、情報をフィールドに入力したときにどのホスト変数を更新するかを定義します。

『ホスト変数の使用』に戻ります。

---

## リスト制御または選択リスト制御の「ホスト変数の更新」プロパティの設定

以下のステップは、ホスト変数を使用する、より大きなタスクの一部です。リスト制御または選択リスト制御の「ホスト変数の更新」プロパティを設定するステップを完了したら、『ホスト変数の使用』に戻ります。

---

リスト制御または選択リスト制御の「ホスト変数の更新」プロパティを設定するには:

1. リスト制御または選択リスト制御を選択します。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「ホスト変数の更新」の隣りにある  をクリックし、リストまたは選択リストの選択結果で更新されるホスト変数を定義できるウィンドウを開きます。
3. 「データ・ソース」列で、リスト制御または選択リスト制御のデータ・ソースとして定義されている表の中から表定義を選択します。「データ・ソース」列には、データ・ソースとして定義された表のみが表示されます。表がない場合は、リストまたは選択リストのデータ・ソースを定義したかどうかを確認してください。
4. 「データ・フィールド」列で、選択したデータ・ソース表から列を選択します。この列に表示される列は、前述のステップで選択したデータ・ソースからのものです。
5. 上記のデータ・ソースおよびデータ・フィールドと関連付けるホスト変数を選択します。何も表示されない場合は、データ・ソースとデータ・フィールドが選択されていること、およびホスト変数が定義されていることを確認してください。
6. 「OK」をクリックし、ウィンドウを閉じます。

『ホスト変数の使用』に戻ります。

---

## フォームの選択基準の設定

以下のステップは、ホスト変数を使用する、より大きなタスクの一部です。フォームの「選択基準」を設定するステップを完了したら、『ホスト変数の使用』に戻ります。

---

フォームの「選択基準」を設定するには:

1. フォームを選択します。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「選択基準」フィールドの隣りにある  をクリックし、どのデータベース列をどのホスト変数に関連付けるかを指定できるウィンドウを開きます。これにより、フォ

ームのパラメーター化された SQL 照会が生成されます。結合 SELECT ステートメントが生成された場合、「ホスト変数」と関連した列を基にして追加されず。

3. 「データ・ソース」列で、プロジェクトにインポートした表の中から表定義を選択します。表がない場合は、プロジェクトにインポートされていることを確認してください。
4. 「データ・フィールド」列で、選択したデータ・ソース表から列を選択します。この列に表示される列は、前述のステップで選択したデータ・ソースからのものです。
5. 上記のデータ・ソースおよびデータ・フィールドと関連付けるホスト変数を選択します。何も表示されない場合は、データ・ソースとデータ・フィールドが選択されていること、およびホスト変数が定義されていることを確認してください。
6. 「OK」をクリックし、ウィンドウを閉じます。

『ホスト変数の使用』に戻ります。

---

## リスト制御または選択リスト制御の選択基準の設定

以下のステップは、ホスト変数を使用する、より大きなタスクの一部です。リスト制御または選択リスト制御の「選択基準」を設定するステップを完了したら、『ホスト変数の使用』に戻ります。

---

リスト制御または選択リスト制御の「選択基準」を設定するには:

1. リスト制御または選択リスト制御を選択します。
2. 「プロパティおよびイベント」ペインの「プロパティ」タブで、「**選択基準**」フィールドの隣にある  をクリックし、どのデータベース列をどのホスト変数に関連付けるかを指定できるウィンドウを開きます。これにより、リスト制御または選択リスト制御の SQL 照会が変更されます。
3. 「データ・ソース」列で、リスト制御または選択リスト制御のデータ・ソースとして定義されている表の中から表定義を選択します。「データ・ソース」列には、データ・ソースとして定義された表のみが表示されます。表がない場合は、リストまたは選択リストのデータ・ソースを定義したかどうかを確認してください。
4. 「データ・フィールド」列で、選択したデータ・ソース表から列を選択します。この列に表示される列は、前述のステップで選択したデータ・ソースからのものです。
5. 上記のデータ・ソースおよびデータ・フィールドと関連付けるホスト変数を選択します。何も表示されない場合は、データ・ソースとデータ・フィールドが選択されていること、およびホスト変数が定義されていることを確認してください。
6. 「OK」をクリックし、ウィンドウを閉じます。

『ホスト変数の使用』に戻ります。

---

## アプリケーションへの印刷機能の追加

Mobile Application Builder は、Palm OS デバイスに、限定された印刷サポートを提供しています。「プリファレンス」で各プロジェクトのためのデフォルト印刷ソリューションを設定することによって、印刷機能にアクセスすることができます。

---

### 関連参照情報:

158 ページの『印刷サポート』

159 ページの『印刷 API』

## PalmPrint のインストール

PalmPrint ソリューションを使用して Mobile Application Builder の印刷機能を使用するには、事前にこのソリューションを Stevens Creek Software から入手しておかなければなりません。

印刷機能のサポートは、ターゲット・プラットフォームによって異なります。

---

### PalmPrint をインストールするには:

1. Stevens Creek Software から PalmPrint.prc ファイルを入手します。
2. PalmPrint.prc ファイルをエミュレーターまたは モバイル・デバイスにインストールします。Stevens Creek Software の指示に従ってください。

詳細については、Stevens Creek Software の Web サイトを参照してください。

---

### 関連参照情報:

158 ページの『印刷サポート』

159 ページの『印刷 API』

## フォームへの印刷ボタンの追加

インプリメントしたい印刷ソリューションを選択したら、印刷ボタンを追加して、フォームに印刷機能を追加します。

印刷機能のサポートは、ターゲット・プラットフォームによって異なります。

---

### フォームに印刷ボタンを追加するには:

1. 制御パレットを使用して、フォームにボタンを追加します。

2. 「フォーム」ペイン上の新しいボタンをクリックし、「プロパティーおよびイベント」ペインで「プロパティー」タブをクリックします。
3. このボタンの「表題」に印刷と入力 (またはカスタマイズ) します。
4. 「イベント」ページで、ボタンに「印刷」イベントを追加します。
  - a. 「アクションなし」ラベルの隣りにある  をクリックし、「イベント・アクションおよびターゲットの選択」ウィンドウを開きます。
  - b. 「アクション」リストで「PrintForm」をクリックします。「PrintForm」アクションが使用できない場合、現行のターゲット・プラットフォームで印刷機能はサポートされていません。
  - c. 「OK」をクリックします。「イベント・アクションおよびターゲットの選択」ウィンドウが閉じます。

---

#### 関連タスク:

101 ページの『PalmPrint のインストール』  
『フォームへの印刷メニュー項目の追加』

#### 関連参照情報:

158 ページの『印刷サポート』  
159 ページの『印刷 API』  
131 ページの『ボタン・プロパティー』

## フォームへの印刷メニュー項目の追加

インプリメントしたい印刷ソリューションを選択したら、印刷メニュー項目を追加して、フォームに印刷機能を追加します。

印刷機能のサポートは、ターゲット・プラットフォームによって異なります。

---

#### フォームに印刷メニュー項目を追加するには:

1. 「プロジェクト」ペインで「メニュー」項目を右マウス・ボタン・クリックし、「新規メニューの追加」をクリックします。「プロジェクト」ペインの「メニュー」セクションに、新しいメニュー項目が作成されます。
2. 「プロジェクト」ペインで新しいメニュー項目をクリックし、「プロパティーおよびイベント」ペインで「プロパティー」タブをクリックします。
3. 「メニュー項目テキスト」に印刷と入力 (またはカスタマイズ) し、「アクセラレーター文字」に p と入力 (またはカスタマイズ) します。
4. 「イベント」ページで、「印刷」メニューにイベントを追加します。
  - a. 右にある  をクリックし、「イベント・アクションおよびターゲットの選択」ウィンドウを開きます。

- b. 「アクション」リストで「PrintForm」をクリックします。「PrintForm」アクションが使用できない場合、現行のターゲット・プラットフォームで印刷機能はサポートされていません。
  - c. 「OK」をクリックします。「イベント・アクションおよびターゲットの選択」ウィンドウが閉じます。
5. 「プロジェクト」ペインで「メニュー・バー」項目を右マウス・ボタン・クリックし、「新規メニュー・バーの追加」をクリックします。
  6. 「プロジェクト」ペインで新しいメニュー・バー項目をクリックし、「プロパティおよびイベント」ペインで「プロパティ」タブをクリックします。
  7. 「プルダウン」にメインメニューと入力 (またはカスタマイズ) します。
  8. 「メニュー項目 ID」プロパティに、ステップ 1 から 4 で作成したメニューの名前を設定します。
  9. 印刷メニューを追加するフォームを選択し、「プロパティおよびイベント」ペインの「プロパティ」タブをクリックします。
  10. フォームの「メニュー ID」プロパティに、ステップ 5 から 8 で作成したメニュー・バーの名前を設定します。
  11. オプション: ステップ 8 から 10 を繰り返して、その他のフォームのメニュー・バーに印刷メニュー項目を追加します。

---

#### 関連タスク:

- 101 ページの『PalmPrint のインストール』
- 101 ページの『フォームへの印刷ボタンの追加』

#### 関連参照情報:

- 158 ページの『印刷サポート』
- 159 ページの『印刷 API』
- 151 ページの『メニュー・プロパティ』
- 151 ページの『メニュー・バー・プロパティ』

---

## アプリケーションへのバーコード・スキャン機能の追加

Mobile Application Builder は、このオプションをサポートするデバイスでのハードまたはソフトのバーコード・スキャンをサポートしています。ハード・スキャン装置には、装置に組み込まれたスキャン・ボタンがあります。ソフト・スキャン装置では、ユーザーがスキャン・ボタンを作成する必要があります。フィールドをスキャン認識と指定することで、フォーム上のフィールドにバーコード・データをスキャンできます。ユーザーがバーコード情報を Palm アプリケーションのフォームにスキャンするには、スキャン認知フォーム上の目的のフィールドにカーソルを置いて、ハードまたはソフト・スキャンを実行します。WinCE アプリケーションの場合、ユーザーはスキャンを実行するだけでよく、カーソルを置く必要はありません。スキャンが正常に実行された後、そのフィールドにデータ・ストリングがコピーされます。

WinCE バーコード・スキャン・アプリケーションの場合、アプリケーションをビルドする前に、CrEme Plus JVM を取得して、MAB プリファレンスでこれを選択する必要があります。CrEme Plus v 3.21 JVM のバグのため、WinCE アプリケーションでバーコード ScannerClose イベントを開始すると、アプリケーションがクローズします。JVM でバグが修正されるまでこの問題を回避するには、このイベントをアプリケーションに追加しないでください。

バーコード・スキャン・インプリメンテーションを含むアプリケーションはエミュレーター上でテストできません。これらのアプリケーションは、実際のモバイル・デバイスを使用してテストする必要があります。

アプリケーションにスキャン機能を追加するには、バーコード・データを受け取るフォームにイベントおよびアクションを追加します。ScannerDecode イベントは必須です。その他のイベントはオプションですが、バーコード・スキャンの動作をさらにカスタマイズするために追加することができます。

バーコード・スキャンのサポートは、ターゲット・プラットフォームによって異なります。

---

## Mobile Application Builder バーコード・スキャン・タスク:

『フォームへのソフト・スキャン・バーコード・スキャン機能の追加』  
106 ページの『フォームへのハード・スキャン・バーコード・スキャン機能の追加』

---

### 関連参照情報:

162 ページの『バーコード・スキャン API』  
169 ページの『Palm アプリケーションでサポートされる 1 次元バーコード・タイプ』  
153 ページの『イベントの説明』  
155 ページの『アクションの説明』

## フォームへのソフト・スキャン・バーコード・スキャン機能の追加

ソフト・バーコード・スキャンのサポートは、ターゲット・プラットフォームによって異なります。CrEme Plus v 3.21 JVM のバグのため、WinCE アプリケーションでバーコード ScannerClose イベントを開始すると、アプリケーションがクローズします。この問題を回避するには、このイベントをアプリケーションに追加しないようにしてください。

---

### フォームにソフト・スキャン機能を追加するには:

1. バーコード・スキャン・データを受け取るためのフォームを選択します。フォームにはフィールド制御が入っていなければなりません。
2. 「イベント」ページで、1 つ目のスキャン・イベントをフォームに追加します。

- a. 左側の「ScannerInit」イベントを選択します。「イベント・アクションおよびターゲットの選択」ウィンドウが開きます。
  - b. このイベントのアクションに「InitializeScanner」を選択します。
  - c. 「OK」をクリックして、「イベント・アクションおよびターゲットの選択」ウィンドウを閉じます。
3. 「イベント」タブで、2 番目のスキャン・イベントをフォームに追加します。
- a. 左側の「ScannerDecode」イベントを選択します。「イベント・アクションおよびターゲットの選択」ウィンドウが開きます。
  - b. このイベントのアクションに「DecodeScanTo」を選択します。  
「DecodeScanTo」アクションが使用できない場合は、選択したフォームにフィールド制御が入っていることを確認してください。
  - c. このフォーム上の、このアクションのターゲットとして、スキャンを認識する必要のあるフィールド (1 つまたは複数) を選択します。
  - d. 「OK」をクリックして、「イベント・アクションおよびターゲットの選択」ウィンドウを閉じます。
4. オプション (Palm アプリケーションのみ): 「イベント」ページで、「NoDecode」イベントをフォームに追加します。このイベントのアクションを選択するか、「アクションなし」を指定してデフォルト・エラー・メッセージを生成します。
5. 「イベント」タブで、最後のスキャン・イベントをフォームに追加します。
- a. 左側の「ScannerClose」イベントを選択します。「イベント・アクションおよびターゲットの選択」ウィンドウが開きます。
  - b. このイベントのアクションに「DeInitScanner」を選択します。
  - c. 「OK」をクリックして、「イベント・アクションおよびターゲットの選択」ウィンドウを閉じます。
6. 制御パレットを使用して、新規ボタンをフォームに追加します。
7. 「プロパティ」ページで、このボタンの「表題」にスキャン (またはカスタマイズ) と入力します。
8. 「イベント」タブで、スキャン・イベントを新規ボタンに追加します。
- a. 「アクションなし」ラベルの隣りにある「省略符号ボタン」  をクリックし、「イベント・アクションおよびターゲットの選択」ウィンドウをオープンします。
  - b. このイベントのアクションに「SoftScan」を選択します。
  - c. 「OK」をクリックして、「イベント・アクションおよびターゲットの選択」ウィンドウを閉じます。

---

#### 関連タスク:

106 ページの『フォームへのハード・スキャン・バーコード・スキャン機能の追加』

#### 関連参照情報:

162 ページの『バーコード・スキャン API』

169 ページの『Palm アプリケーションでサポートされる 1 次元バーコード・タイプ』

153 ページの『イベントの説明』

155 ページの『アクションの説明』

## フォームへのハード・スキャン・バーコード・スキャン機能の追加

バーコード・スキャンのサポートは、ターゲット・プラットフォームによって異なります。CrEme Plus v 3.21 JVM のバグのため、WinCE アプリケーションでバーコード `ScannerClose` イベントを開始すると、アプリケーションがクローズします。この問題を回避するには、このイベントをアプリケーションに追加しないようにしてください。

---

### フォームにハード・スキャン機能を追加するには:

1. バーコード・スキャン・データを受け取るためのフォームを選択します。
2. 「イベント」ページで、1 つ目のスキャン・イベントをフォームに追加します。
  - a. 左側の「ScannerInit」イベントを選択します。「イベント・アクションおよびターゲットの選択」ウィンドウが開きます。
  - b. このイベントのアクションに「InitializeScanner」を選択します。
  - c. 「OK」をクリックして、「イベント・アクションおよびターゲットの選択」ウィンドウを閉じます。
3. 「イベント」ページで、2 番目のスキャン・イベントをフォームに追加します。
  - a. 左側の「ScannerDecode」イベントを選択します。「イベント・アクションおよびターゲットの選択」ウィンドウが開きます。
  - b. このイベントのアクションに「DecodeScanTo」を選択します。
  - c. このフォーム上の、このアクションのターゲットとして、スキャンを認識する必要のあるフィールド (1 つまたは複数) を選択します。
  - d. 「OK」をクリックして、「イベント・アクションおよびターゲットの選択」ウィンドウを閉じます。
4. オプション (Palm アプリケーションのみ): 「イベント」ページで、「NoDecode」イベントをフォームに追加します。このイベントのアクションを選択するか、「アクションなし」を指定してデフォルト・エラー・メッセージを生成します。
5. 「イベント」タブで、最後のスキャン・イベントをフォームに追加します。
  - a. 左側の「ScannerClose」イベントを選択します。「イベント・アクションおよびターゲットの選択」ウィンドウが開きます。
  - b. このイベントのアクションに「DeInitScanner」を選択します。
  - c. 「OK」をクリックして、「イベント・アクションおよびターゲットの選択」ウィンドウを閉じます。

---

### 関連タスク:

104 ページの『フォームへのソフト・スキャン・バーコード・スキャン機能の追加』

**関連参照情報:**

162 ページの『バーコード・スキャン API』

169 ページの『Palm アプリケーションでサポートされる 1 次元バーコード・タイプ』

153 ページの『イベントの説明』

155 ページの『アクションの説明』

---

## アプリケーションへの同期機能の追加

DB2 Everyplace 同期サーバーの機能を使用して、アプリケーションに同期機能を追加することができます。アプリケーションに同期機能を追加する前に、同期サーバーをインストールし、構成しなければなりません。

ユーザーが同期を開始すると、構成ストア内にあり使用可能になっているすべてのサブスクリプション・セットが同期化されます。デバイス上の DB2 Everyplace データベースのロケーションが、構成ストアのロケーションとして使用されます。

同期サーバー・サブスクリプション・セットおよび構成ストアについては、「*IBM DB2 Everyplace 同期サーバー 管理ガイド*」を参照してください。

同期化を行うには、実行時にユーザーからいくつかの入力パラメーターを入手する必要があります。同期を試行する前に、これらのパラメーターをユーザーから入手するために、同期プリファレンス・ダイアログをアプリケーションに明示的に追加することをお勧めします。同期アクションが試行され、これらのプリファレンスが事前に設定されていなかった場合は、Mobile Application Builder が自動的にこのダイアログをいったん呼び出します。しかし、そのダイアログも明示的にアプリケーションに追加されていなかった場合、ユーザーがこれらのプリファレンスを変更または訂正する方法がありません。

---

### 同期機能を制御に追加するには:

1. 同期機能を追加したい制御を選択します。
2. 同期イベントを制御に追加します。
  - a. 「イベント」ページの左側でイベントを選択します。「イベント・アクションおよびターゲットの選択」ウィンドウが開きます。
  - b. このイベントのアクションとして「同期アプリケーション」を選択します。

### 同期プリファレンス・ダイアログを表示するために、制御にイベントを追加するには:

1. 制御を選択します。
2. 「イベント」ページの左側でイベントを選択します。「イベント・アクションおよびターゲットの選択」ウィンドウが開きます。
3. このイベントのアクションとして「表示」を選択し、アクションのターゲットとして「同期プリファレンス」ダイアログを選択します。

**関連タスク:**

60 ページの『フォームへの制御の追加』

**関連参照情報:**

153 ページの『イベントの説明』

155 ページの『アクションの説明』

---

## Mobile Application Builder アプリケーションのビルド

ターゲットのモバイル・デバイスが読み取ることができる 1 つのファイルに制御とリソースをすべてコンパイルするようにアプリケーションをビルドします。

---

**関連タスク:**

113 ページの『Mobile Application Builder アプリケーションのテスト』

**関連参照情報:**

110 ページの『コード生成/ビルド・ファイル記述』

## プリプロセッサ・ステートメントの定義

プリプロセッサ・ステートメントを使用して、コード生成中に使用するステートメントを定義します。「プリプロセッサ・ステートメント」ウィンドウに入力したステートメントはコードの先頭に挿入されます。

プリプロセッサ・ステートメントのサポートは、ターゲット・プラットフォームによって異なります。

---

**プリプロセッサ・ステートメントを定義するには:**

1. 「プロジェクト」ペインで「グローバル定義」をクリックします。
2. 「プロパティおよびイベント」ペインの「プロパティ」ページで、「プリプロセッサ・ステートメント」の隣りにある  をクリックして、「プリプロセッサ・ステートメント」ウィンドウを開きます。
3. コード生成中に使用するプリプロセッサ・ステートメントを定義します。
4. 「OK」をクリックし、ウィンドウを閉じます。

---

**関連参照情報:**

110 ページの『コード生成/ビルド・ファイル記述』

## アプリケーションのビルド

ターゲットのモバイル・デバイスが読み取ることができる 1 つのファイルに制御とリソースをすべてコンパイルするようにアプリケーションをビルドします。

アプリケーションをビルドする前に、Mobile Application Builder 環境が正しくセットアップされていることを確認します。詳細については、『セットアップおよび構成』を参照してください。

固有アプリケーション ID が必要なターゲット・アプリケーションの場合は、アプリケーションが固有アプリケーション ID を持っていることを確認してください。この ID がないと、アプリケーションは正常に動作しないことがあります。

---

### アプリケーションをビルドするには:

1. メニューまたはツールバーから、「ビルド」 ->  「ビルド」の順にクリックします。「アプリケーションのビルド」状況ダイアログに、ビルドの進行状況が表示されます。このステップで、アプリケーションが生成されます。「ビルド完了」ダイアログに、完了したビルドの状況が表示されます。完了すると、PRC ファイルが、複数の中間ファイルと共に、プロジェクト・サブディレクトリーに作成されます。
2. エラーが発生した場合は、『トラブルシューティング』を参照して解決方法を探してください。

---

### 関連タスク:

- 113 ページの『Mobile Application Builder アプリケーションのテスト』  
『メッセージ・ログの表示およびクリア』
- 110 ページの『メッセージ・ログの保管』

### 関連参照情報:

- 110 ページの『コード生成/ビルド・ファイル記述』

## メッセージ・ログの表示およびクリア

メッセージ・ログには、現行セッションで行われた各ビルドに関する情報が入っています。メッセージ・ログは、手動でクリアすることもできます。最新の情報は、ログの一番下にあります。

---

### メッセージ・ログを表示するには:

1. メインメニューから、「ヘルプ」->「ログ・ファイルの表示」の順にクリックします。「メッセージ・ログ」ウィンドウが開きます。

### メッセージ・ログをクリアするには:

1. 「メッセージ・ログ」ウィンドウで、「クリア」をクリックします。

---

**関連タスク:**

109 ページの『アプリケーションのビルド』  
『メッセージ・ログの保管』

**関連参照情報:**

『コード生成/ビルド・ファイル記述』  
170 ページの『トラブルシューティング』

## メッセージ・ログの保管

解決できないエラーが発生した場合は、メッセージ・ログ・ファイルを保管します。ログは Mobile Application Builder をクローズするたびに消去されるので、セッションでエラーが発生した場合は、そのセッションを終了する前にログを保管しておくことが大切です。

---

**メッセージ・ログを保管するには:**

1. ../ApplicationBuilder/bin/ ディレクトリーで db2emab.log ログ・ファイルを探します。
2. ログ・ファイルのコピーを同じディレクトリーに作成します。

---

**関連タスク:**

109 ページの『アプリケーションのビルド』  
109 ページの『メッセージ・ログの表示およびクリア』

**関連参照情報:**

『コード生成/ビルド・ファイル記述』  
170 ページの『トラブルシューティング』

## コード生成/ビルド・ファイル記述

Mobile Application Builder は、最終的なアプリケーション実行可能ファイルを作成するための、中間の C コードまたは Java コードを生成します。どちらを生成するかは、選択したターゲット・プラットフォームによって決まります。下記の表に、サポートされる各ターゲット・プラットフォームごとにどちらのコードが生成されるかについて、一般的な説明があります。

表 15. Mobile Application Builder コード生成

ターゲット・プラットフォーム	生成されるコード
Palm	C

表 15. Mobile Application Builder コード生成 (続き)

ターゲット・プラットフォーム	生成されるコード
Symbian	Java
WinCE	Java
Sharp Zaurus	Java
Generic Java	Java

アプリケーションをビルドしたとき、最終的な実行可能ファイルを作成するための、その他多数の中間ファイルが作成されます。これらの中間ファイルに対して手作業で変更を加えると、Mobile Application Builder を使用して再度アプリケーションをビルドした場合にそれらの変更が失われます。

ビルド・プリファレンスで「リリース」オプションを選択した場合、いくつかの中間ファイルがビルドの後に保管されません。下記の表に、サポートされる各ターゲット・プラットフォームごとに生成されるファイルについての説明があります。

Palm アプリケーションの場合、次のファイルが生成されます。

表 16. Palm C コード・アプリケーション・ビルド・ファイル

ファイル	説明	「リリース」モードで削除されるか?
*.prc	Palm アプリケーション実行可能ファイル。これは、プロジェクトが正常にビルドされたときに作成される最終出力です。	いいえ
*.c	アプリケーションのソース・コード	はい
sql*.h	アプリケーションで DB2 Everyplace エンジン機能へのアクセスが必要な場合のヘッダー・ファイル。	はい
*.h	アプリケーションを生成するためのヘッダー・ファイル。	はい
*.makefile	Palm 用の実行可能アプリケーションをビルドするために使用されます。	はい
*.rcp	アプリケーション内の Palm リソースの定義。	はい
MAB-sections.*	複数コード・セグメント・サポートに関連したファイル。	いいえ
*.bin	アプリケーション内の実行可能 Palm リソース。	はい (正常にビルドされた後は常に削除されます)

表 16. Palm C コード・アプリケーション・ビルド・ファイル (続き)

ファイル	説明	「リリース」モードで削除されるか?
*.s, *.ld, *.stamp, MAB*.o	これらの中間ファイルは、正常にビルドされた後、常に削除されます。ビルドに失敗した場合はこれらの中間ファイルが保管されますが、デバッグには役立ちません。	はい (正常にビルドされた後は常に削除されます)
MAB_Icon.bmp, *.def	これらのファイルが正常にビルドされた後に削除されるのは、ビルド・プリファレンスで「リリース」オプションを選択した場合のみです。これらのファイルを保管しておく、デバッグに役立つ可能性があります。	はい

次のファイルは、Java コードの生成が必要なすべてのターゲット・プラットフォームで生成されます。

表 17. Java アプリケーション・ビルド・ファイル (すべてのプラットフォーム)

ファイル	説明	「リリース」モードで削除されるか?
*.java	生成された Java ソース・ファイル。	はい
*.class	Java コンパイラーによって生成されたクラス・ファイル。	はい
*.jar	クラス・ファイルから生成されたアプリケーション jar。	いいえ
*.bat	これらのファイルは、ソースをコンパイルし、アプリケーションをパッケージするために、生成され、実行されます。	いいえ

Symbian アプリケーションの場合、上記の表 5 のファイルに加えて、次のファイルが生成されます。

表 18. Symbian アプリケーション・ビルド・ファイル

ファイル	記述	「リリース」モードで削除されるか?
*.sis	Symbian アプリケーション実行可能ファイル。	いいえ

表 18. Symbian アプリケーション・ビルド・ファイル (続き)

ファイル	記述	「リリース」モードで削除されるか?
*.bmp、*.mbm	これらは、アプリケーションのアイコンに関連したファイルです。*.bmp ファイルはユーザーによって提供され、*.mbm ファイルは Symbian ツール・チェーンによって生成されます。	いいえ
*.app、*.aifb、*.aif、*.txt、*.pkg	コード生成に関連した中間ファイル。デバッグには役立ちません。	いいえ

WinCE アプリケーションの場合、上記の表 5 のファイルに加えて、次のファイルが生成されます。

表 19. WinCE アプリケーション・ビルド・ファイル

*.cab	WinCE アプリケーション実行可能ファイル。	いいえ
*.inf、*.DAT、*.exe	コード生成に関連した中間ファイル。デバッグには役立ちません。	いいえ
ファイル	記述	「リリース」モードで削除されるか?

#### 関連タスク:

108 ページの『Mobile Application Builder アプリケーションのビルド』  
47 ページの『ビルド・プリファレンスの設定』

## Mobile Application Builder アプリケーションのテスト

エミュレーターまたは実際のモバイル・デバイスを使用してアプリケーションのテストを行います。

アプリケーションをエミュレーターまたはモバイル・デバイス上でテストする前に、そのアプリケーションをビルドしておく必要があります。

Java で生成されたアプリケーションを使用するには、デバイスまたはエミュレーターが Java 仮想マシン (JVM) をサポートしていなければなりません。JVM をサポートできる他のデバイス・プラットフォームもサポートされますが、最小 16MB の RAM を備えたデバイスの使用をお勧めします。新しいデバイスでは、JVM が組み込まれているものもあるでしょう。添付 CD に JVM をパッケージしているデバイスもあります。また、ご自身で、他の JVM を Web を通して購入したりダウンロードしたりできます。

---

#### 関連タスク:

108 ページの『Mobile Application Builder アプリケーションのビルド』  
15 ページの『MAB バージョン 8.1.4 のセットアップおよび構成』

#### 関連参照情報:

110 ページの『コード生成/ビルド・ファイル記述』

## 表の作成および移植

表の定義を、アプリケーションの作成に使用するために Mobile Application Builder にインポートしても、アプリケーションで使用する表は、アプリケーションを最初に実行する前に、エミュレーターまたはモバイルで作成または移植されている必要があります。アプリケーションが実行されるときに、アプリケーションが自身の表を移植する (たとえば、空の表で始動する) 場合でも、一度データベースに表を作成する必要があります。これを行う方法はいくつかあります。アプリケーションの使用法と、表にロードしたいデータの量によって、方法の選択が変わってきます。

以下のステップは、エミュレーターまたはモバイル・デバイス上でアプリケーションをテストする、より大きなタスクの一部です。表を作成して移植するステップを終了したら、「Mobile Application Builder アプリケーションのテスト」の中の該当するセクションに戻ってください。

---

#### Schemalnit を使用して WinCE アプリケーション用の表を作成する:

NurseInit.bat は、提供された SchemaInit.java プログラムを使用して、ASCII ファイルの行内容に基づくデータベース表を作成し移植するサンプル・スクリプトです。DB2 Everyplace がインストールされている場所へのパス情報を調整するように NurseInit.bat を変更してから、実行して、PersonList および VNApp サンプルがデスクトップで使用するスキーマを任意のディレクトリーに作成します。Schemalnit は、表がすでに存在していれば、それらをデータベースから除去し、ターゲット・ディレクトリー (Windows) が存在していなければ、それを作成します。

このスクリプトを使用してアプリケーション用の表を作成および移植するには、  
..%Projects%Samples%Win32 ディレクトリーの tables.txt および data.txt を変更して、アプリケーションでアクセスする表およびデータを反映させます。

#### CLP を使用して表を作成する:

アプリケーションのプロトタイピングまたはテストのために表データが少量だけ必要な場合は、デバイスまたはエミュレーター上で実行できる CLP 機能を使用することができます。CREATE TABLE ステートメントを入力して表を作成し、INSERT INTO を使用して表に行を追加します。SQL ステートメントについて詳しくは、「IBM DB2 Everyplace アプリケーション開発ガイド」を参照してください。CLP の使用について詳しくは、「IBM DB2 Everyplace インストールおよびユーザーズ・ガイド」を参照してください。

### CSV ファイル (コンマ区切り値ファイル) からインポートして表を作成する:

手作業では入力できない大量のデータを使用するアプリケーションでは、DB2 Everyplace のインポート・ツールを使用して、表を作成し、CSV ファイルからデータをロードすることができます。このファイルは、さまざまなデスクトップ・ツールを使用して作成されます。DB2 コントロール・センターに含まれる表に対して「DDL の生成...」メニュー機能を使用して、ご使用のワークステーション上のファイル・システムに表を作成するためのデータ定義言語 (DDL) ステートメントを生成することができます。CSV ファイルを作成して (表に対する「エクスポート」メニューから、「区切り ASCII 形式」を使用して作成する)、行データを含むファイルを生成することもできます。このファイルは Windows 上の PalmImport.bat プログラム、および、デバイスまたはエミュレーターの DB2eImport アプリケーションで、データのロードに使用されます。

### DB2e CLP から IMPORT コマンドを使用して表を作成する:

提供されたアプリケーションの DB2e CLP は、バージョン 7.2 の新機能です。これは CLP 機能の拡張バージョンです。DB2e CLP から IMPORT コマンドを発行することができます。Palm プラットフォームの場合、バージョン 7.2 以上で使用可能です。これは以前、他の Palm 以外のプラットフォームで使用可能でした。CREATE TABLE コマンドを入力後、IMPORT コマンドを発行して、Palm メモ・アプリケーションに入れたデータから表の行を移植することができます。メモの最初の一行は Palm 名で、この名前と同じ名前が IMPORT コマンドに対して指定されます。メモの残りの部分には、コンマで区切られた (CSV 形式) 行の値が含まれている必要があります。内容は、デバイス、エミュレーター、および Palm デスクトップ・ソフトウェア (HotSync) に作成されます。「**IBM DB2 Everyplace インストールおよびユーザズ・ガイド**」の第 6 章『CLP アプリケーション』を参照してください。

### アプリケーションを作成して表を作成する:

ユーザー・アプリケーションに必要な表を作成し、移植する機能を持ったアプリケーションを作成することもできます。メイン・アプリケーションを使用する前に、この初期化アプリケーションを一度実行してください。提供されたサンプル・プログラム NurseInit は、この方法で機能します。NurseInit は DB2 CLI 関数を使用し、CodeWarrior コード開発ツールを使用して作成されたものです。MAB 作成のアプリケーション内で使用するために CLI コード (スクリプト) を書くことはできませんが、現在の MAB の設計でこれを行うのは容易ではありません。CLI プログラミングについて詳しくは、「**IBM DB2 Everyplace アプリケーション開発ガイド**」を参照してください。C ソース・コードおよび関連ファイルは、DB2 Everyplace データベース・コンポーネントをインストールした Samples ディレクトリーにインストールされています。

### 同期サーバーを使用して、バックエンド・データベースと同期することによって表を作成する:

IBM DB2 Everyplace 同期サーバーのコンポーネントがインストールされている場合、同期サブスクリプションを定義した後は、デバイスまたはエミュレーター上での IBM Sync アプリケーションの「最新表示」設定を使用して、バックエンド・データベースにすでに存在している表からデバイスまたはエミュレーター上で必要な

表を作成し、初期化 (移植) することができます。詳しくは、「*IBM DB2 Everyplace 同期サーバー 管理ガイド*」を参照してください。

『エミュレーター上での Palm アプリケーションのテスト』に戻ります。

118 ページの『モバイル・デバイス上での Palm アプリケーションのテスト』に戻ります。

119 ページの『エミュレーター上での Symbian Crystal アプリケーションのテスト』に戻ります。

120 ページの『モバイル・デバイス上での Symbian Crystal アプリケーションのテスト』に戻ります。

123 ページの『デスクトップ上での WinCE アプリケーションのシミュレート』に戻ります。

124 ページの『モバイル・デバイス上での WinCE アプリケーションのテスト』に戻ります。

---

## Palm アプリケーションのテスト

Palm アプリケーションはエミュレーターまたはデバイスでテストできます。

---

### Mobile Application Builder Palm アプリケーション・テスト・タスク:

- 『エミュレーター上での Palm アプリケーションのテスト』
- 118 ページの『モバイル・デバイス上での Palm アプリケーションのテスト』

---

### エミュレーター上での Palm アプリケーションのテスト

エミュレーターを使用してアプリケーションをテストすることができます。ほとんどのモバイル・デバイスにはエミュレーターまたはシミュレーターがあり、開発マシンのモバイル環境でアプリケーションのテストを行うことができます。エミュレーターを使用すると、実際のモバイル・デバイスがなくてもアプリケーションをアップロードすることができます。

Palm エミュレーター・テストの前提条件:

- ご使用のワークステーションに Palm エミュレーターを正しくインストールし、構成しておく必要があります。詳しくは、『Palm セットアップおよび構成』を参照してください。
- Mobile Application Builder にアプリケーションを正常にビルドしておく必要があります。

---

エミュレーター上で Palm アプリケーションをテストするには:

1. Mobile Application Builder のメニュー・バーから「ビルド」->  「テスト」をクリックするか、あるいは、ツールバーから  をクリックします。インストールされているアプリケーションとともに、エミュレーター・セッションが起動します。
2. 使用するアプリケーションに必要な表を作成し、移植します。
3. アプリケーション・アイコンをタップし、アプリケーションを立ち上げてください。

---

#### 関連タスク:

- 108 ページの『Mobile Application Builder アプリケーションのビルド』
- 118 ページの『モバイル・デバイス上での Palm アプリケーションのテスト』

#### 関連参照情報:

- 110 ページの『コード生成/ビルド・ファイル記述』

### Palm OS 5 Simulator 上での Palm アプリケーションのテスト

Palm OS 5 Simulator を使用して Palm OS 5.x アプリケーションをテストできます。シミュレーターを使用すると、実際のモバイル・デバイスがなくてもアプリケーションをアップロードすることが可能です。

Palm OS 5 Simulator テストの前提条件:

- ご使用のワークステーションに Palm OS 5 Simulator を正しくインストールし、構成しておく必要があります。詳しくは、『Palm セットアップおよび構成』を参照してください。
- Mobile Application Builder にアプリケーションを正常にビルドしておく必要があります。

---

#### Palm OS 5 Simulator 上で Palm OS 5.x アプリケーションをテストするには:

1. PalmSim.exe をダブルクリックしてシミュレーターを開始します。
2. シミュレーター・セットアップ・ステップの完了後に保管したシミュレーター・セッションをオープンします。
  - a. シミュレーター画面の任意の場所を右マウス・ボタン・クリックし、「ストレージ」→「ロード」をクリックします。
  - b. 適切なシミュレーター・セッションまでブラウズして、「オープン」をクリックします。
3. シミュレーター画面の任意の場所を右マウス・ボタン・クリックし、ポップアップ・メニューの「インストール」→「データベース」をクリックします。
4. アプリケーションまでブラウズして、「オープン」をクリックします。アプリケーションがシミュレーターにインストールされます。
5. 使用するアプリケーションに必要な表を作成し、移植します。

6. アプリケーション・アイコンをタップし、アプリケーションを立ち上げてください。

---

**関連タスク:**

108 ページの『Mobile Application Builder アプリケーションのビルド』  
『モバイル・デバイス上での Palm アプリケーションのテスト』

**関連参照情報:**

110 ページの『コード生成/ビルド・ファイル記述』

**モバイル・デバイス上での Palm アプリケーションのテスト**

エミュレーターまたは実際のモバイル・デバイスを使用してアプリケーションのテストを行います。エミュレーター上でのテストが機能する場合でも、実際のモバイル・デバイス上でアプリケーションの機能を確認するようにしてください。

Palm モバイル・デバイス・テストの前提条件:

- Palm モバイル・デバイスをテスト用に正しく構成しておく必要があります。詳しくは、『Mobile Application Builder セットアップおよび構成』を参照してください。
- Mobile Application Builder にアプリケーションを正常にビルドしておく必要があります。

---

**モバイル・デバイス上で Palm アプリケーションをテストするには:**

1. Palm デスクトップ・ソフトウェアのインストール機能を使用して、デバイスにアプリケーションをインストールします。
2. 使用するアプリケーションに必要な表を作成し、移植します。
3. 次の Palm HotSync でアプリケーションを追加します。
4. アプリケーション・アイコンをタップし、アプリケーションを立ち上げてください。

---

**関連タスク:**

108 ページの『Mobile Application Builder アプリケーションのビルド』  
116 ページの『エミュレーター上での Palm アプリケーションのテスト』

**関連参照情報:**

110 ページの『コード生成/ビルド・ファイル記述』

## Symbian Crystal アプリケーションのテスト

Symbian Crystal アプリケーションはエミュレーターまたはデバイスでテストできません。

---

### Mobile Application Builder Symbian Crystal アプリケーション・テスト・タスク:

- 『エミュレーター上での Symbian Crystal アプリケーションのテスト』
- 120 ページの『モバイル・デバイス上での Symbian Crystal アプリケーションのテスト』

---

### エミュレーター上での Symbian Crystal アプリケーションのテスト

エミュレーターを使用してアプリケーションをテストすることができます。ほとんどのモバイル・デバイスにはエミュレーターまたはシミュレーターがあり、開発マシンのモバイル環境でアプリケーションのテストを行うことができます。エミュレーターを使用すると、実際のモバイル・デバイスがなくてもアプリケーションをアップロードすることができます。

Mobile Application Builder を使用して作成されたアプリケーションは、エミュレーター上のデフォルト・ロケーションでデータベース表を探します。Symbian Crystal の場合、これは `\system\data\isync\` です。アプリケーションを使用する前に、必要なデータベース表がこのロケーションに存在していることを確認してください。

Mobile Application Builder から Symbian Crystal アプリケーションをテストすると、Mobile Application Builder はビルド・アプリケーション・ファイルをデスクトップの

```
x:\Symbian\6.0\NokiaJava\Epoc32\Wins\c\System\Apps\
```

ディレクトリーにインストールされます。インストールするその他の .SIS ファイルをセットアップするには、そのファイルを

```
x:\Symbian\6.0\NokiaJava\erj
```

ディレクトリーにコピーします。

Symbian Crystal エミュレーター・テストの前提条件:

- ご使用のワークステーションに Symbian Crystal エミュレーターを正しくインストールし、構成しておく必要があります。詳しくは、『Mobile Application Builder セットアップおよび構成』を参照してください。
- Mobile Application Builder にアプリケーションを正常にビルドしておく必要があります。

---

エミュレーター上で Symbian Crystal アプリケーションをテストするには:

1. Mobile Application Builder のメニュー・バーから「ビルド」->  「テスト」をクリックするか、あるいは、ツールバーから  をクリックします。インストールされているアプリケーションと共に、エミュレーター・セッションが立ち上げられます。
2. 使用するアプリケーションに必要な表を作成し、移植します。
3. エミュレーターのアプリケーション・アイコンをクリックして、アプリケーションを起動します。

---

#### 関連タスク:

108 ページの『Mobile Application Builder アプリケーションのビルド』 『モバイル・デバイス上での Symbian Crystal アプリケーションのテスト』

#### 関連参照情報:

110 ページの『コード生成/ビルド・ファイル記述』

## モバイル・デバイス上での Symbian Crystal アプリケーションのテスト

Mobile Application Builder は、デバイス・インストール可能な .SIS ファイル (*ApplicationIconName.SIS*) をプロジェクト・ディレクトリーに生成します。

Mobile Application Builder を使用して作成されたアプリケーションは、デバイス上のデフォルト・ロケーションでデータベース表を探します。Symbian Crystal の場合、これは `\system\data\isync\` です。アプリケーションを使用する前に、必要なデータベース表がこのロケーションに存在していることを確認してください。

Symbian Crystal モバイル・デバイス・テストの前提条件:

- Symbian Crystal デバイスを正しくインストールし、構成しておく必要があります。詳しくは、『Mobile Application Builder セットアップおよび構成』を参照してください。
- Mobile Application Builder にアプリケーションを正常にビルドしておく必要があります。

---

モバイル・デバイス上で Symbian Crystal アプリケーションをテストするには:

1. *ApplicationIconName.SIS* ファイルをモバイル・デバイスにインストールします。アプリケーションが、モバイル・デバイス上の **Extras** バーに表示されません。
  2. 使用するアプリケーションに必要な表を作成し、移植します。
  3. アプリケーション・アイコンを選択し、それを始動してアプリケーションを立ち上げます。
-

#### 関連タスク:

108 ページの『Mobile Application Builder アプリケーションのビルド』

119 ページの『エミュレーター上での Symbian Crystal アプリケーションのテスト』

#### 関連参照情報:

110 ページの『コード生成/ビルド・ファイル記述』

## Symbian UIQ アプリケーションのテスト

Symbian UIQ アプリケーションはエミュレーターまたはデバイスでテストできます。

---

#### Mobile Application Builder Symbian UIQ アプリケーション・テスト・タスク:

- 『エミュレーター上での Symbian UIQ アプリケーションのテスト』
- 122 ページの『モバイル・デバイス上での Symbian UIQ アプリケーションのテスト』

---

### エミュレーター上での Symbian UIQ アプリケーションのテスト

エミュレーターを使用してアプリケーションをテストすることができます。ほとんどのモバイル・デバイスにはエミュレーターまたはシミュレーターがあり、開発マシンのモバイル環境でアプリケーションのテストを行うことができます。エミュレーターを使用すると、実際のモバイル・デバイスがなくてもアプリケーションをアップロードすることができます。

Mobile Application Builder を使用して作成されたアプリケーションは、エミュレーター上のデフォルト・ロケーションでデータベース表を探します。Symbian UIQ の場合、これは `\system\data\isync\` です。アプリケーションを使用する前に、必要なデータベース表がこのロケーションに存在していることを確認してください。

Mobile Application Builder から Symbian UIQ アプリケーションをテストすると、Mobile Application Builder はビルド・アプリケーション・ファイルをデスクトップの

```
x:\Symbian\UIQ_70\epoc32\winscw\c\system\Apps\<application name>
```

ディレクトリーに入れます。インストールするその他の .SIS ファイルをセットアップするには、そのファイルを

```
x:\Symbian\UIQ_70\erj
```

ディレクトリーにコピーします。

Symbian UIQ エミュレーター・テストの前提条件:

- ご使用のワークステーションに Symbian UIQ エミュレーターを正しくインストールし、構成しておく必要があります。詳しくは、『Mobile Application Builder セットアップおよび構成』を参照してください。
- Mobile Application Builder にアプリケーションを正常にビルドしておく必要があります。

---

#### エミュレーター上で Symbian UIQ アプリケーションをテストするには:

1. Mobile Application Builder のメニュー・バーから「ビルド」->  「テスト」  
をクリックするか、あるいは、ツールバーから  をクリックします。インストールされているアプリケーションと共に、エミュレーター・セッションが立ち上げられます。
2. 使用するアプリケーションに必要な表を作成し、移植します。
3. エミュレーターのアプリケーション・アイコンをクリックして、アプリケーションを起動します。

---

#### 関連タスク:

108 ページの『Mobile Application Builder アプリケーションのビルド』  
『モバイル・デバイス上での Symbian UIQ アプリケーションのテスト』

#### 関連参照情報:

110 ページの『コード生成/ビルド・ファイル記述』

### モバイル・デバイス上での Symbian UIQ アプリケーションのテスト

Mobile Application Builder は、デバイス・インストール可能な .SIS ファイル (*ApplicationIconName.SIS*) をプロジェクト・ディレクトリーに生成します。

Mobile Application Builder を使用して作成されたアプリケーションは、デバイス上のデフォルト・ロケーションでデータベース表を探します。Symbian UIQ の場合、これは `\system\data\isync\` です。アプリケーションを使用する前に、必要なデータベース表がこのロケーションに存在していることを確認してください。

#### Symbian UIQ モバイル・デバイス・テストの前提条件:

- Symbian UIQ デバイスを正しくインストールし、構成しておく必要があります。詳しくは、『Mobile Application Builder セットアップおよび構成』を参照してください。
- Mobile Application Builder にアプリケーションを正常にビルドしておく必要があります。

---

#### モバイル・デバイス上で Symbian UIQ アプリケーションをテストするには:

1. `ApplicationIconName.SIS` ファイルをモバイル・デバイスにインストールします。
2. 使用するアプリケーションに必要な表を作成し、移植します。
3. アプリケーション・アイコンを選択し、それを始動してアプリケーションを立ち上げます。

---

#### 関連タスク:

108 ページの『Mobile Application Builder アプリケーションのビルド』

121 ページの『エミュレーター上での Symbian UIQ アプリケーションのテスト』

#### 関連参照情報:

110 ページの『コード生成/ビルド・ファイル記述』

## WinCE アプリケーションのテスト

WinCE アプリケーションは、デバイス上、またはデスクトップのシミュレート・アプリケーションでテストできます。

---

#### Mobile Application Builder WinCE アプリケーション・テスト・タスク:

- 『デスクトップ上での WinCE アプリケーションのシミュレート』
- 124 ページの『モバイル・デバイス上での WinCE アプリケーションのテスト』

---

### デスクトップ上での WinCE アプリケーションのシミュレート

Win32 エミュレーション機能を使用して、デスクトップ上で WinCE アプリケーションを実行することができます。これは真のエミュレーションではありませんが、デバイス上でアプリケーションがどのように見えるかを知ることができます。

Mobile Application Builder を使用して作成されたアプリケーションは、デバイスまたはエミュレーター上のデフォルト・ロケーションでデータベース表を探します。WinCE/PocketPC の場合、これは ¥ です。アプリケーションを使用する前に、必要なデータベース表がこのロケーションに存在していることを確認してください。

#### WinCE デスクトップ・シミュレーションの前提条件:

- ご使用のワークステーションに Win32 エミュレーターを正しくインストールし、構成しておく必要があります。詳しくは、『Mobile Application Builder セットアップおよび構成』を参照してください。
- Mobile Application Builder にアプリケーションを正常にビルドしておく必要があります。

デスクトップ上で **WinCE** アプリケーションをシミュレートするには:

1. Mobile Application Builder のメニュー・バーから「ビルド」->  「テスト」をクリックするか、あるいは、ツールバーから  をクリックします。インストールされているアプリケーションと共に、Win32 エミュレーター・セッションが立ち上げられます。
2. 使用するアプリケーションに必要な表を作成し、移植します。

---

#### 関連タスク:

108 ページの『Mobile Application Builder アプリケーションのビルド』  
『モバイル・デバイス上での WinCE アプリケーションのテスト』

#### 関連参照情報:

110 ページの『コード生成/ビルド・ファイル記述』

### モバイル・デバイス上での WinCE アプリケーションのテスト

Mobile Application Builder を使用して作成されたアプリケーションは、デバイス上のデフォルト・ロケーションでデータベース表を探します。WinCE¥PocketPC の場合、これは ¥ です。アプリケーションを使用する前に、必要なデータベース表がこのロケーションに存在していることを確認してください。

WinCE モバイル・デバイス上でのアプリケーション・テストの前提条件

- Win32 モバイル・デバイスを正しく構成しておく必要があります。詳しくは、『Mobile Application Builder セットアップおよび構成』を参照してください。
- Mobile Application Builder にアプリケーションを正常にビルドしておく必要があります。

---

モバイル・デバイス上で **WinCE** アプリケーションをテストするには:

Mobile Application Builder は、JVM を持つ任意のデバイスにインストール可能なアプリケーションを生成します。アプリケーションをビルドする前に、JVM に合わせて WinCE ビルド・プリファレンスを設定する必要があります。

1. `projectname.cab` をデバイス上の任意のディレクトリーにコピーします。デバイス上のファイルをタップしてアプリケーションをインストールします。
2. 使用するアプリケーションに必要な表を作成し、移植します。
3. 「スタート」メニューのアプリケーション・アイコンをタップしてアプリケーションを起動します。

---

#### 関連タスク:

108 ページの『Mobile Application Builder アプリケーションのビルド』  
123 ページの『デスクトップ上での WinCE アプリケーションのシミュレート』

**関連参照情報:**

110 ページの『コード生成/ビルド・ファイル記述』

## Sharp Zaurus アプリケーションのテスト

Sharp Zaurus アプリケーションは、デバイス上、またはデスクトップのシミュレート・アプリケーションでテストできます。

---

**Mobile Application Builder Sharp Zaurus アプリケーション・テスト・タスク:**

- 『デスクトップ上での Sharp Zaurus アプリケーションのシミュレート』
- 126 ページの『モバイル・デバイス上での Zaurus アプリケーションのテスト』

---

### デスクトップ上での Sharp Zaurus アプリケーションのシミュレート

Win32 エミュレーション機能を使用して、デスクトップ上で Sharp Zaurus アプリケーションを実行できます。これは真のエミュレーションではありませんが、デバイス上でアプリケーションがどのように見えるかを知ることができます。

**Zaurus デスクトップ・シミュレーションの前提条件:**

- ご使用のワークステーションに Win32 エミュレーターを正しくインストールし、構成しておく必要があります。詳しくは、『Mobile Application Builder セットアップおよび構成』を参照してください。
- Mobile Application Builder にアプリケーションを正常にビルドしておく必要があります。

---

**デスクトップ上で Zaurus アプリケーションをシミュレートするには:**

1. Mobile Application Builder のメニュー・バーから「ビルド」->  「テスト」をクリックするか、あるいは、ツールバーから  をクリックします。インストールされているアプリケーションと共に、Win32 エミュレーター・セッションが立ち上げられます。
2. 使用するアプリケーションに必要な表を作成し、移植します。

---

**関連タスク:**

108 ページの『Mobile Application Builder アプリケーションのビルド』  
『モバイル・デバイス上での Zaurus アプリケーションのテスト』

**関連参照情報:**

110 ページの『コード生成/ビルド・ファイル記述』

**モバイル・デバイス上での Zaurus アプリケーションのテスト**

Mobile Application Builder は、デバイス・インストール可能な .IPK ファイルをプロジェクト・ディレクトリーに生成します。

Zaurus モバイル・デバイス・テストの前提条件:

- Sharp Zaurus モバイル・デバイスを正しく構成しておく必要があります。詳しくは、『Mobile Application Builder セットアップおよび構成』を参照してください。
- Mobile Application Builder にアプリケーションを正常にビルドしておく必要があります。

---

モバイル・デバイス上で **Zaurus** アプリケーションをテストするには:

1. .IPK ファイルをモバイル・デバイスにインストールします。アプリケーションは、モバイル・デバイスの「**文書 (Documents)**」の場所に、Jeode の下のショートカットで表示されます。
2. 使用するアプリケーションに必要な表を作成し、移植します。
3. アプリケーション・アイコンをタップしてアプリケーションを起動します。

---

**関連タスク:**

108 ページの『Mobile Application Builder アプリケーションのビルド』

125 ページの『デスクトップ上での Sharp Zaurus アプリケーションのシミュレート』

**関連参照情報:**

110 ページの『コード生成/ビルド・ファイル記述』

---

## 第 2 章 概念/参照

本節では、Mobile Application Builder の概念および参照情報を提供しています。

---

### パレット制御アイコン

制御パレットは、アプリケーションを組み立てるのに使用される制御が含まれます。Mobile Application Builder には、制御の標準パレットが付いています。ユーザーのアプリケーションのために使用可能な制御は、ターゲット・プラットフォームによって異なります。

表 20. パレット制御アイコン：記述

アイコン	記述
	フォーム・ビットマップ
	ボタン
	チェック・ボックス
	フィールド
	Graffiti シフト
	インク
	blob
	ラベル
	リスト
	ポップアップ・トリガー (Palm プラットフォーム)、選択リスト (Java プラットフォーム)
	プッシュボタン
	繰り返しボタン
	スクロール・バー
	選択トリガー

**関連タスク:**

60 ページの『制御』

**関連参照情報:**

『プロパティの制御』

---

## プロジェクト・プロパティ

MAB プロジェクトのプロパティを設定します。プロジェクト・プロパティは、ターゲット・プラットフォームによって異なります。

表 21. プロジェクト・プロパティ

プロパティ	説明
名前	ターゲット・プラットフォームを識別するために使用される名前です。このプロパティを変更することはできません。
始動フォーム	アプリケーションの実行時に最初に表示されるフォームです。
アプリケーション ID	このアプリケーションの固有 ID です。アプリケーション ID の詳細については、50 ページの『アプリケーション ID』を参照してください。
カテゴリ	Palm OS エミュレーターまたはデバイスでの、このアプリケーションのアプリケーション・カテゴリです。このカテゴリがまだ存在しない場合は、Palm アプリケーション・ランチャーに作成されます。
データベース・ロケーション	このアプリケーションのデータベース表のロケーションです。

**関連タスク:**

38 ページの『MAB プロジェクト』

50 ページの『アプリケーション ID』

---

## プロパティの制御

制御パレットは、アプリケーションを組み立てるのに使用される制御が含まれます。それぞれの制御ごとに、ご使用のアプリケーションの要求に合うようカスタマイズできる属性がたくさんあります。

使用可能な制御は、ターゲット・プラットフォームによって異なります。

---

### Mobile Application Builder 制御プロパティ:

『フォーム・ビットマップ・プロパティ』  
 131 ページの『ボタン・プロパティ』  
 133 ページの『チェック・ボックス・プロパティ』  
 134 ページの『選択リスト・プロパティ』  
 135 ページの『コマンド・ボタン配列 (CBA) プロパティ』  
 136 ページの『フィールド・プロパティ』  
 138 ページの『Graffiti シフト・プロパティ』  
 139 ページの『Ink プロパティ』  
 140 ページの『ラベル・プロパティ』  
 140 ページの『リスト・プロパティ』  
 142 ページの『ポップアップ・トリガー・プロパティ』  
 143 ページの『プッシュボタン・プロパティ』  
 144 ページの『繰り返しボタン・プロパティ』  
 146 ページの『スクロール・バー・プロパティ』  
 147 ページの『選択トリガー・プロパティ』

#### 関連タスク:

60 ページの『制御』

#### 関連参照情報:

127 ページの『パレット制御アイコン』

## フォーム・ビットマップ・プロパティ

フォーム上でビットマップ・イメージを表示します。白黒または直接色 (Palm OS 4.0 またはそれ以上でサポートされます) のビットマップ形式でイメージを保管できる任意の外部ドロー・プログラムを使用して、ビットマップ・イメージを作成することができます。ビットマップは、アプリケーションのプロジェクト・ディレクトリーに保管します。

この制御のサポートは、ターゲット・プラットフォームによって異なります。

表 22. フォーム・ビットマップ・プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
表題	「Mobile Application Builder フォーム」ペインのビットマップに表示されるテキストです。この表題テキストは、ユーザーには表示されません。
左そろえ	制御の左端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
トップ	制御の上端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。

表 22. フォーム・ビットマップ・プロパティ (続き)

プロパティ	説明
ビットマップ・リソース名	ビットマップ・ファイルの名前です。ここでこの名前を選択する前に、プロジェクト・ツリー・ビューでビットマップ・リソースを定義する必要があります。
幅	制御の幅 (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。
高さ	制御の高さ (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

#### 関連タスク:

60 ページの『制御』

75 ページの『ビットマップの作成および表示』

#### 関連参照情報:

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

## blob プロパティ

この制御を使用して、イメージを BLOB データ・タイプとして保管します。この制御を使用するには、Java 仮想マシンのデバイスで blob がサポートされていなければなりません。これは Java 1.2 の機能です。カスタムの Java 仮想マシンを使用する場合、SQL パッケージに blob サポートが含まれていることを確認する必要があります。

この制御のサポートは、ターゲット・プラットフォームによって異なります。

表 23. blob プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
データ・ソース	制御が読み取り、書き込みを行うデータ・フィールドや列を含む表の名前です。使用可能な「データ・ソース」のリストは、そのプロジェクトにインポートされた表に基づいています。詳しくは、52 ページの『表定義ファイルの作成およびインポート』を参照してください。
データ・フィールド	制御が読み取り、書き込みを行う列の名前です。使用可能な「データ・フィールド」のリストは、「データ・ソース」プロパティで選択された表に基づいています。
使用可能	使用可能になっている制御は、ユーザーに表示されます。「使用可能」はこのプロパティのデフォルト設定です。使用不可能になっている制御はフォーム上にドロウできませんが実行中はドロウするようにプログラムできます。

表 23. blob プロパティ (続き)

プロパティ	説明
左方	制御の左端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
トップ	制御の上端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
幅	制御の幅 (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。
高さ	制御の高さ (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

#### 関連タスク:

60 ページの『制御』

#### 関連参照情報:

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

## ボタン・プロパティ

イベントを実行します。イベントはアクションを実行します。アクションにはターゲットがあります。たとえば、ボタン・タップ・イベントは、ターゲット・スクリプトを実行する RunScript アクションを実行します。

この制御のサポートは、ターゲット・プラットフォームによって異なります。

表 24. ボタン・プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
表題	ボタンに表示されるテキストです。
フォント	その制御の表題で使用されるフォントです。
ビットマップ	選択されていない状態で、制御上に表示されるビットマップ・イメージです。ここでビットマップ・イメージを指定できるようにするためには、事前にそのビットマップ・イメージをプロジェクトに追加しておく必要があります。ビットマップを制御上に表示するためには、「ビットマップ」と「選択されたビットマップ」の両方のプロパティ用にイメージを指定しておかなければなりません。これらのイメージは、同じでも構いません。ビットマップのサポートは、ターゲット・プラットフォームによって異なります。

表 24. ボタン・プロパティ (続き)

プロパティ	説明
選択されたビットマップ	選択された状態で、制御上に表示されるビットマップ・イメージです。ここでビットマップ・イメージを指定できるようにするためには、事前にそのビットマップ・イメージをプロジェクトに追加しておく必要があります。ビットマップを制御上に表示するためには、「ビットマップ」と「選択されたビットマップ」の両方のプロパティ用にイメージを指定しておかなければなりません。これらのイメージは、同じでも構いません。ビットマップのサポートは、ターゲット・プラットフォームによって異なります。
使用可能	使用可能になっている制御は、ユーザーに表示されます。「使用可能」はこのプロパティのデフォルト設定です。使用不可になっている制御はフォーム上にドロウできませんが実行中はドロウするようにプログラムできます。
バックグラウンド	この制御の背景の色です。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
フォアグラウンド	この制御の前景の色です。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
左方アンカー	左方アンカーが選択されている場合、制御の左方結合が固定されます。左方アンカーはこのプロパティのデフォルト設定です。右方アンカーが選択されている場合、制御の右方結合が固定されます。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
フレーム	フレームが選択されている場合、制御の周りに角が丸い長方形のフレームが描かれます。フレームなしが選択されている場合、制御の周りに境界線は描かれませんが、太いフレームが選択されている場合、制御の周りに角が丸く太い長方形のフレームが描かれます。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
左方	制御の左端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
トップ	制御の上端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
幅	制御の幅 (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。
高さ	制御の高さ (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

**関連タスク:**

60 ページの『制御』

**関連参照情報:**

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

## チェック・ボックス・プロパティ

チェックの有無によって 2 通りの状態を持つブール制御です。状態を検査するスクリプトを実行するためのタップ・イベントを実行することができます。「データ・ソース」プロパティおよび「データ・フィールド」プロパティによって、初期状態を設定することができます。チェックするか、あるいはチェックをクリアすることで、「データ・ソース」および「データ・フィールド」の更新イベントを実行することができます。

表 25. チェック・ボックス・プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
表題	制御の右方に表示されるテキストです。
データ・ソース	制御が読み取り、書き込みを行うデータ・フィールドや列を含む表の名前です。使用可能な「データ・ソース」のリストは、そのプロジェクトにインポートされた表に基づいています。詳しくは、52 ページの『表定義ファイルの作成およびインポート』を参照してください。
データ・フィールド	制御が読み取り、書き込みを行う列の名前です。使用可能な「データ・フィールド」のリストは、「データ・ソース」プロパティで選択された表に基づいています。
フォント	その制御の表題で使用されるフォントです。
使用可能	使用可能になっている制御は、ユーザーに表示されます。「使用可能」はこのプロパティのデフォルト設定です。使用不可になっている制御はフォーム上にドローできませんが実行中はドローするようにプログラムできます。
左方アンカー	左方アンカーが選択されている場合、制御の左方結合が固定されます。左方アンカーはこのプロパティのデフォルト設定です。右方アンカーが選択されている場合、制御の右方結合が固定されます。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
グループ ID	表の同じ列にマップされるチェック・ボックスごとの固有の番号です。Mobile Application Builder は、同じ列にリンクするチェック・ボックスごとに固有の値を自動的に割り当てます。非表示制御はアプリケーションの一部として見なされず、ドローできません。
チェック済み	真の場合、チェック・ボックスの初期選択状態はチェック済みになっています。偽の場合、チェック・ボックスの初期選択状態は未チェックになっています。
バックグラウンド	この制御の背景の色です。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
フォアグラウンド	この制御の前景の色です。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。

表 25. チェック・ボックス・プロパティ (続き)

プロパティ	説明
左方	制御の左端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
トップ	制御の上端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
幅	制御の幅 (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。
高さ	制御の高さ (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

#### 関連タスク:

60 ページの『制御』

#### 関連参照情報:

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

## 選択リスト・プロパティ

制御をタップした際に、ユーザーがドロップダウン・リストの多肢選択から選択できるようにします。選択リスト制御がクリックされると、アクションの命令が開始されます。次にリストが表示され、ユーザーは選択を行います。選択リストのラベルが変わり選択したものが表示される前に、制御に関連するすべてのイベントが実行されます。

この制御は、Java プラットフォームでのみ使用可能です。

表 26. 選択リスト・プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
使用可能	使用可能になっている制御は、ユーザーに表示されます。「使用可能」はこのプロパティのデフォルト設定です。使用不可になっている制御はフォーム上にドロウできませんが実行中はドロウするようにプログラムできます。
データ・ソース	制御が読み取り、書き込みを行うデータ・フィールドや列を含む表の名前です。使用可能な「データ・ソース」のリストは、そのプロジェクトにインポートされた表に基づいています。詳細については、『表定義ファイルの作成およびインポート』を参照してください。
データ・フィールド	制御が読み取り、書き込みを行う列の名前です。使用可能な「データ・フィールド」のリストは、「データ・ソース」プロパティで選択された表に基づいています。

表 26. 選択リスト・プロパティ (続き)

プロパティ	説明
選択基準	「 <b>選択基準</b> 」フィールドの右にある楕円をクリックして、データベース表の列を、既に定義してあるホスト変数に関連付けるためのウィンドウを開きます。詳しくは、『ホスト変数の使用』を参照してください。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
ホスト変数の更新	「ホスト変数の更新」の右側にある楕円をクリックし、ウィンドウを開きます。このウィンドウで、前に定義したどのホスト変数を、選択リストの選択結果で更新するかを定義することができます。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
左方アンカー	左方アンカーが選択されている場合、制御の左方結合が固定されます。左方アンカーはこのプロパティのデフォルト設定です。右方アンカーが選択されている場合、制御の右方結合が固定されます。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
バックグラウンド	この制御の背景の色です。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
フォアグラウンド	この制御の前景の色です。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
左方	制御の左端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
トップ	制御の上端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
幅	制御の幅 (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。
高さ	制御の高さ (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

#### 関連タスク:

60 ページの『制御』

#### 関連参照情報:

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

## コマンド・ボタン配列 (CBA) プロパティ

CBA とは、Nokia コミュニケーター上の 4 つのハードウェア・ボタンから成るセットです。これらのボタンは通常、現行表示に応じて、最も頻度の高いユーザー対話オプションに割り当てられます。CBA は、連絡機構上のすべてのアプリケーションで使用され、このデバイスの基本機能です。

Mobile Application Builder は 4 つの CBA 制御を、Symbian Crystal ターゲット・プラットフォーム用に作成された新規フォームに追加します。これらの制御は、デバイス上の 4 つのハードウェア・ボタンに対応しています。CBA 制御のためにプロパティおよびイベントを構成することができますが、フォーム上の別の場所に移動したり、フォームから除去したり、あるいはサイズを変更したりすることはできません。

表 27. コマンド・ボタン配列 (CBA) プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
表題	制御用に表示されるテキストです。
使用可能	使用可能になっている制御は、ユーザーに表示されます。「使用可能」はこのプロパティのデフォルト設定です。使用不可になっている制御はフォーム上にドロウできませんが実行中はドロウするようにプログラムできます。
左方	制御の左端から画面の端までの距離です。編集はできません。
トップ	制御の上端から画面の端までの距離です。編集はできません。
幅	制御の幅 (ピクセル表示) です。編集はできません。
高さ	制御の高さ (ピクセル表示) です。編集はできません。

#### 関連タスク:

60 ページの『制御』

#### 関連参照情報:

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

## フィールド・プロパティ

データの入力を表示または許可します。「データ・ソース」プロパティおよび「データ・フィールド」プロパティを使用して、初期状態を設定することができます。フィールドの変更で、「データ・ソース」および「データ・フィールド」の更新イベントを実行することができます。また、ストリングを使用して更新される場合もあります。

表 28. フィールド・プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。

表 28. フィールド・プロパティ (続き)

プロパティ	説明
データ・ソース	制御が読み取り、書き込みを行うデータ・フィールドや列を含む表の名前です。使用可能な「データ・ソース」のリストは、そのプロジェクトにインポートされた表に基づいています。詳細については、『表定義ファイルの作成およびインポート』を参照してください。
データ・フィールド	制御が読み取り、書き込みを行う列の名前です。使用可能な「データ・フィールド」のリストは、「データ・ソース」プロパティで選択された表に基づいています。
データ・フィールド・タイプ	「データ・フィールド」プロパティ用に選択した列の、データ・タイプを表示します。編集はできません。
固有値の使用	<p>「固有値の使用」フィールドの右の</p>  <p>をクリックして、フィールドに固有値を割り当てるかどうかを指定するウィンドウを開きます。また、固有値に組み込むパラメーターも指定できます。これは、データ・フィールド・タイプが VARCHAR または CHAR の場合にのみ使用可能です。また、この制御の「最大文字数」プロパティが 26 以上の場合にのみ使用可能です。</p>
バックグラウンド	この制御の背景の色です。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
フォアグラウンド	この制御の前景の色です。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
リンク・ソース	リンクする表の名前です。
リンク・フィールド	リンクするキーを含む列の名前です。
ホスト変数の更新	フィールドが変更されたときに、前に定義したどのホスト変数を更新するかを選択します。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
フォント	その制御の表題で使用されるフォントです。
使用可能	使用可能になっている制御は、ユーザーに表示されます。「使用可能」はこのプロパティのデフォルト設定です。使用不可になっている制御はフォーム上にドロウできませんが実行中はドロウするようにプログラムできます。
編集可能	編集可能が選択されている場合、フィールドの内容への変更が許可されます。編集不可能が選択されている場合、フィールドの内容への変更は許可されません。
単一行	単一行が選択されている場合、テキスト行の一行のみ見ることができます。単一行フィールドは並行スクロールできず、また、フィールドの長さを超えるテキストは許可されていません。複数行が選択されている場合、フィールドを複数行で構成することができます。フィールドのサイズは複数行に対応するよう調整する必要があります。フィールドのサイズは自動的に変更されません。テキストの各行に下線が付きます。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。

表 28. フィールド・プロパティ (続き)

プロパティ	説明
位置調整	テキストを左そろえまたは右そろえに位置合わせします。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
最大文字数	フィールドに入力できる最大文字数です。ユーザーは指定された最大文字数より多く入力することはできません。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
左方	制御の左端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
トップ	制御の上端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
幅	制御の幅 (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。
高さ	制御の高さ (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

#### 関連タスク:

60 ページの『制御』

#### 関連参照情報:

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

## Graffiti シフト・プロパティ

データの入力中に、上向きのペン・ストロークでテキストの大文字小文字を変更することができます。上向きストローク 1 回で大文字小文字シフトにします。上向きストローク 2 回でシフト・ロックします。上向きストローク 3 回でシフト・ロックを解除します。

この制御のサポートは、ターゲット・プラットフォームによって異なります。

表 29. Graffiti シフト・プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
左方	制御の左端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
トップ	制御の上端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
幅	制御の幅 (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

表 29. Graffiti シフト・プロパティ (続き)

プロパティ	説明
高さ	制御の高さ (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

**関連タスク:**

60 ページの『制御』

**関連参照情報:**

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

## Ink プロパティ

署名を BLOB データ・タイプとして保管するために使用します。

この制御のサポートは、ターゲット・プラットフォームによって異なります。

表 30. Ink プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
データ・ソース	制御が読み取り、書き込みを行うデータ・フィールドや列を含む表の名前です。使用可能な「データ・ソース」のリストは、そのプロジェクトにインポートされた表に基づいています。詳しくは、52 ページの『表定義ファイルの作成およびインポート』を参照してください。
データ・フィールド	制御が読み取り、書き込みを行う列の名前です。使用可能な「データ・フィールド」のリストは、「データ・ソース」プロパティで選択された表に基づいています。
左方	制御の左端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
トップ	制御の上端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
幅	制御の幅 (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。
高さ	制御の高さ (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

**関連タスク:**

60 ページの『制御』

#### 関連参照情報:

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

## ラベル・プロパティ

単一行読み取り専用テキスト・ラベル。

表 31. ラベル・プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
表題	この制御で表示されるテキストです。
フォント	その制御の表題で使用されるフォントです。
使用可能	使用可能になっている制御は、ユーザーに表示されます。「使用可能」はこのプロパティのデフォルト設定です。使用不可能になっている制御はフォーム上にドローできませんが実行中はドローするようにプログラムできます。
バックグラウンド	この制御の背景の色です。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
フォアグラウンド	この制御の前景の色です。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
左方	制御の左端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
トップ	制御の上端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
幅	制御の幅 (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。
高さ	制御の高さ (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

---

#### 関連タスク:

60 ページの『制御』

#### 関連参照情報:

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

## リスト・プロパティ

静的表情報を表示します。最初に、「データ・ソースのリスト」ウィンドウで定義された「データ・ソース」および「データ・フィールド」で取り込まれます。タップ・イベントは、データベース・アクションを実行する際に使用される場合があります。リストの内容は 1 つの表、または別に定義されたリストが元となります。

表 32. リスト・プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
データ・ソース	制御が読み取り、書き込みを行うデータ・フィールドや列を含む表の名前です。使用可能な「データ・ソース」のリストは、そのプロジェクトにインポートされた表に基づいています。詳細については、『表定義ファイルの作成およびインポート』を参照してください。
リスト項目	リスト項目の右側の楕円をクリックしウィンドウをオープンして、そのリストにカスタム値を入力します。このプロパティはポップアップ・トリガーの内容を設定するために、よく使用されます。リストには 1 つのソースのみが許可されています。「データ・ソース」プロパティは、「リスト項目」プロパティと共に使用することはできません。
順序づけ	「順序づけ」の右側にある楕円をクリックし、ウィンドウを開きます。このウィンドウで、「データ・ソース」、「データ・フィールド」、昇順または降順を定義することができます。
特殊	特殊な行を検索するには「はい」を、すべての行を検索するには「いいえ」を選択します。「特殊」プロパティを設定すると、リスト制御が参照専用になります。リストで選択された行を、行を表示または更新するために使用しないでください (たとえば、リスト制御に表示されたレコードをタップして表示したレコード詳細フォームなど)。Mobile Application Builder は、「特殊」が使用されたとき、表示または更新する正確な行を判別できないためです。
限度	検索する行数を定義します。
選択基準	「選択基準」フィールドの右にある楕円をクリックして、データベース表の列を、既に定義してあるホスト変数に関連付けるためのウィンドウを開きます。詳しくは、『ホスト変数の使用』を参照してください。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
ホスト変数の更新	「ホスト変数の更新」の右側にある楕円をクリックし、ウィンドウを開きます。このウィンドウで、前に定義したどのホスト変数を、リスト選択結果で更新するかを定義することができます。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
使用可能	使用可能になっている制御は、ユーザーに表示されます。「使用可能」はこのプロパティのデフォルト設定です。使用不可になっている制御はフォーム上にドロウできませんが実行中はドロウするようにプログラムできます。
バックグラウンド	この制御の背景の色です。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
フォアグラウンド	この制御の前景の色です。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。

表 32. リスト・プロパティ (続き)

プロパティ	説明
左方	制御の左端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
トップ	制御の上端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
幅	制御の幅 (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。
高さ	制御の高さ (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

#### 関連タスク:

60 ページの『制御』

#### 関連参照情報:

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

## ポップアップ・トリガー・プロパティ

制御をタップした際に、ユーザーがドロップダウン・リストの多肢選択から選択できるようにします。リストの内容は、ポップアップ・トリガーを持つリスト制御にリンクすることで定義されます。ポップアップ・トリガーがクリックされると、アクションの命令が開始されます。次にリストが表示され、ユーザーは選択を行います。ポップアップ・トリガーのラベルが変わり選択したものが表示される前に、制御に関連するすべてのイベントが実行されます。

この制御のサポートは、ターゲット・プラットフォームによって異なります。

表 33. ポップアップ・トリガー・プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
表題	制御の右方に表示される表題です。
フォント	その制御の表題で使用されるフォントです。
使用可能	使用可能になっている制御は、ユーザーに表示されます。「使用可能」はこのプロパティのデフォルト設定です。使用不可能になっている制御はフォーム上にドロウできませんが実行中はドロウするようにプログラムできます。
左方アンカー	左方アンカーが選択されている場合、制御の左方結合が固定されます。左方アンカーはこのプロパティのデフォルト設定です。右方アンカーが選択されている場合、制御の右方結合が固定されます。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。

表 33. ポップアップ・トリガー・プロパティ (続き)

プロパティ	説明
リスト ID	ポップアップ・トリガーと関連するリストの名前です。ドロップダウン・リストからリストを選択します。
左方	制御の左端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
トップ	制御の上端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
幅	制御の幅 (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。
高さ	制御の高さ (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

#### 関連タスク:

60 ページの『制御』

68 ページの『ポップアップ・トリガーの使用』

#### 関連参照情報:

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

## プッシュボタン・プロパティ

同じグループ ID 内のすべてのプッシュボタンから、ユーザーが 1 つのボタンを選択できるようにします。同じグループ ID 内のプッシュボタンは、互いに排他的です。ボタン制御に関しては、同じイベントとアクションがサポートされます。

この制御のサポートは、ターゲット・プラットフォームによって異なります。

表 34. プッシュボタン・プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
表題	制御のテキストです。
ビットマップ	選択されていない状態で、制御上に表示されるビットマップ・イメージです。ここでビットマップ・イメージを指定できるようにするためには、事前にそのビットマップ・イメージをプロジェクトに追加しておく必要があります。ビットマップを制御上に表示するためには、「ビットマップ」と「選択されたビットマップ」の両方のプロパティ用にイメージを指定しておかなければなりません。これらのイメージは、同じでも構いません。

表 34. プッシュボタン・プロパティ (続き)

プロパティ	説明
選択されたビットマップ	選択された状態で、制御上に表示されるビットマップ・イメージです。ここでビットマップ・イメージを指定できるようにするためには、事前にそのビットマップ・イメージをプロジェクトに追加しておく必要があります。ビットマップを制御上に表示するためには、「ビットマップ」と「選択されたビットマップ」の両方のプロパティ用にイメージを指定しておかなければなりません。これらのイメージは、同じでも構いません。
フォント	その制御の表題で使用されるフォントです。
使用可能	使用可能になっている制御は、ユーザーに表示されます。「使用可能」はこのプロパティのデフォルト設定です。使用不可になっている制御はフォーム上にドローできませんが実行中はドローするようにプログラムできます。
左方アンカー	左方アンカーが選択されている場合、制御の左方結合が固定されます。左方アンカーはこのプロパティのデフォルト設定です。右方アンカーが選択されている場合、制御の右方結合が固定されます。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
グループ ID	表の同じ列にマップされるプッシュボタンごとの固有の番号です。Mobile Application Builder は、同じ列にリンクするプッシュボタンごとに固有の値を自動的に割り当てます。非表示制御はアプリケーションの一部として見なされず、ドローできません。
左方	制御の左端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
トップ	制御の上端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
幅	制御の幅 (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。
高さ	制御の高さ (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

**関連タスク:**

60 ページの『制御』

**関連参照情報:**

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

## 繰り返しボタン・プロパティ

このボタンは、ペンを上げるまでペン・ダウン・イベントで繰り返し選択されます。ボタン制御に関しては、同じイベントとアクションがサポートされます。

この制御のサポートは、ターゲット・プラットフォームによって異なります。

表 35. 繰り返しボタン・プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
表題	制御のテキストです。
ビットマップ	選択されていない状態で、制御上に表示されるビットマップ・イメージです。ここでビットマップ・イメージを指定できるようにするためには、事前にそのビットマップ・イメージをプロジェクトに追加しておく必要があります。ビットマップを制御上に表示するためには、「ビットマップ」と「選択されたビットマップ」の両方のプロパティ用にイメージを指定しておかなければなりません。これらのイメージは、同じでも構いません。
選択されたビットマップ	選択された状態で、制御上に表示されるビットマップ・イメージです。ここでビットマップ・イメージを指定できるようにするためには、事前にそのビットマップ・イメージをプロジェクトに追加しておく必要があります。ビットマップを制御上に表示するためには、「ビットマップ」と「選択されたビットマップ」の両方のプロパティ用にイメージを指定しておかなければなりません。これらのイメージは、同じでも構いません。
フォント	その制御の表題で使用されるフォントです。
使用可能	使用可能になっている制御は、ユーザーに表示されます。「使用可能」はこのプロパティのデフォルト設定です。使用不可になっている制御はフォーム上にドロウできませんが実行中はドロウするようにプログラムできます。
左方アンカー	左方アンカーが選択されている場合、制御の左方結合が固定されます。左方アンカーはこのプロパティのデフォルト設定です。右方アンカーが選択されている場合、制御の右方結合が固定されます。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
フレーム	フレームが選択されている場合、制御の周りに角が丸い長方形のフレームが描かれます。フレームなしが選択されている場合、制御の周りに境界線は描かれませんが、太いフレームが選択されている場合、制御の周りに角が丸く太い長方形のフレームが描かれます。
グループ ID	表の同じ列にマップされるプッシュボタンごとの固有の番号です。Mobile Application Builder は、同じ列にリンクするプッシュボタンごとに固有の値を自動的に割り当てます。非表示制御はアプリケーションの一部として見なされず、ドロウできません。
左方	制御の左端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
トップ	制御の上端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。

表 35. 繰り返しボタン・プロパティ (続き)

プロパティ	説明
幅	制御の幅 (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。
高さ	制御の高さ (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

**関連タスク:**

60 ページの『制御』

**関連参照情報:**

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

## スクロール・バー・プロパティ

「フィールド」または「リスト」で使用されます。

表 36. スクロール・バー・プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
使用可能	使用可能になっている制御は、ユーザーに表示されます。「使用可能」はこのプロパティのデフォルト設定です。使用不可になっている制御はフォーム上にドローできませんが実行中はドローするようにプログラムできます。
現行値	スクロール・ボックスの先頭現行値です。
最小値	スクロール・バーの先頭にあるときのスクロール・ボックスの位置です。ほとんどのアプリケーションでは、デフォルトはゼロです。
最大値	スクロール・バーの一番下にあるときのスクロール・ボックスの位置です。
ページ・サイズ	一度にスクロールする行数です。
左方	制御の左端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
トップ	制御の上端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
幅	制御の幅 (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。
高さ	制御の高さ (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

---

**関連タスク:**

60 ページの『制御』

**関連参照情報:**

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

## 選択トリガー・プロパティ

長方形のフレームで枠取りされたラベルを表示します。これによりスクリプトの実行イベントを呼び出し、表示されたテキストをユーザーが変更できるようにします。

表 37. 選択トリガー・プロパティ

プロパティ	説明
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
表題	制御用に表示されるテキストです。
使用可能	使用可能になっている制御は、ユーザーに表示されます。「使用可能」はこのプロパティのデフォルト設定です。使用不可になっている制御はフォーム上にドロウできませんが実行中はドロウするようにプログラムできます。
左方アンカー	左方アンカーが選択されている場合、制御の左方結合が固定されます。左方アンカーはこのプロパティのデフォルト設定です。右方アンカーが選択されている場合、制御の右方結合が固定されます。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
左方	制御の左端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
トップ	制御の上端から画面の端までの距離です。このプロパティは制御の位置を手動で設定する際に使用します。
幅	制御の幅 (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。
高さ	制御の高さ (ピクセル表示) です。このプロパティは制御のサイズを手動で設定する際に使用します。

---

**関連タスク:**

60 ページの『制御』

**関連参照情報:**

127 ページの『パレット制御アイコン』

128 ページの『プロパティの制御』

---

## リソース・プロパティ

Mobile Application Builder は、アプリケーションで使用する多くのリソースを提供します。リソースのプロパティおよびイベントを設定し、それらをスクリプトで使用します。

---

### Mobile Application Builder リソース・プロパティ:

『フォーム・プロパティ』

149 ページの『アラート・プロパティ』

150 ページの『ストリング・プロパティ』

151 ページの『メニュー・プロパティ』

151 ページの『メニュー・バー・プロパティ』

152 ページの『アイコン・プロパティ』

153 ページの『ビットマップ・プロパティ』

---

### 関連タスク:

72 ページの『プロジェクト・リソース』

### 関連参照情報:

128 ページの『プロパティの制御』

## フォーム・プロパティ

表 38. フォーム・プロパティ: 記述

プロパティ	記述
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
表題	「フォーム」タブに表示されるテキストです。このテキストは、「プロジェクト」ペインにも表示されます。
左方	フォームの左端から画面の端までの距離です。
トップ	フォームの上端から画面の端までの距離です。
幅	フォームの幅 (ピクセル表示) です。標準的な Palm フォームの幅は 160 です。
高さ	フォームの高さ (ピクセル表示) です。標準的な Palm フォームの高さは 160 です。
モーダル	真の場合、ユーザーは継続する前にこのフォームに応答しなければなりません。このフォームはクローズされるまで、他のすべてのフォームの最前面にとどまります。

表 38. フォーム・プロパティ (続き): 記述

プロパティ	記述
メニュー ID	フォームと関連付けるメニュー・バーの名前を入力します。
データ・ソース	情報を含む DB2 表の名前です。
データ・フィールド	このフォームが関連する列の名前です。
リンク・ソース	リンクする DB2 表の名前です。
リンク・フィールド	リンクするキーを含む列の名前です。
選択基準	<p>選択基準の右の</p>  <p>をクリックして、「列とホスト変数との関連付け」ウィンドウを開きます。この「列とホスト変数との関連付け」ウィンドウを使用して、データベース表の列を、既に定義してあるホスト変数に関連付けます。詳しくは、『ホスト変数』を参照してください。</p>
ヘルプ ID	モーダル・フォームのヘルプを表示する際に使用するストリングの ID です。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。
バックグラウンド	この制御の背景の色です。
フォアグラウンド	この制御の前景の色です。

#### 関連タスク:

72 ページの『プロジェクト・リソース』

56 ページの『フォーム』

#### 関連参照情報:

128 ページの『プロパティの制御』

## アラート・プロパティ

アラートは、ユーザーの応答を要求するプロンプトを出すのに使用されます。ユーザーは処理を行う前に、アラートに応答する必要があります。スクリプトを使用してアラートを参照します。

アラートのサポートは、ターゲット・プラットフォームによって異なります。

表 39. アラート・プロパティ: 記述

プロパティ	記述
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
表題	テキストに表示されるテキストです。このテキストは、「プロジェクト」ペインにも表示されます。

表 39. アラート・プロパティ (続き): 記述

プロパティ	記述
アラート・タイプ	以下のいずれかのアラート・タイプを選択してください。ダイアログ上でタイプごとに異なるアイコンが表示されます。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。 <ul style="list-style-type: none"> <li>• 確認</li> <li>• 警告</li> <li>• 情報</li> <li>• エラー</li> </ul>
メッセージ	アラートで表示されるメッセージです。
ボタン・テキスト	アラート・ボタンに表示されるテキストです。
デフォルト・ボタン ID	アラート・デフォルト・ボタン ID です。アラートに表示させようとしているボタンの数です。ID=0 は、ボタンを 1 つ表示することを意味します。ID=1 は、ボタンを 2 つ表示することを意味し、以下同様です。このプロパティのサポートは、ターゲット・プラットフォームによって異なります。

#### 関連タスク:

- 72 ページの『プロジェクト・リソース』
- 72 ページの『新規アラートの作成』

#### 関連参照情報:

- 128 ページの『プロパティの制御』

## ストリング・プロパティ

文字のストリングを表示するには、ストリングを使用します。ストリングを使用して、バージョン番号などのアプリケーションに関するメッセージを表示します。ストリングの参照には、スクリプトを使用します。

ストリングのサポートは、ターゲット・プラットフォームによって異なります。

表 40. ストリング・プロパティ: 記述

プロパティ	記述
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
ストリング	ストリングのテキストです。このテキストは、「プロジェクト」ペインにも表示されます。
ストリング・ファイル名	ストリングを含むファイルの名前です。ストリング・ファイルはプロジェクト・ディレクトリーに保管します。

---

**関連タスク:**

72 ページの『プロジェクト・リソース』

81 ページの『新規ストリングの作成』

**関連参照情報:**

128 ページの『プロパティの制御』

## メニュー・プロパティ

メニューまたはメニュー項目は、メニュー・バーからアクセス可能な個々の機能で、画面上の他の場所にはありません。

表 41. メニュー・プロパティ: 記述

プロパティ	記述
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
メニュー項目テキスト	メニュー項目のテキストです。このテキストは、関連するメニュー・バーのドロップダウン・リストに表示されます。このテキストは、「プロジェクト」ペインにも表示されます。
アクセラレーター文字	メニュー項目のショートカット・キーボード文字です。ショートカットを示す 1 文字を入力します。ショートカットが、同時に使用可能な他のメニュー項目で使用されていないことを確認してください。

---

**関連タスク:**

72 ページの『プロジェクト・リソース』

79 ページの『新規メニュー項目の作成』

**関連参照情報:**

128 ページの『プロパティの制御』

## メニュー・バー・プロパティ

メニュー・バーは、画面上の他の場所で使用できない追加機能 (メニュー項目) へのアクセスを提供します。メニュー・ボタンがタップされると、メニュー・バーが使用可能になります。アプリケーション内のフォームごとに複数のメニュー・バーを持つことができます。

表 42. メニュー・バー・プロパティ：記述

プロパティ	記述
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
プルダウン	各メニュー・プルダウンの上端にあるメニュー・バー上に表示されるテキストまたは表題です。このテキストは、「プロジェクト」ペインにも表示されます。
メニュー項目 ID	メニュー・バーの下にリストするメニューの名前です。メニューは、プロパティ・ページで表示されるのと同じ順序で、メニュー・バー・プルダウン・リストに表示されます。

#### 関連タスク:

- 72 ページの『プロジェクト・リソース』
- 80 ページの『新規メニュー・バーの作成』

#### 関連参照情報:

- 128 ページの『プロパティの制御』

## アイコン・プロパティ

各 Palm アプリケーションには、「Palm アプリケーション」ペインに表示されるアイコンが含まれます。

アイコンのサポートは、ターゲット・プラットフォームによって異なります。

表 43. アイコン・プロパティ：記述

プロパティ	記述
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
アイコン・ファイル名	アイコンのパスおよびファイル名です。アイコンはプロジェクト・ディレクトリーに保管することをお勧めします。

#### 関連タスク:

- 72 ページの『プロジェクト・リソース』
- 77 ページの『新規アイコンの作成』

#### 関連参照情報:

- 128 ページの『プロパティの制御』

## ビットマップ・プロパティ

フォーム上でビットマップ・イメージを表示します。ビットマップ・イメージは、白黒または直接色形式 (Palm OS 4.0 またはそれ以上でサポートされます) でビットマップ・イメージを保管できる、任意の外部ドロー・プログラムを使用して作成することができます。ビットマップは、アプリケーションのプロジェクト・ディレクトリーに保管します。フォーム・ビットマップ制御をフォームに追加して、ビットマップを表示してください。

ビットマップのサポートは、ターゲット・プラットフォームによって異なります。

表 44. ビットマップ・プロパティ：記述

プロパティ	記述
名前	制御のための固有なシステム生成名です。ユーザーには表示されません。編集はできません。この制御名を使用するには、スクリプトを更新します。
ビットマップ・ファイル名	ビットマップを含むファイルの名前です。ビットマップは、プロジェクト・ディレクトリーに保管します。
圧縮タイプ	以下のいずれかのビットマップ圧縮タイプを選択してください。 <ul style="list-style-type: none"><li>• NOCOMPRESS: ビットマップを圧縮しません</li><li>• COMPRESS</li><li>• FORCECOMPRESS: ビットマップ圧縮を強制的に行います</li></ul>

### 関連タスク:

72 ページの『プロジェクト・リソース』

75 ページの『ビットマップの作成および表示』

### 関連参照情報:

128 ページの『プロパティの制御』

## イベントの説明

Mobile Application Builder イベントのサポートは、制御およびターゲット・プラットフォームによって異なります。

### Mobile Application Builder イベント

#### イベント

表 45. イベントの説明:

イベント	対応する Palm イベント	記述
制御イベント		
ペンを下ろす	ctlEnterEvent	制御上でペンを下ろします。

表 45. イベントの説明 (続き):

イベント	対応する Palm イベント	記述
<b>制御イベント</b>		
終了	ctlExitEvent	制御内でペンを下ろし、制御外で持ち上げます。
繰り返し	ctlRepeatEvent	ペンを下ろし、その間にもう一つ取得します。繰り返しボタンと一緒に使用されます。
タップ	ctlSelectEvent	制御内でペンを下ろし、持ち上げます。
<b>フィールド・イベント</b>		
スクロール	fldChangedEvent	ドラッグ選択したためにスクロールされたフィールドです。
選択	fldEnterEvent	フィールドの境界でペンを下ろします。
高さの変更	fldHeightChangedEvent	フィールドの高さに変更されました。通常は開発者によって処理されません。
<b>フォーム・イベント</b>		
オープン	Form init	このイベントは、フォームがオープンするたびに実行されます。
<b>フォーム・イベントのバーコード・スキャン</b>		
スキャンの初期化	ScannerInit	このイベントは、下記の InitializeScanner アクションに関連しています。フォーム・オープン・イベントの直後にこのイベントが呼び出されます。
デコード・スキャン	ScannerDecode	このイベントは、DecodeScanTo アクションに関連しています。ハードまたはソフト・スキャンの後に、このイベントが呼び出されます。
デコードなし	NoDecode	このイベントは、バーコードでレーザーが起動され、バーコードが正常にデコードされなかった場合に発生します。このイベントには任意のアクションを関連付けることができます。アクションが指定されない場合、このイベントの発生時に、ユーザーにデフォルトのエラー・メッセージが送達されます。
スキャンのクローズ	ScannerClose	このイベントは、DeInitScanner アクションに関連しています。フォーム・クローズ・イベントの直後にこのイベントが呼び出されます。
<b>リスト・イベント</b>		
ペンを下ろす	lstEnterEvent	リスト内でペンを下ろします。
終了	lstExitEvent	リスト内でペンを下ろし、リスト外で持ち上げます。
タップ	lstSelectEvent	同じリスト内でペンを下ろし、持ち上げます。
<b>ポップアップ・トリガー・イベント</b>		

表 45. イベントの説明 (続き):

イベント	対応する Palm イベント	記述
<b>制御イベント</b>		
ポップアップ	popSelectEvent	ポップアップ・リストから選択した項目です。
<b>スクロール・イベント</b>		
ペンを下ろす	sclEnterEvent	制御内でペンを下ろします。
終了	sclExitEvent	制御内でペンを下ろし、外で持ち上げます。
スクロール	sclRepeatEvent	ペンを下ろし、引き続き保持します。
<b>メニュー・イベント</b>		
メニューの選択	menuEvent	強調表示メニュー項目にペンを下ろし、強調表示されたメニュー項目でペンを持ち上げます。

#### 関連タスク:

82 ページの『イベント、アクション、およびターゲット』

#### 関連参照情報:

『アクションの説明』

## アクションの説明

Mobile Application Builder アクションのサポートは、イベントおよびターゲット・プラットフォームによって異なります。

### Mobile Application Builder アクション

#### アクション

表 46. アクションの説明:

アクション	記述
アクションなし	アクションは起きません。
表示	ターゲット・フォームまたはダイアログの表示
フォームの消去	ターゲット・フォーム上のフィールド、インク制御、または blob 制御をすべて消去します。
同期アプリケーション	同期サーバーを使用して、バックエンド・データベースとの同期を初期化します。
フォームの印刷	フォームの内容を印刷します。
直前のフォームに戻る	直前のフォームをオープンします。
消去	フォーム上のターゲット・フィールド、blob 制御、またはインク制御を選択して消去します。

表 46. アクションの説明 (続き):

アクション	記述
イメージの選択	ユーザーがファイル・システムからイメージを選択できるファイル・ダイアログを開いて、このイメージを Blob 制御で表示します。これにより現行イメージがアクティブ・イメージになり、ユーザーがレコードを挿入する場合に、選択されたイメージが挿入されます。
イメージの保管	Blob 制御に関連付けられた Blob イメージを、デバイス上のファイル・システムに保管します。
日付表示ピッカー	エンド・ユーザーによる、日付ピッカーを使用した日付の選択が可能なダイアログを開きます。日付はターゲット・フィールドに挿入されます。
時刻表示ピッカー	エンド・ユーザーによる、時刻ピッカーを使用した時刻の選択が可能なダイアログを開きます。時刻はターゲット・フィールドに挿入されます。
現在の状態での表示	このアクションのサポートは、ターゲット・プラットフォームによって異なります。ターゲット・フォームに焦点を合わせます (たとえば、フォームを可視にします) が、フォームの再取り込みは行いません。現在の状態で表示されます。
アプリケーションのクローズ	このアクションのサポートは、ターゲット・プラットフォームによって異なります。アプリケーションをクローズします。ターゲット・プラットフォームによっては、ユーザーが新規アプリケーションに切り替えたときに前のアプリケーションが自動的にクローズされるため、このアクションが必要ないものもあります。
スキャナーの初期化	バーコード・スキャンを行なうために、スキャナー・ハードウェアを初期化します。
ソフト・スキャン	このアクションは、ボタンまたはメニュー項目のタップ時にレーザーを起動するために使用する必要があります。バーコード・スキャンを実行するためにレーザー光線を起動するハードウェア・ボタンがターゲット・デバイスにない場合は、このアクションを組み込むことが必須です。
スキャナーのクローズ	フォームのバーコード・スキャン認識をクローズさせます。
スキャンのデコード	スキャンしたデータをフィールドにデコードします。
レコードの作成	ターゲット表にレコードを作成します。
レコードの削除	ターゲット表のレコードを削除します。
レコードの更新	ターゲット表のレコードを更新します。
次のレコードに移動	ターゲット表の次のレコードが表示されます。
直前のレコードに移動	ターゲット表の直前のレコードが表示されます。
最初のレコードに移動	ターゲット表の最初のレコードが表示されます。
最後のレコードに移動	ターゲット表の最後のレコードが表示されます。
スクリプトの実行	ターゲット・スクリプトが実行されます。
消去	フォーム上のターゲット・フィールドと ink 制御を選択して消去します。

---

**関連タスク:**

82 ページの『イベント、アクション、およびターゲット』

**関連参照情報:**

153 ページの『イベントの説明』

---

## スクリプト記述サポート

スクリプト記述のサポートは、ターゲット・プラットフォームによって異なります。下記のプラットフォームをクリックして、各プラットフォームのスクリプト記述サポートに関する情報を参照してください。

---

**Mobile Application Builder スクリプト記述サポート:**

『Java コード・プラットフォームのためのスクリプト記述サポート』

158 ページの『Palm C コード・プラットフォームのためのスクリプト記述サポート』

---

**関連タスク:**

87 ページの『Mobile Application Builder 中のスクリプト』

## Java コード・プラットフォームのためのスクリプト記述サポート

現時点では、Mobile Application Builder は、ポイドを戻すイベント・スクリプト関数のみをサポートし、ユーザーが作成した関数からの戻り値を使用する機能はありません。ただし、グローバル・スクリプトを使用して、値を戻す関数を記述し、後にイベント・スクリプトでそれらを使用することができます。

より詳細なスクリプト記述を行うには、プロジェクト用に生成された Java コードを手作業で編集し、バッチ・ファイルを実行する必要があります。ほとんどのコード生成ツールと同様、Mobile Application Builder は、Java コードに手作業で行われた変更を認識せず、また使用もしません。重要: 手作業による Java コードへの変更はすべて、Mobile Application Builder 内からアプリケーションを再度ビルドすると失われます。

---

**関連タスク:**

87 ページの『Mobile Application Builder 中のスクリプト』

## Palm C コード・プラットフォームのためのスクリプト記述サポート

現時点では、Mobile Application Builder は、ポイドを戻すイベント・スクリプト関数のみをサポートし、ユーザーが作成した関数からの戻り値を使用する機能はありません。ただし、グローバル・スクリプトを使用して、値を戻す関数を記述し、後にイベント・スクリプトでそれらを使用することができます。

より詳細なスクリプト記述を行うには、プロジェクト用に生成された C コードを手作業で編集し、make ファイルを実行する必要があります。ほとんどのコード生成ツールと同様、Mobile Application Builder は、C コードに手作業で行われた変更を認識せず、また使用もしません。重要: 手作業による C コードへの変更はすべて、Mobile Application Builder 内からアプリケーションを再度ビルドすると失われます。

---

### 関連タスク:

87 ページの『Mobile Application Builder 中のスクリプト』

---

## 印刷サポート

Mobile Application Builder は、Palm OS デバイスに、限定された印刷サポートを提供しています。「プリファレンス」で各プロジェクトのためのデフォルト印刷ソリューションを設定することによって、印刷機能にアクセスすることができます。印刷機能のサポートは、ターゲット・プラットフォームによって異なります。

印刷は、フォーム・レベルにインプリメントされており、以下のような特性を持ちます。

- すべてのラベル、フィールド、およびリスト制御の内容が印刷されます。その他の制御は印刷されません。
- ラベルおよびフィールドまたは 2 つのリストを、お互い隣同士に印刷できます。そのためには、両方の制御の Y 座標 (プロパティ上端の値) を同じ値に設定してください。許容度は 2 ピクセルが許されます。
- 制御は、以下のようにフォーム上の位置合わせを基にして印刷されます。
  - 左から右
  - 上から下
- 正確な印刷動作 (たとえば、長い文字ストリングの場合の折り返しと切り捨て) は、使用するプリンターによって決定されます。

### サポートされる印刷テクノロジー

#### PalmPrint

PalmPrint は、Stevens Creek Software が開発した Palm OS 印刷ソリューションです。これを使用するとご使用のアプリケーションから、すべての PalmOS 3.0/4.0 デバイス (Palm III、V、VII、m100、および m500 のほか、Symbol SPT 1500 および 1700、IBM WorkPad、Handspring Visor、および Sony Clie を含む) から、赤外線を通じて印刷することができます。

PalmPrint ソリューションは、あらかじめ Stevens Creek Software から入手しておく必要があります。入手後にこのソリューションを使用して、Mobile Application Builder の印刷機能にアクセスできます。

#### 関連タスク:

101 ページの『アプリケーションへの印刷機能の追加』

#### 関連参照情報:

『印刷 API』

## 印刷 API

以下の印刷 API 呼び出しを使用して、印刷の動作をカスタマイズします。

印刷機能のサポートは、ターゲット・プラットフォームによって異なります。

表 47. 印刷 API 呼び出し、概要

印刷ソリューション	API
MAB_PRINT_VIA_PALMPRINT	MAB_Print_SetAlertIDForMessages
	MAB_Print_InitPrinter
	MAB_Print_Line
	MAB_Print_Buffer
	MAB_Print_ClosePrinter

#### 戻りコード

表 48. 戻りコード

状況	記述
MAB_PRINT_SUCCESS_CODE	印刷操作が成功しました。
MAB_PRINT_ERROR_UNKNOWN_CODE	印刷操作が失敗しました。

#### MAB\_Print\_SetAlertIDForMessages

**目的:** この関数は、エラー・メッセージのアラート ID を設定するために使用します。この値が設定されない場合、DBInformationAlert という名前の MAB システム・アラートがデフォルトとなります。カスタム・アラートでエラー・メッセージを表示するには、上記 API のいずれかを呼び出す前にこれを設定してください。

#### 関数プロトタイプ

```
void MAB_Print_SetAlertIDForMessages(UIInt16 alertID)
```

#### 関数の引き数:

データ・タイプ	引き数	記述
Word	alertID	カスタム・アラートの ID。

戻り: なし。

#### 使用法

```
MAB_Print_SetAlertIDForMessages (Alert1);
```

#### MAB\_Print\_InitPrinter

**目的:** この関数は、1 行ずつ印刷するようにプリンターを初期化するために使用します。印刷に使用する印刷ライブラリーを渡します。この関数は、バッファの印刷またはフォームの印刷には必要ありません。

#### 関数プロトタイプ

```
MAB_PRINT_STATUS MAB_Print_InitPrinter(MAB_PrintLibrary lib)
```

#### 関数の引き数:

データ・タイプ	引き数	記述
MAB_PrintLibrary	Lib	有効な印刷ライブラリーです。サポートされる印刷ライブラリーは、前述したものです。

戻り: MAB\_PRINT\_STATUS

#### 使用法

```
MAB_Print_InitPrinter(MAB_PRINT_VIA_PALMPRINT);
```

#### MAB\_Print\_Line

**目的:** この関数は、1 行印刷するために使用します。この API を呼び出す前に、MAB\_Print\_InitPrinter を使用してプリンターを初期化しておく必要があります。MAB\_Print\_Line 呼び出しに続けて、MAB\_Print\_ClosePrinter を呼び出す必要があります。

#### 関数プロトタイプ

```
MAB_PRINT_STATUS MAB_Print_Line(char *buffer)
```

**関数の引き数:**

データ・タイプ	引き数	記述
Char *	Buffer	データ

戻り: MAB\_PRINT\_STATUS

**使用法**

```
MAB_Print_Line("Hello");
```

**MAB\_Print\_Buffer**

**目的:** この関数は、バッファを印刷するために使用します。バッファには、複数の行が含まれている場合があります。プリンターを初期化またはクローズする必要はありません。

**関数プロトタイプ**

```
MAB_PRINT_STATUS MAB_Print_Buffer(MAB_PrintLibrary lib, char *buffer)
```

**関数の引き数:**

データ・タイプ	引き数	記述
MAB_PrintLibrary	Lib	サポートされるライブラリー名。
Char *	Buffer	データ

戻り: MAB\_PRINT\_STATUS

**使用法**

```
MAB_Print_Buffer(MAB_PRINT_VIA_PALMPRINT, "Hello");
```

**MAB\_Print\_ClosePrinter**

**目的:** この関数は、プリンターをクローズするために使用します。

**関数プロトタイプ**

```
MAB_PRINT_STATUS MAB_Print_ClosePrinter()
```

**関数の引き数:** なし。

戻り: MAB\_PRINT\_STATUS

**使用法**

```
MAB_Print_ClosePrinter();
```

---

#### 関連タスク:

101 ページの『アプリケーションへの印刷機能の追加』

#### 関連参照情報:

158 ページの『印刷サポート』

---

## バーコード・スキャン API

以下のバーコード・スキャン API 呼び出しを使用して、バーコード・スキャンの動作をカスタマイズします。API は Palm バーコード・スキャン・アプリケーションでのみサポートされ、WinCE ではサポートされません。

バーコード・スキャンのサポートは、ターゲット・プラットフォームによって異なります。

表 49. バーコード・スキャン API 呼び出し、概要

バーコード・スキャン API
MAB_ScanSetAlertIDForMessages
MAB_ScanSetAimDuration
MAB_ScanSetTriggeringMode
MAB_ScanGetLastErrorCode
MAB_ScanGetLastErrorMessage
MAB_ScanDisableBarCodeType
MAB_ScanEnableBarCodeType
MAB_InitializeScanner
MAB_DoSoftScan
MAB_DecompileScan
MAB_ScanHandleNoDecode
MAB_DeInitScanner
MAB_ScanErrorHandler

## 戻りコード

表 50. 戻りコード

状況	記述
MAB_SCAN_STATUS_OK	正常終了。エラーは戻りませんでした。
MAB_SCAN_UNKNOWN_ERROR	内部エラーが発生しました。
MAB_SCAN_BARCODE_NOT_SUPPORTED	このバーコード・タイプは、サポートされていないか、このアプリケーションには使用可能ではありません。
MAB_SCAN_COMMUNICATIONS_ERROR	ハードウェアと通信できませんでした。

表 50. 戻りコード (続き)

MAB_SCAN_BAD_PARAM	スキャナーを初期化するために使用された 1 つまたは複数のパラメーターが範囲外でした。
MAB_SCAN_BATCH_ERROR	初期化に使用されたパラメーターが多すぎます。
MAB_SCAN_NODECODE	スキャナーがバーコードを正常にデコードできませんでした。
MAB_SCAN_SOFTSCAN_FAILED	ソフト・スキャンの開始に失敗しました。
MAB_SCAN_CURRENT_FIELD_NOT_SCAN_AWARE	現行フィールドがスキャン認識ではありません。

### MAB\_ScanSetAlertIDForMessages

**目的:** この関数は、エラー・メッセージのアラート ID を設定するために使用します。この値が設定されない場合、DBErrorAlert という名前の MAB システム・アラートがデフォルトとなります。カスタム・アラートで エラー・メッセージを表示するには、MAB\_InitializeScanner を呼び出す前にこれを設定してください。

#### 関数プロトタイプ

```
void MAB_ScanSetAlertIDForMessages(Word alertID)
```

#### 関数の引き数:

データ・タイプ	引き数	記述
Word	alertID	カスタム・アラートの ID。

戻り: なし。

#### 使用法

```
MAB_ScanSetAlertIDForMessages(Alert1);
```

### MAB\_ScanSetAimDuration

**目的:** この関数は、バーコードのスキャン中にレーザーを持続させる時間の長さを設定するために使用します。指定の時間間隔の前に正常なデコードが行われない場合、レーザーはその指定時間のみ持続されます。このパラメーターのデフォルト値は 20 です。

#### 関数プロトタイプ

```
Boolean MAB_ScanSetAimDuration(UInt16 duration)
```

#### 関数の引き数:

データ・タイプ	引き数	記述
UInt16	Duration	スキャン中にレーザーを持続させる時間の長さです。このパラメーターの有効な範囲は 5 から 99 です。ここでの 20 という値は、レーザーが 2.0 秒間 持続することを意味します。

戻り: ブール - 渡された値が範囲内の場合は真、そうでなければ偽。

#### 使用法

```
MAB_ScanSetAimDuration(20);
```

#### MAB\_ScanSetTriggeringMode

**目的:** この関数は、スキャナーの起動モードを設定します。ここでの値によって、フォームがハード・スキャンで使用可能とするか、ソフト・スキャンで使用可能とするかが決定されます。デフォルト値は HOST が前提とされます。これは、両方のタイプのスキャンを使用可能にします。

#### 関数プロトタイプ

```
Boolean MAB_ScanSetTriggeringMode(UInt16 mode)
```

#### 関数の引き数:

データ・タイプ	引き数	記述
UInt16	Mode	HOST: ユーザーがアプリケーションで ソフト・スキャンとハード・スキャンの両方を行う必要があるときに、これを設定することが必要です。 LEVEL: ユーザーがアプリケーションで ハード・スキャンのみを行う必要があるときに、このモードを設定することが必要です。この場合、正常なデコードが行われたとき、またはハードウェア起動ボタンが離されたときに、レーザーがオフにされます。

戻り: なし。

## 使用法

```
MAB_ScanSetTriggeringMode(HOST);
```

## MAB\_ScanGetLastErrorCode

**目的:** この関数は、最後の MAB スキャナー API 呼び出しの結果得られるエラー・コードを取得するために使用します。

### 関数プロトタイプ

```
MAB_SCAN_STATUS MAB_ScanGetLastErrorCode()
```

**関数の引き数:** なし。

**戻り:** MAB\_SCAN\_STATUS

## 使用法

```
int errorcode = MAB_ScanGetLastErrorCode();
```

## MAB\_ScanGetLastErrorMessage

**目的:** この関数は、最後の MAB スキャナー API 呼び出しの結果得られるエラー・コードに対応するエラー・メッセージを取得するために使用します。

### 関数プロトタイプ

```
CharPtr MAB_ScanGetLastErrorMessage()
```

**関数の引き数:** なし。

**戻り:** CharPtr - メッセージ。

## 使用法

```
FrmCustomAlert(alertid,MAB_ScanGetLastErrorMessage(),"");
```

## MAB\_ScanDisableBarcodeType

**目的:** この関数は、バーコード・タイプがアプリケーション・ユーザーによってスキャンされないようにするために使用します。デフォルトでは、アプリケーションですべてのバーコード・タイプが使用可能にされます。

### 関数プロトタイプ

```
void MAB_ScanDisableBarcodeType(Word barcodeType);
```

関数の引き数:

データ・タイプ	引き数	記述
Word	Barcode type	MAB によってサポートされる任意のバーコード・タイプ。

戻り: なし。

使用法

```
MAB_ScanDisableBarCodeType(barUPCA);
```

### MAB\_ScanEnableBarCodeType

**目的:** この関数は、アプリケーションで必要なすべてのバーコード・タイプを使用可能にするために使用します。デフォルトでは、アプリケーションですべてのバーコード・タイプが使用可能になるので、これより前にアプリケーション内の他の箇所で MAB\_ScanDisableBarCodeType を使用してバーコード・タイプが使用不可にされたときに、この関数が役立ちます。

関数プロトタイプ

```
void MAB_ScanEnableBarCodeType(Word barCodeType);
```

関数の引き数:

データ・タイプ	引き数	記述
Word	Barcode type	MAB によってサポートされる任意のバーコード・タイプ。

戻り: なし。

使用法

```
MAB_ScanEnableBarCodeType(barUPCA);
```

### MAB\_InitializeScanner

**目的:** この関数は、スキャナー・ハードウェアを初期化します。この関数が呼び出された後、スキャナーは電源の受給を開始します。 Mobile Application Builder のアクション・スキャナーの初期化でもこの関数が使用されます。

関数プロトタイプ

```
MAB_SCAN_STATUS MAB_InitializeScanner()
```

関数の引き数: なし。

戻り: MAB\_SCAN\_STATUS

#### 使用法

```
if(MAB_InitializeScanner() == MAB_SCAN_STATUS_OK){
    // Successful initialization.
}
else{
    //display error - unsuccessful initialization
}
```

#### MAB\_DoSoftScan

**目的:** この関数は、ソフト・スキャンを開始するために使用します。この関数は、ボタンまたはメニュー項目への応答として、どこからでも呼び出すことができます。この関数は、Mobile Application Builder のアクション「ソフト・スキャンの実行」の結果として、呼び出されます。

#### 関数プロトタイプ

```
MAB_SCAN_STATUS MAB_DoSoftScan();
```

**関数の引き数:** なし。

戻り: MAB\_SCAN\_STATUS

#### 使用法

```
if(MAB_DoSoftScan() == MAB_SCAN_STATUS_OK){
    //successful initiation
}
else{
    //could not initiate a soft scan.
}
```

#### MAB\_DecodeScan

**目的:** この関数は、開始されたソフト・スキャンまたはハード・スキャンをデコードするために使用します。この関数は、イベント scanDecodeEvent のハンドラーであるスクリプトでのみ使用してください。この関数は、Mobile Application Builder のアクション「スキャンのデコード」によって使用されます。

#### 関数プロトタイプ

```
CharPtr MAB_DecodeScan (void);
```

**関数の引き数:** なし。

戻り: CharPtr - デコードが正常に終了した場合、デコードしたバーコードを返します。バーコードをデコードできなかった場合、NULL を返します。MAB\_SCAN\_STATUS を使用すると、最後の操作のエラー・コードを入手することができます。

## 使用法

```
...
        szText = MAB_DeCodeScan();
        if(szText){
            //successful decode
        }
        else{
            //no decode
        }
    }
```

## MAB\_DeInitScanner

**目的:** この関数は、スキャナー・ハードウェアを初期化解除するために使用します。この関数は、Mobile Application Builder のアクション「DeInitScanner」の結果として、呼び出されます。

### 関数プロトタイプ

```
MAB_SCAN_STATUS MAB_DeInitScanner(void);
```

**関数の引き数:** なし。

**戻り:** MAB\_SCAN\_STATUS

## 使用法

```
...
        szText = MAB_DeInitScanner ();
        if(szText){
            //deinit successful
        }
        else{
            //deinit failed.
        }
    }
```

## MAB\_ScanErrorHandler

**目的:** この関数は、最後に実行されたアクションの結果受け取る状況のエラー・メッセージを表示するために使用します。この関数では、関数 MAB\_ScanSetAlertIDForMessages を使用して設定された AlertID を利用します。

### 関数プロトタイプ

```
Boolean MAB_ScanErrorHandler(UInt16 status);
```

関数の引き数:

データ・タイプ	引き数	記述
UInt16	status	前述の状況のいずれか。

戻り: ブール - 常に真を戻します。

使用法

```
MAB_ScanErrorHandler(MAB_SCAN_NODECODE);
```

関連タスク:

103 ページの『アプリケーションへのバーコード・スキャン機能の追加』

関連参照情報:

『Palm アプリケーションでサポートされる 1 次元バーコード・タイプ』

---

## Palm アプリケーションでサポートされる 1 次元バーコード・タイプ

表 51. Palm アプリケーションでサポートされる 1 次元バーコード・タイプ

barCODE39, barUPCA, barUPCE, barUPCE1, barEAN13, barEAN8, barD25, barI2OF5, barCODABAR, barCODE128, barCODE93, barTRIOPTIC39, barUCC_EAN128, barMSI_PLESSEY, barUPCE1, barBOOKLAND_EAN, barISBN128, barCOUPON, barPDF417
--

関連タスク:

103 ページの『アプリケーションへのバーコード・スキャン機能の追加』

#### 関連参照情報:

162 ページの『バーコード・スキャン API』  
『WinCE アプリケーションでサポートされるバーコード・タイプ』

---

## WinCE アプリケーションでサポートされるバーコード・タイプ

表 52. WinCE アプリケーションでサポートされる 1 次元バーコード・タイプ

EAN8
EAN13
CODE39
UPCA
UPCE0
UPCE1
MSI
CODE93
CODE128

---

#### 関連タスク:

103 ページの『アプリケーションへのバーコード・スキャン機能の追加』

#### 関連参照情報:

169 ページの『Palm アプリケーションでサポートされる 1 次元バーコード・タイプ』

---

## トラブルシューティング

アプリケーションのビルドまたはテストで問題が発生した場合、下記に示した、考えられる解決方法のリストを参照してください。

### トラブルシューティング

#### Palm 開発環境のセットアップ・エラー

Palm 開発環境をセットアップする際に発生する可能性のある、一般的なエラー。

表 53. Palm 開発環境のセットアップ・エラー：説明

症状	考えられる解決方法
アプリケーションがビルドされません。	パスが正しく設定されていません。パスには、必ず次の内容を含めるようにしてください。 <ul style="list-style-type: none"><li>• &lt;cygwin install path&gt;%cygwin_b20%hi586_cygwin32%bin</li><li>• &lt;prc-tools install path&gt;%bin</li></ul>
「:cpp - ファイルまたはディレクトリが検出できません」というようなエラーで、Make が失敗しました。	MAKE_MODE が「UNIX」に設定されていません。Make が command.com ではなく sh.exe でコマンドを実行するように、環境変数 MAKE_MODE=UNIX を設定しておかなければなりません。

表 53. Palm 開発環境のセットアップ・エラー (続き): 説明

症状	考えられる解決方法
「:m68k-palmos-gcc - ファイルまたはディレクトリーが検出できません」というようなエラーで、Make が失敗しました。	マウントが正しくセットアップされていません。コンパイラーが palmdev ツールを検出できません。 bash シェルを実行して、マウントを認識するためにマウントをタイプしてください。必要なすべてのマウントがそこにあることを確認してください。
コンパイラー・エラー - PalmOS.h を検出できません。	SDK が正しくインストールされていないか、指定されていないか、あるいはセットアップされていません。 <ul style="list-style-type: none"> <li>• SDK が &lt;palmdev location&gt;/sdk-## にインストールされていることを確認してください。この ## は SDK のバージョン番号で、たとえば、/sdk-4.0 などです。</li> <li>• コンパイラーが使用する SDK のバージョンを -palmos## フラグで指定します。この ## は、SDK のバージョン番号です。</li> <li>• &lt;palmdev location&gt; ディレクトリーに、デフォルトで使用する SDK をポイントするシンボリック・リンク /sdk を作成します。</li> </ul>
コンパイラーがライブラリーまたはツールを検出できないことを示す、コンパイル・エラー。	GCC_EXEC_PREFIX が、いまだに古い GNU ツール・セットの設定になっています。GCC_EXEC_PREFIX 環境変数の設定を解除してください。
DB2 Everyplace アプリケーションはエラーなくビルドされましたが、エミュレーターまたは Palm デバイスでクラッシュが発生しました。	0.5.0 ツール・セットでコンパイルされた古い DB2eGNU.o にリンクしています。DB2eGNU.o を新しい PRC-Tools でコンパイルする必要があります。

### Palm ビルド・エラー

ビルドが正常に終了しなかったことを示す、Mobile Application Builder のエラー・メッセージが出力されます。メッセージ・ログを表示してください。

表 54. Palm ビルド・エラー: 説明

症状	考えられる解決方法
メッセージ・ログ内で、ファイル PalmOS.h を検出できませんでした。	SDK が正しくインストールされていないか、誤ったレベルであるかのいずれかです。SDK のレベルは 4.0 でなければなりません。

表 54. Palm ビルド・エラー (続き): 説明

症状	考えられる解決方法
<p>メッセージ・ログに、「MAB.rcp」と、それに続けてエラー条件が出力されています。</p>	<ul style="list-style-type: none"> <li>• バックレベルのリソース・コンパイラを使用している可能性があります。リソース・コンパイラのバージョンを判別するには:             <ol style="list-style-type: none"> <li>1. DOS または Windows NT の「コマンド・プロンプト」をオープンします。</li> <li>2. pilrc と入力します。</li> <li>3. version 2.4 というように表示されます。</li> </ol> </li> <li>• GNU PRC-Tools をインストールするときに、ステップをスキップしてしまった可能性もあります。「セットアップおよび構成」で、GNU PRC-Tools のインストール情報を参照してください。</li> <li>• 問題が解決できない場合は、Mobile Application Builder ファイル (*.mab) と、その解決できないエラーに関するメッセージ・ログを保管しておいてください。</li> </ul>
<p>関数での「未定義」エラーが、メッセージ・ログに出力されました。</p>	<ul style="list-style-type: none"> <li>• エレメントの名前が変更されたにもかかわらず、スクリプト内では更新されていない可能性があります。スクリプト内で、今だに古いエレメント名が参照されています。</li> <li>• 「部品の貼り付け」ウィンドウを使用して、テキストの設定関数またはテキストの取得関数をスクリプトに追加する必要があります。</li> </ul>
<p>「エラー : FORM、MENU、ALERT、VERSION、STRING、CATEGORIES、APPLICATIONICONNAME、APPLICATION、BITMAP、SMALLICON、ICON、TRAP、FONT、または TRANSLATION を予期していました ... 作成: *** [bin.stamp] エラー 1」が、メッセージ・ログに出力されました。</p>	<p>PilRC V2.8 が正しくインストールされていません。</p>
<p>「MAB.rcp(27): エラー: ビットマップがモノクロームではありません」が、メッセージ・ログに出力されました。</p>	<p>Palm の 4.0 より前のバージョンでは、白黒またはモノクロのイメージのみをサポートしています。使用しているイメージがこの形式で保管されていることを確認してください。</p>
<p>メッセージ・ログ内で、ビットマップ・ファイルを検出できませんでした。ファイルへのパスを、スラッシュ抜きで表示します。</p>	<p>ビットマップは、アプリケーションのローカル・プロジェクト・ディレクトリーに保管してください。ファイルへのパスを入力せずに、ファイル名のみを入力してください。</p>

表 54. Palm ビルド・エラー (続き): 説明

症状	考えられる解決方法
「pabappic.bmp を見つけることができません」がメッセージ・ログに表示されました。	発生する可能性の低いエラーですが、8.1 より前のバージョンで作成したアプリケーションをバージョン 8.1 にマイグレーションした場合に発生することがあります。既存のアプリケーションで pabappic.bmp を参照している場合は、MAB_Icon.bmp に変更する必要があります。Mobile Application Builder マイグレーション・ユーティリティーは、このエラーを訂正するように設計されています。
「構造体でも共用体でもない部分でメンバー 'Column1' に対する要求」がメッセージ・ログに表示されました。	このエラーは、Mobile Application Builder の内部データ構造にアクセスしようとした場合に発生することがあります。Mobile Application Builder v8.1 では、すべてのバッファ変数定義が、表構造タイプの静的変数からタイプ表構造のポインタに変更されています。この問題を修正するには、構造のデータ・メンバーへのアクセスの仕方を変更します。つまり、「.」を参照「->」に変更します。
一般のビルド・エラー	問題が解決できない場合は、Mobile Application Builder ファイル (*.mab) と、その解決できないエラーに関するメッセージ・ログを保管しておいてください。

## Palm アプリケーション・エラー

表 55. Palm アプリケーション・エラー

症状	考えられる解決方法
複数行データが Windows マシンで入力され、Palm に同期化され、Palm の複数行フィールドに表示されると、ジャンク文字 [] がデバイスの画面に表示されます。	<p>この問題は、改行文字に使用されるアプローチが Windows と Windows 以外の環境で異なるために発生します。改行のために、Windows では復帰 (CR) と改行 (LF) が使用され、Windows 以外のプラットフォームでは LF のみが使用されます。Palm OS では CR 文字が認識されないため、[] 文字が表示されます。</p> <p><b>修正:</b> アプリケーションにこの文字が表示されている場合は、この文字を除去するフラグをコード生成時に設定できます。「プロジェクト」ペインの「グローバル定義」を選択し、次に「プロパティおよびイベント」ペインの「プリプロセッサ・ステートメント」の横にある</p> <div style="border: 1px solid black; width: 20px; height: 15px; display: flex; align-items: center; justify-content: center; margin: 5px 0;">...</div> <p>をクリックしてプリプロセッサ・ステートメントを入力できるウィンドウを開きます。このウィンドウで #define TRIMCR true と入力します。このウィンドウでは、大文字小文字が区別されるため、上記のとおりに入力してください。</p> <p><b>制限事項:</b> これは一方向のみのプロセスです。つまり、/r 文字について検査されるのは入力データのみです。表に保管されているデータでは、Palm 改行文字 /n が使用されます。</p>

## Symbian Crystal ビルド・エラー

ビルドが正常に終了しなかったことを示す、Mobile Application Builder のエ

ラー・メッセージが出力されます。メッセージ・ログを表示してください。

表 56. Symbian Crystal ビルド・エラー

症状	考えられる解決方法
アプリケーションがビルドされません。	別のバージョンの Perl がインストールされている場合、PATH 環境変数で、他の Perl バージョンが Symbian SDK の Perl より前に指定されていると、アプリケーション・ビルド・プロセスは失敗します。このような場合は、PATH を適切に設定する必要があります。「スタート」->「コントロール パネル」->「システム」->「拡張」->「環境変数」->「システム変数」と進みます。「PATH」を選択して「編集」をクリックします。C:%Perl%bin が他のどの Perl バージョンよりも前に現れていることを確認してください。
一般のビルド・エラー	問題が解決できない場合は、Mobile Application Builder ファイル (*.mab) と、その解決できないエラーに関するメッセージ・ログを保管しておいてください。

### Symbian UIQ ビルド・エラー

ビルドが正常に終了しなかったことを示す、Mobile Application Builder のエラー・メッセージが出力されます。メッセージ・ログを表示してください。

表 57. Symbian Crystal ビルド・エラー：説明

症状	考えられる解決方法
メッセージ・ログのエラーに次のように表示されます: 「内部エラー - cpp.exe がエラー・コード 33 を戻しました。」	aiftool.rh ファイルを Symbian\UIQ_70\epoc32\include ディレクトリに移動します。このファイルは com.symbian.api.GT-shared_0_0_70.sdkpkg IIRC から抽出できます。

### Palm アプリケーションをテストできない

表 58. Palm テスト・エラー

症状	考えられる解決方法
Mobile Application Builder インターフェースから Palm エミュレーターをオープンできません。	Palm OS エミュレーターをインストールするときに、ステップをスキップしてしまった可能性があります。「セットアップと構成」で、エミュレーターのインストール情報を参照してください。
2 つの異なるアプリケーションを、同時に動かすことはできません。	アプリケーションのための Palm アプリケーション ID が、同じである可能性があります。詳しくは、50 ページの『アプリケーション ID』を参照してください。
アプリケーションを実行したときに、表内にデータが何もありませんでした。	開発中に使用した表は、実際にはデバイス上にない可能性があります。モバイル・デバイス上の表へのロードについて詳しくは、「IBM DB2 Everyplace 同期サーバー 管理ガイド」を参照してください。

表 58. Palm テスト・エラー (続き)

症状	考えられる解決方法
アプリケーションをオープンし、使用開始することは可能ですが、エラー・メッセージ (SQLSTATE=####) が出力されます。	SQL エラー・コードの情報については、DB2 Everyplace エンジンと一緒にインストールされている「 <i>IBM DB2 Everyplace アプリケーション開発ガイド</i> 」を参照してください。
Palm のエミュレーターまたはデバイスが、予期したとおりに機能しません。	詳しくは、ご使用のデバイスに同梱されていた資料を参照してください。

### その他の一般エラー

表 59. 一般エラー

症状	考えられる解決方法
プロジェクトでは、表の変更は使用できません。	表定義ファイルが最初にプロジェクトにインポートされた後にその表定義ファイルが変更された場合、その変更内容は表示されません。変更された表定義ファイルをプロジェクトから削除して、再度インポートしてください。詳しくは、『表およびデータ・ソース』を参照してください。
「新規」プロジェクト・ウィンドウまたは「名前変更」プロジェクト・ウィンドウに、「OK」ボタンがありません。	プロジェクト名、プロジェクト・ディレクトリー、またはアプリケーション名の命名制限に違反していないことを確認してください。詳しくは、41 ページの『プロジェクトとアプリケーションの命名規則』を参照してください。

### 関連タスク:

15 ページの『MAB バージョン 8.1.4 のセットアップおよび構成』

108 ページの『Mobile Application Builder アプリケーションのビルド』

113 ページの『Mobile Application Builder アプリケーションのテスト』

## 他の情報ソースへのリンク

以下は、検討時に役立つ他の情報ソースへのリンクです。

- DB2 Everyplace 情報
  - DB2 Everyplace Web サイト
  - DB2 Everyplace フォーラム
  - DB2 Everyplace ライブラリー
  - DB2 Everyplace サポート
- Palm OS 情報
  - Palm OS 資料
  - Palm OS Creator ID データベース

- サポート・ソフトウェア
  - Palm OS エミュレーター (POSE)
  - Palm OS SDK 4.0 for Windows
  - PalmPrint FAQ (Steven's Creek Software)
- バーコード・スキャン
  - Symbol Developer ゾーン
  - Symbol Development SDK ダウンロード・サイト
- Java ツール/情報
  - IBM developerWorks Java テクノロジー・ゾーン
  - java.sun.com
  - Insignia
- Symbian 情報
  - Symbian UID 情報
  - Forum Nokia
  - Sony Ericsson Mobility World

## 第 3 章 例

本節では、Mobile Application Builder の例を提供しています。

### DB2 Everyplace のサンプル・アプリケーション

DB2 Everyplace には、下の表で説明されているように、各ターゲット・プラットフォーム用にサンプル・アプリケーションが用意されています。

表 60. ターゲット・プラットフォーム別 DB2 Everyplace サンプル・アプリケーション

ターゲット・プラットフォーム	タイプ	サンプル・アプリケーション
Palm	クライアント	<ul style="list-style-type: none"><li>• DB2eCLP</li><li>• VNurse</li><li>• NurseInit</li><li>• DB2eAppl</li><li>• DB2eJavaCLP</li><li>• Sync GUI</li></ul>
Palm	MAB	<ul style="list-style-type: none"><li>• PersonList</li><li>• VNApp</li><li>• VNPlus</li></ul>
Symbian 6	クライアント	<ul style="list-style-type: none"><li>• DB2eCLP</li><li>• VNurse</li><li>• NurseInit</li><li>• DB2eAppl</li><li>• DB2eJavaCLP</li><li>• Sync GUI</li><li>• Common</li></ul>
Symbian 6	MAB	<ul style="list-style-type: none"><li>• PersonList</li><li>• VNApp</li></ul>
Symbian 7	クライアント	<ul style="list-style-type: none"><li>• DB2eCLP</li><li>• NurseInit</li><li>• Sync GUI</li><li>• PersonList</li></ul>
Symbian 7	MAB	<ul style="list-style-type: none"><li>• PersonList</li><li>• VNApp</li></ul>

表 60. ターゲット・プラットフォーム別 DB2 Everyplace サンプル・アプリケーション (続き)

ターゲット・プラットフォーム	タイプ	サンプル・アプリケーション
WinCE	クライアント	<ul style="list-style-type: none"> <li>DB2eCLP</li> <li>VNurse</li> <li>VNurseInit</li> <li>DB2eAppl</li> <li>DB2eJavaCLP</li> <li>Sync GUI</li> </ul>
WinCE	MAB	<ul style="list-style-type: none"> <li>PersonList</li> <li>VNApp</li> </ul>
Win32	クライアント	<ul style="list-style-type: none"> <li>DB2eCLP</li> <li>DB2eAppl</li> <li>DB2eJavaCLP</li> </ul>
MIDP	クライアント	<ul style="list-style-type: none"> <li>DB2eCLP</li> <li>Dump.java</li> </ul>
EmbeddedLinux および Neutrino (Sharp Zaurus を含む)	クライアント	<ul style="list-style-type: none"> <li>コマンド行 CLP</li> <li>NurseInit</li> <li>Sync GUI</li> <li>PersonList</li> </ul>
EmbeddedLinux および Neutrino (Sharp Zaurus を含む)	MAB	<ul style="list-style-type: none"> <li>PersonList</li> <li>VNApp</li> </ul>

表 61. バージョンおよびロケーション別 DB2 Everyplace サンプル・アプリケーション

アプリケーション	説明	バージョン	ディレクトリー
<b>DB2eCLP</b>	ユーティリティ・ファイル	EE/DE/SDK	Client\platform\database\lang\DB2eCLP
<b>VNurse</b>	稼働中サンプル、NurseInit と Nurse の組み合わせ	EE/DE/SDK	Client\platform\database\lang\Samples\VNurse
<b>Sync GUI</b> <ul style="list-style-type: none"> <li>testicync.exe</li> <li>isyncui.prc</li> <li>isyncui.exe</li> <li>isync.sis</li> <li>goISync</li> </ul>	稼働中同期サンプル	EE/SDK	Client\platform\sync\lang
<b>ClientAPISample</b> <ul style="list-style-type: none"> <li>ISyncSample</li> <li>GoISyncConsole</li> </ul>	同期 Java サンプル	SDK	Clients\clientapisample\Java_API
<b>NurseInit</b>	表の作成およびデータ取り込み	SDK	SDK\Samples\platform\language\NurseInit
<b>Common</b> <ul style="list-style-type: none"> <li>DB2eEng.cpp</li> <li>DB2eEng.h</li> </ul>	Symbian 6 のコード・サンプル	SDK	SDK\Samples\Symbian6\language\Common
<b>JDBC Sample</b> <ul style="list-style-type: none"> <li>DB2eAppl</li> <li>DB2eJavaCLP</li> </ul>	JDBC サンプル・アプリケーション	SDK	Clients\platform\database\JDBC\lang

表 61. バージョンおよびロケーション別 DB2 Everyplace サンプル・アプリケーション (続き)

アプリケーション	説明	バージョン	ディレクトリー
MIDP Dump.java	MIDP サンプル・アプリケーション  com.ibm.mobileserves.demo	SDK	Clients\MIDP\lang\Samples
PersonList	サンプル MAB アプリケーション	SDK	SDK\ApplicationBuilder\Projects\Samples\platform
VNApp	サンプル MAB アプリケーション	SDK	SDK\ApplicationBuilder\Projects\Samples\platform\
VNPlus	サンプル MAB アプリケーション	SDK	SDK\ApplicationBuilder\Projects\Samples\platform\

## CREATE TABLE ステートメントの例

下記に、CREATE TABLE ステートメントの例を示します。これは、VNPerson.ddl (巡回個人情報) で使用されている CREATE TABLE ステートメントです。

表 62. CREATE TABLE ステートメントの例

```
CREATE TABLE 巡回個人情報
(ID Char(9) PRIMARY KEY,
名前 Varchar(40),
住所 Varchar(50),
自宅電話 Varchar(20),
勤務先電話 Varchar(20),
携帯電話 Varchar(20))
```

### 関連タスク:

52 ページの『表およびデータ・ソース』

## データベース照会の定義

アプリケーション・ビルダーのプロジェクトは、各アプリケーション内に複数の表を持つことができます。これらの表を結合して、最大 2 つの表からの情報を一度に表示することができます。そのためには、各表の構造と内容について知っておくことが重要です。

巡回医療サービスのサンプル・アプリケーションは、Mobile Application Builder で可能な表結合機能およびデータベース照会の良い例をいくつか示しています。

### データベース照会タスク例:

180 ページの『巡回医療サービス・サンプル・アプリケーションのオープン』

180 ページの『ユーザー・アプリケーションでデータベース照会を定義するための

ガイドライン』

- 181 ページの『サンプル照会 — すべてのレコードの選択』
- 182 ページの『サンプル照会 — 特定のレコードの選択』
- 182 ページの『サンプル照会 — 2 つの表の結合』
- 183 ページの『サンプル照会 — 列の値が一致するレコードの選択』

関連タスク:

52 ページの『表およびデータ・ソース』

## 巡回医療サービス・サンプル・アプリケーションのオープン

巡回医療サービスのサンプル・アプリケーションは、Mobile Application Builder で可能な表結合機能およびデータベース照会の良い例をいくつか示しています。

---

巡回医療サービス・アプリケーションをオープンするには:

1. 「ファイル」 → 「プロジェクトのオープン」をクリックします。「プロジェクトのオープン」ウィンドウが開きます。
2. Application Builder のインストール・ディレクトリーにナビゲートし、  
...¥Projects¥Samples¥<target OS>¥VNAApp ディレクトリー内の VNAApp.pab ファイルを選択します。
3. 「オープン」をクリックします。

---

関連タスク:

52 ページの『表およびデータ・ソース』

## ユーザー・アプリケーションでデータベース照会を定義するためのガイドライン

フォーム、フィールド、およびリストに指定したプロパティから、ユーザー・アプリケーションのためのデータベース照会が生成されます。データ・ソース・プロパティおよびデータ・フィールド・プロパティによって、照会で使用するデータベースの表と列が定義されます。また、リンク・ソース・プロパティおよびリンク・フィールド・プロパティが指定されている場合は、それらによって、SQL SELECT ステートメント内でどのように表をパラメーター化し、結合するかが定義されます。

ガイドラインとしては、1 つのフォームに関連付ける照会を 1 つだけにすることで、フォームをシンプルに保つことをお勧めします。複数の結合照会を同じフォームで定義すると、期待通りの結果が得られなくなります。このような状態は、フォーム上のリストとフィールドの両方に対して「リンク・ソース」および「リンク・フィールド」プロパティを指定した場合に起こり得ます。

#### 関連タスク:

52 ページの『表およびデータ・ソース』

## 複数のパラメーターを使用する照会の生成

フォームおよびリスト制御は「選択基準」プロパティを使用します。このプロパティを使用して、照会の WHERE 文節に複数のパラメーターが含まれる SELECT 照会を生成できます。プロパティ設定によって、ユーザーは、パラメーター・マーカーを持つ WHERE 文節内に任意数の列を入れ、実行時にホスト変数を使用してパラメーターの値をバインドすることができます。一般にホスト変数の値は、パラメーターのバインドに使用される前に、「ホスト変数の更新」プロパティの設定値で設定されます。

---

#### 関連タスク:

97 ページの『ホスト変数の使用』

100 ページの『リスト制御または選択リスト制御の選択基準の設定』

## サンプル照会 — すべてのレコードの選択

このサンプル照会は、表からすべてのレコードを選択する方法を示します。

この照会は、クイック・ツアー (Palm アプリケーションのオープン、ビルド、およびテスト) でデモンストレーションされる PersonList サンプルで使用されているものです。この照会では、巡回個人情報表で各レコードを検索します。データベースの表と列をフォーム・フィールドに関連付けることによって、下記のサンプル照会が生成されます。

サンプル SQL:

```
SELECT ID, 名前, 住所, 自宅電話, 勤務先電話, 携帯電話 FROM 巡回個人情報
```

表 63. PersonList フィールドと、それに対応するデータ・ソースおよびデータ・フィールドの各プロパティ

フィールド・ラベル	データ・ソース	データ・フィールド
ID	巡回個人情報	ID
名前	巡回個人情報	名前
住所	巡回個人情報	住所
市区町村	巡回個人情報	市区町村
電話(自宅)	巡回個人情報	電話(自宅)
電話(職場)	巡回個人情報	電話(職場)
携帯電話	巡回個人情報	携帯電話

---

#### 関連タスク:

52 ページの『表およびデータ・ソース』  
95 ページの『フォーム・イベントの SQL を変更』

## サンプル照会 — 特定のレコードの選択

このサンプル照会は、列値がパラメーター値と等しい特定のレコードを表から選択する方法を示します。

この照会は、巡回医療サービス・サンプルの患者情報フォームで使用されているものです。この照会では、巡回個人情報表から、表の ID がパラメーターの ID の現行値と等しいレコードを選択します。巡回医療サービス・アプリケーションでは、ID の現行値が、予定フォームで選択された患者の ID と等しくなります。

データベースの表と列をフォーム・フィールドに関連付けることによって、下記のサンプル照会が生成されます。さらに、「フォーム」、「データ・ソース」、および「データ・フィールド」プロパティに値を指定することで、現行 ID による巡回個人情報レコードの照会を限定する検索条件も組み込まれます。

サンプル SQL:

```
SELECT ID, 名前, 住所, 自宅電話, 勤務先電話, 携帯電話 FROM 巡回個人情報  
WHERE ID=?
```

表 64. 患者情報のフォームおよびフィールドと、それに対応する「データ・ソース」および「データ・フィールド」プロパティ

フォーム/フィールド・ラベル	データ・ソース	データ・フィールド
患者情報フォーム	巡回個人情報	ID
名前	巡回個人情報	名前
住所	巡回個人情報	住所
市区町村	巡回個人情報	市区町村
電話(自宅)	巡回個人情報	電話(自宅)
電話(職場)	巡回個人情報	電話(職場)
携帯電話	巡回個人情報	携帯電話

### 関連タスク:

52 ページの『表およびデータ・ソース』  
95 ページの『フォーム・イベントの SQL を変更』

## サンプル照会 — 2 つの表の結合

このサンプル照会は、2 つの表を結合する方法を示します。

この照会は、巡回医療サービス・サンプルの最初のフォームである予定フォームで使用されているものです。この照会では、巡回予定表と巡回個人情報表から、巡回予定表の患者ID と巡回個人情報表の ID が等しいレコードを結合します。この検索条件は、巡回予定.患者ID の列にリンク・ソース値およびリンク・フィールド値を定義することで指定されます。

**注:** これら 2 つの表を正しく結合するためには、患者の固有 ID が入っている巡回予定.患者ID の列と巡回個人情報.ID の列を指定する必要があります。しかし、この ID をリストに表示したくないので、使用可能属性を UNUSABLE に設定しておきます。

照会ではデフォルトによって、アプリケーション内で定義された列だけではなく、表に定義されているすべての列が取り出されます。

サンプル SQL:

```
SELECT 巡回予定.患者ID, 巡回予定.時間, 巡回個人情報.ID, 巡回個人情報.名前,
巡回個人情報.住所, 巡回個人情報.自宅電話, 巡回個人情報.勤務先電話, 巡回個人情報.携帯電話
FROM 巡回予定, 巡回個人情報 WHERE 巡回予定.患者ID=巡回個人情報.ID
```

表 65. 予定フォーム・リスト制御と、「データ・ソース」、「データ・フィールド」、「リンク・ソース」、「リンク・フィールド」の各プロパティ

リスト 1 フォーム 1	データ・ソース	データ・フィールド	リンク・ソース	リンク・フィールド
列 1	巡回予定	時間		
列 2	巡回個人情報	名前		
列 3	巡回予定	患者ID	巡回個人情報	ID

**関連タスク:**

52 ページの『表およびデータ・ソース』

95 ページの『フォーム・イベントの SQL を変更』

## サンプル照会 — 列の値が一致するレコードの選択

このサンプル照会は、ある表の列値が別の表の列値と等しいレコードを表から選択する方法を示します。

この照会は、巡回医療サービス・サンプルの連絡先情報フォームで使用されているものです。この照会では、巡回個人情報表から、この表の ID が巡回連絡先.連絡先 ID の現行値と等しいレコードを選択します。巡回医療サービス・アプリケーションでは、連絡先 ID の現行値が、緊急連絡先フォームで選択された連絡先の連絡先 ID と等しくなります。

この照会は前に挙げた例と同じですが、異なるパラメーター値で実行されます。前の照会では、巡回個人情報.ID パラメーターが現行値と等しいものでした。この照会では、巡回個人情報.ID パラメーターが現行の巡回連絡先.連絡先 ID の値と等しいものです。このパラメーター指定は、フォームのデータ・ソース、データ・フィールド、リンク・ソース、およびリンク・フィールドの、各プロパティによって指定されます。

サンプル SQL:

```
SELECT ID, 名前, 住所, 自宅電話, 勤務先電話, 携帯電話 FROM 巡回個人情報
WHERE ID=?
```

表 66. 連絡先情報のフォームおよびフィールドと、それに対応するデータ・ソース、データ・フィールド、リンク・ソース、リンク・フィールドの各プロパティ

フォーム/フィールド・ラベル	データ・ソース	データ・フィールド	リンク・ソース	リンク・フィールド
連絡先情報フォーム	巡回個人情報	ID		
連絡先情報フォーム			巡回連絡先	連絡先 ID
名前	巡回個人情報	名前		
住所	巡回個人情報	住所		
市区町村	巡回個人情報	市区町村		
電話(自宅)	巡回個人情報	電話(自宅)		
電話(職場)	巡回個人情報	電話(職場)		
携帯電話	巡回個人情報	携帯電話		

#### 関連タスク:

52 ページの『表およびデータ・ソース』

95 ページの『フォーム・イベントの SQL を変更』

## スクリプト内での DB2 CLI 関数の使用

このセクションのコードは以下の表定義を前提とします:

```
CREATE TABLE blob (id INT, data BLOB(8192))
```

フォームまたは制御に関連したスクリプトの表列の値を挿入しリトリブするよう要求される場合があります。下記は、DB2 CLI 関数を使用して BLOB データ・タイプに対してどのようにこれを行うことができるかを示す 2 つのルーチンです。BLOB へのポインターを一度リトリブすると、Palm API WinDrawBitmap を使用して BLOB 値を表示することができます。

```
// *****
// データを BLOB データ・タイプに挿入する機能
// *****
SQLRETURN DB2e_insert_blob(Word id, Word size, char *blob)
{
    // SQL parameters                                1 2 3
    static SQLCHAR SQL_insert[] = "insert into blobs values (?,?,:)";
    SQLINTEGER     SQL_size     = size;
```

```

SQLRETURN      rc;

// 挿入を準備します。
if (rc = SQLPrepare(stmt, SQL_insert, SQL_NTS))
    return myerror("SQLPrepare (insert)", rc);

// blob id をバインドします。
if (rc = SQLBindParameter(stmt, 1,
    SQL_PARAM_INPUT, SQL_C_SHORT, SQL_INTEGER, 0, 0,
    &id, 0, NULL))
    return myerror("SQLBindParameter (insert/1)", rc);

// blob サイズをバインドします。
if (rc = SQLBindParameter(stmt, 2,
    SQL_PARAM_INPUT, SQL_C_SHORT, SQL_INTEGER, 0, 0,
    &size, 0, NULL))
    return myerror("SQLBindParameter (insert/2)", rc);

// blob データをバインドします。
if (rc = SQLBindParameter(stmt, 3,
    SQL_PARAM_INPUT, SQL_C_BINARY, SQL_BLOB, size, 0,
    blob, size, &SQL_size))
    return myerror("SQLBindParameter (insert/3)", rc);

// ここで挿入を行います。
if (rc = SQLExecute(stmt))
    return myerror("SQLExecute (insert)", rc);

return SQL_SUCCESS;
}

// *****
// BLOB へのポインターを取得する機能
// *****
char *DB2e_get_blob(Word id)
{
    // SQL パラメーター 1
    static SQLCHAR SQL_select[] = "select size,
blob from blobs where id = ?";

    Word      size;
    char      *blob;
    SQLINTEGER SQL_size;

    // 選択を準備します。
    if (SQLPrepare(stmt, SQL_select, SQL_NTS))
        return myerror("SQLPrepare (select)", (char *)NULL;

```

```

// blob id をバインドします。
if (SQLBindParameter(stmt, 1,
    SQL_PARAM_INPUT, SQL_C_SHORT, SQL_INTEGER, 0, 0,
    &id, 0, NULL))
    return myerror("SQLBindParameter (select/1)",
(char *)NULL;

// ここで選択を行ないます。
if (SQLExecute(stmt))
    return myerror("SQLExecute (select)", (char *)NULL;

// blob の取り出しを試行します。
switch (SQLFetch(stmt)) {
case SQL_SUCCESS:
    break;
case SQL_NO_DATA_FOUND:
    return (char *)NULL;
default:
    return myerror("SQLFetch"), (char *)NULL;
}

// サイズ列を読み取ります。
if (SQLGetData(stmt, 1,
    SQL_C_SHORT,
    &size, 0,
    NULL))
    return myerror("SQLGetData (size)", (char *)NULL;

if (!(blob = MemPtrNew(size)))
    return (char *)NULL;

// blob 列を読み取ります。
if (SQLGetData(stmt, 2,
    SQL_C_BINARY,
    blob, size,
    &SQL_size))
    return myerror("SQLGetData (data)", (char *)NULL;

if (size != SQL_size)
    return ErrDisplay("DB2e: Retrieved corrupted blob"),
(char *)NULL;

return blob;
}

```

---

関連概念:

157 ページの『スクリプト記述サポート』

**関連タスク:**

87 ページの『Mobile Application Builder 中のスクリプト』

---

## 変数のメモリの解放

変数にメモリーを割り振って、メモリーを解放しなかった場合、Palm Emulator v3.5以降を使用しているときにメモリー・リークが発生する可能性があります。MAB生成関数の中には、メモリーを割り振り、割り振られたメモリーにポインターを戻すものがあります。スクリプトでこれらの関数を使用する場合は、メモリー・リークを避けるために、割り振られたメモリーを明示的に解放する必要があります。

下の例は、割り振られたメモリーを解放する方法を示しています。

---

```
CharPtr commentTxt;
commentTxt = PAB_getFieldText(Field8Form7);
.
.
.
.
// commentTxt に割り振られたメモリーを解放します
if (commentTxt)
    MemPtrFree(commentTxt) ;
```

---

**関連概念:**

157 ページの『スクリプト記述サポート』

**関連タスク:**

87 ページの『Mobile Application Builder 中のスクリプト』



# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

## [ア行]

アイコン  
作成 77  
制御 127  
プロパティ 152

アクション  
オプション 86  
概要 82  
記述 155  
構成 83  
ターゲット 83

アクセシビリティ 12

アプリケーション  
テスト 113  
ビルド, Palm 109  
命名規則 41

アプリケーション ID  
概要 50  
登録 51  
変更 51

アプリケーション, Palm  
エミュレーター上でのテスト 116  
モバイル・デバイス上でテスト 118  
Palm OS 5 Simulator でのテスト 117

アプリケーション, サンプル 177

アプリケーション, Sharp Zaurus  
デスクトップ上でのシミュレート 125  
モバイル・デバイス上でテスト 126

アプリケーション, Symbian Crystal  
エミュレーター上でのテスト 119  
モバイル・デバイス上でテスト 120

アプリケーション, Symbian UIQ  
エミュレーター上でのテスト 121  
デバイス上でのテスト 122

アプリケーション, WinCE  
デスクトップ上でのシミュレート 123  
モバイル・デバイス上でテスト 124

アプリケーション, Zaurus  
デスクトップ上でのシミュレート 125  
モバイル・デバイス上でテスト 126

アプリケーションのビルド  
概要 108

アプリケーション・プリファレンスの設定  
48

アラート  
作成 72  
プロパティ 149

アンインストール, Mobile Application Builder の 38

暗号化サポート 85

位置合わせ  
制御 61

位置ダイアログ, 表示 85

一般プリファレンスの設定 46

イベント  
位置ダイアログ, 表示 85  
概要 82  
記述 153  
制御の構成 83  
制御の再配列 84  
データベース接続ダイアログ, 表示 85  
データベース・プリファレンス・ダイアログ, 表示 85

イベント・スクリプト  
削除 92  
作成 91  
名前変更 93  
編集 92

色, 変更 66

インク制御, プロパティ 139

印刷  
概要 101  
サポート 158  
API 呼び出し 159

印刷プリファレンスの設定 49

印刷ボタン, フォームへの追加 101

印刷メニュー項目, フォームへの追加 102

エラー, 診断 170

## [カ行]

確認ダイアログのためのプリファレンスの設定 48

キーボード 12

行順序  
変更 56

繰り返しボタン制御, プロパティ 144

グローバル変数, 定義 90

グローバル・スクリプト  
削除 91  
作成 89  
編集 91

結果列, 選択 54

コード  
Java サポート 157  
Palm C コード・サポート 158

コード生成の記述 110

コマンド・ボタン配列制御, プロパティ 135

固有 ID, 生成, フィールド制御の 71

## [サ行]

削除  
制御 61

削除, 物理 86

サンプル照会  
すべてのレコードの選択 181  
特定のレコードの選択 182  
列の値が一致するレコードの選択 183  
2つの表の結合 182

サンプル・アプリケーション  
巡回医療サービス 180

サンプル・アプリケーション, 説明と場所 177

「順序づけ」ウィンドウ, オープン 70

ショートカット・キー 12

照会  
複数のパラメーターで生成 181

初期フォーム  
設定 58

「新規プロジェクト」ウィンドウ, オープン 39

身体障害 12

ズーム率, 変更, フォームの 58

スクリプト  
概説 87  
ヘルプ機能の追加 94

Java 87

Java サポート 157

Palm C コード・サポート 158

スクリプト・プリファレンスの設定 48

スクロール・バー制御, プロパティ 146

ストリング  
作成 81  
プロパティ 150

制御  
アイコンの記述 127  
位置合わせ 61  
移動 62  
色の変更 66  
概要 60  
構成, イベントの 83

制御 (続き)  
サイズ変更 63  
再配列 63  
削除 61  
スクリプトへの貼り付け 95  
デバイスの構成 64  
フォームへの追加 60  
フォントの変更 65  
プロパティ 128  
プロパティの構成 64  
制御パレット 60  
制御パレット、記述 12  
選択トリガー制御、プロパティ 147  
選択リスト制御、プロパティ 134  
前提条件ツール・プリファレンスの設定 50  
ソース・ファイル、プロジェクトへの追加 43

## [タ行]

ターゲット 83  
ターゲット、概要 82  
チェック・ボックス制御、プロパティ 133  
ツールバー 7  
データベース照会  
定義 179  
定義のためのガイドライン 180  
複数のパラメーターで生成 181  
データベース接続ダイアログ、表示 85  
データベース・プリファレンス・ダイアログ、表示 85  
データ・ソース  
概要 52  
定義、リスト制御 54  
テスト、アプリケーションの、概説 113  
同期機能、アプリケーションへの追加 107  
同期プリファレンス・ダイアログの追加 107  
トラブルシューティング 170

## [ハ行]

バーコード・スキャン  
概要 103  
ソフト・スキャン機能の追加 104  
ハード・スキャン機能の追加 106  
API 呼び出し 162  
Palm アプリケーションでサポートされるバーコード・タイプ 169  
WinCE でサポートされるバーコード・タイプ 170  
パラメーター、複数 181

汎用 Java アプリケーション 40  
ビットマップ  
概要 75  
新規作成 75  
制御上での表示 77  
フォーム上での表示 76  
プロパティ 153  
表  
エミュレーターまたはモバイル・デバイス上での作成および移植 114  
結合 179  
表、結合 182  
表定義ファイル  
インポート 52  
概要 52  
更新 53  
削除 54  
作成 52  
ビルド・エラー、診断 170  
ビルド・ファイルの記述 110  
ビルド・プリファレンスの設定 47  
フィールド制御  
固有 ID の生成 71  
プロパティ 136  
フォーム  
ウィザードを使用した作成 57  
概要 56  
サイズ変更 59  
削除 59  
手動での作成 57  
ズーム率の変更 58  
設定、初期 58  
プロパティ 148  
SQL の削除 96  
SQL の変更 95  
「フォーム」ペイン、記述 11  
フォームのサイズ変更 59  
フォーム・サイズ  
デフォルトの設定、汎用 Java アプリケーションの 40  
フォーム・ビットマップ制御、プロパティ 129  
フォント、変更 65  
複数コード・セグメント 47  
プッシュボタン制御、プロパティ 143  
物理削除 86  
ブラウザ・プリファレンスの設定 46  
プリファレンス  
アプリケーションの設定 48  
一般の設定 46  
印刷の設定 49  
ウェルカム・ダイアログの設定 46  
概説 46  
確認ダイアログの設定 48  
スクリプトの設定 48  
前提条件ツールのための設定 50

プリファレンス (続き)  
デバッグの設定 47  
ビルドの設定 47  
ブラウザの設定 46  
UI カラーの設定 48  
プリプロセッサ・ステートメント、定義 108  
プロジェクト  
概要 38  
既存のオープン 40  
クローズ 45  
新規作成 39  
ソース・ファイルの追加 43  
名前変更 42  
プロパティ 128  
プロパティの構成 43  
保管 45  
命名規則 41  
ライブラリー・ファイルの追加 44  
プロジェクトのクローズ 45  
プロジェクトの保管 45  
プロジェクト・リソース  
概要 72  
プロパティ 148  
プロパティ  
構成、制御 64  
制御の記述 128  
プロジェクトの記述 128  
プロジェクトのための構成 43  
リソースの記述 148  
「プロパティおよびイベント」ペイン、記述 11  
ヘルパー機能  
スクリプトへの追加 94  
制御のための 95  
変数、メモリーの解放 187  
ホスト変数  
概要 97  
定義 97  
フィールド制御のプロパティの設定 98  
フォームのプロパティの設定 99  
リスト制御または選択リスト制御のプロパティの設定 99, 100  
ボタン制御、プロパティ 131  
ポップアップ・トリガー制御  
使用法の例 68  
プロパティ 142

## [マ行]

命名規則 41  
メッセージ・ログ  
表示 109  
保管 110  
メニュー、プロパティの 151

メニュー項目、作成 79  
メニュー・バー  
作成 80  
プロパティ 151  
メモリー、変数についての解放 187  
メモリー・リーク 187

## [ヤ行]

ユーザー ID とパスワード・ダイアログ、表示 85

## [ラ行]

ライブラリー・ファイル、プロジェクトへの追加 44  
ラジオ・ボタン制御  
作成 69  
ラベル制御、プロパティ 140  
リスト制御  
データ・ソースの定義 54  
フィールド順序の定義 70  
リスト制御、プロパティ 140

## [数字]

2 次ストレージ・デバイス、サポート 85  
5 ウェイ・ナビゲーション 67  
5-way ナビゲーション 67

## B

blob 制御、プロパティ 130

## C

C コード  
サポート 158  
CBA 制御、プロパティ 135  
CLI 関数、使用法の例 184  
CLP サンプル・アプリケーション  
Palm エミュレーターへのインストール 19  
Symbian UIQ デバイスへのインストール 30  
CREATE TABLE ステートメント、例 179  
Cygwin ツール、インストールおよび構成 16

## G

Graffiti シフト制御、プロパティ 138

## I

IBM J9 31

## J

Java アプリケーション  
生成されるビルド・ファイル 112  
Java スクリプト  
作成 87

## M

Mobile Application Builder  
アンインストール 38  
UI の概要 1

## N

Nokia SDK 23  
NurseInit サンプル・アプリケーション  
Palm エミュレーターへのインストール 19  
Palm シミュレーターへのインストール 21  
Palm デバイスへのインストール 22  
Sharp Zaurus デバイスへのインストール 37  
Symbian Crystal エミュレーターへのインストール 25  
Symbian Crystal デバイスへのインストール 26  
Symbian UIQ デバイスへのインストール 30  
NurseInit.bat、使用 34

## P

Palm  
ビルド環境のセットアップ 15  
5 ウェイ・ナビゲーション 67  
Palm OS 5 Simulator  
セットアップおよび構成 20  
Palm OS SDK、インストールおよび構成 16  
Palm アプリケーション  
エミュレーター上でのテスト 116  
生成されるビルド・ファイル 111  
モバイル・デバイス上でテスト 118  
Palm OS 5 Simulator でのテスト 117  
Palm エミュレーター  
サンプル・アプリケーションのインストール 19  
セットアップおよび構成 18  
デフォルト設定の変更 20

Palm エミュレーター (続き)  
同期のための構成 19  
Palm シミュレーター  
サンプル・アプリケーションのインストール 21  
同期のための構成 21  
Palm デバイス  
サンプル・アプリケーションのインストール 22  
テストのためのセットアップ 22  
同期のための構成 22  
Palm ビルド・ツール・プリファレンス、設定 17  
PalmPrint、インストール 101  
PersonList サンプル・アプリケーション  
Sharp Zaurus デバイスへのインストール 37  
Symbian UIQ デバイスへのインストール 30

## S

SDK、Palm 16  
Sharp Zaurus  
アプリケーションをビルドするためのセットアップ 35  
Sharp Zaurus アプリケーション  
デスクトップ上でのシミュレート 125  
モバイル・デバイス上でテスト 126  
Sharp Zaurus デスクトップ・シミュレーション、テストのためのセットアップ 33, 36  
Sharp Zaurus デバイス  
NurseInit サンプル・アプリケーションのインストール 37  
PersonList サンプル・アプリケーションのインストール 37  
Sony Ericsson SDK 27  
SQL  
フォーム・イベントのための削除 96  
フォーム・イベントのための変更 95  
Symbian 7.0 SDK 27  
Symbian Crystal  
ビルド環境のセットアップ 23  
Symbian Crystal アプリケーション  
エミュレーター上でのテスト 119  
モバイル・デバイス上でテスト 120  
Symbian Crystal エミュレーター  
起動時間の短縮 25  
サンプル・アプリケーションのインストール 25  
テストのためのセットアップ 24  
同期のための構成 25  
Symbian Crystal デバイス  
サンプル・アプリケーションのインストール 26

- Symbian Crystal デバイス (続き)
  - テストのためのセットアップ 26
  - 同期のための構成 26
- Symbian UIQ
  - ビルド環境のセットアップ 27
- Symbian UIQ アプリケーション
  - エミュレーター上でのテスト 121
  - デバイス上でのテスト 122
- Symbian UIQ デバイス
  - テストのためのセットアップ 29
  - 同期のための構成 30
  - CLP サンプル・アプリケーションのインストール 30
  - NurseInit サンプル・アプリケーションのインストール 30
  - PersonList サンプル・アプリケーションのインストール 30
- Symbian アプリケーション
  - 生成されるビルド・ファイル 112
- Symbian エミュレーター
  - テストのためのセットアップ 28
- Symbian ビルド・プリファレンス、設定 24, 28

## U

- UI カラー・プリファレンス、設定 48

## W

- Web サイト、リンク 175
- WinCE JVM 31
- WinCE JVM プリファレンス、設定 31
- WinCE アプリケーション
  - 生成されるビルド・ファイル 113
  - デスクトップ上でのシミュレート 123
  - ビルドのためのプリファレンスの設定 48
  - モバイル・デバイス上でテスト 124
- WinCE デスクトップ・シミュレーション、テストのためのセットアップ 33
- WinCE デバイス、テストのためのセットアップ 34
- WinCE 表、作成およびデータの追加 34

## Z

- Zaurus アプリケーション
  - デスクトップ上でのシミュレート 125
  - モバイル・デバイス上でテスト 126
- Zaurus デスクトップ・シミュレーション、テストのためのセットアップ 36
- Zaurus、アプリケーションをビルドするためのセットアップ 35

---

## 特記事項

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-0032  
東京都港区六本木 3-2-31  
IBM World Trade Asia Corporation  
Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Canada Limited

Office of the Lab Director  
8200 Warden Avenue  
Markham, Ontario  
L6G 1C7  
CANADA

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

#### 著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物には、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年)。このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. \_西暦年を入力\_. All rights reserved.

---

## 商標

以下は IBM Corporation の商標または登録商標であり、DB2 UDB 資料ライブラリ一内の少なくとも 1 つの資料で使用されています。

ACF/VTAM	LAN Distance
AISPO	MVS
AIX	MVS/ESA
AIXwindows	MVS/XA
AnyNet	Net.Data
APPN	NetView
AS/400	OS/390
BookManager	OS/400
C Set++	PowerPC
C/370	pSeries
CICS	QBIC
Database 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/400
DB2 Extenders	SQL/DS
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	Tivoli
eServer	VisualAge
Extended Services	VM/ESA
FFST	VSE/ESA
First Failure Support Technology	VTAM
IBM	WebExplorer
IMS	WebSphere
IMS/ESA	WIN-OS/2
iSeries	z/OS
	zSeries

以下は IBM Corporation の商標または登録商標であり、DB2 UDB 資料ライブラリ一内の少なくとも 1 つの資料で使用されています。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Intel および Pentium は、Intel Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

UNIX は、The Open Group がライセンスしている米国およびその他の国における登録商標です。

他の会社名、製品名、およびサービス名等はそれぞれ各社の商標です。





Printed in Japan