DB2 Everyplace Mobile Application Builder version 8.1.4

Table of contents

Chapter 2. Concepts/Reference. 109

Chapter 3. Examples 155

Index 165

Notices 169

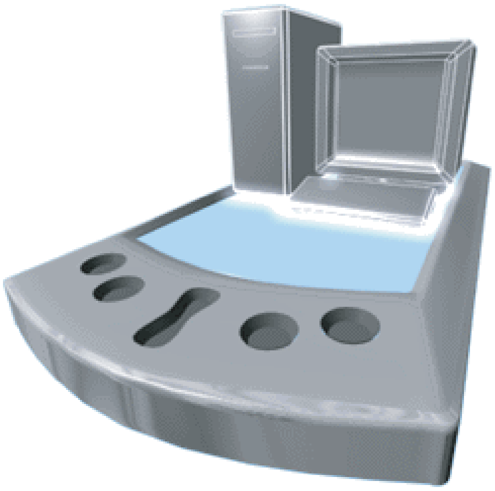
Chapter 1. Tasks

This section provides task information for the Mobile Application Builder.

Getting started with Mobile Application Builder

What is DB2 Everyplace?

DB2 Everyplace is part of IBM's solution for truly mobile computing. With DB2 Everyplace, mobile professionals (such as sales people, inspectors, auditors, field service technicians, doctors, realtors, and insurance claim adjusters) can keep in touch with vital data that they need while away from the office.



Specifically, organizations are now able to deliver their DB2 enterprise data to mobile devices. With DB2 Everyplace, you can access and perform updates to a database on your mobile device and synchronize the data through your desktop with other DB2 databases in your enterprise.

Mobile Application Builder

DB2 Everyplace is a relational database that resides on your mobile device. Applications can be used to access the data on the mobile device. The DB2 Everyplace Mobile Application Builder is a visual authoring tool that makes it easy to produce custom-made mobile applications. Mobile Application Builder puts the power into the hands of the application developers.

Task list:

Fundamental tasks

- Creating a new project
- Setting project preferences
- Building Applications
- Testing Applications

Advanced tasks

- Adding printing capabilities to an application
- Adding barcode scanning capabilities to an application
- Adding synchronization capabilities to an application

- Scripting in Mobile Application Builder

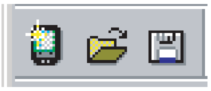
You can do many other tasks with Mobile Application Builder. See the table of contents for this help in the left pane of your browser for a list of tasks.

The Mobile Application Builder interface:

Mobile Application Builder has the following elements:

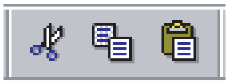
Menu bar
Use the menu bar to work with objects in Mobile Application Builder, and access online help. See "Mobile Application Builder menu bar" on page 3 for more information.

Mobile Application Builder project toolbar



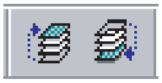
Use the project toolbar icons above the Project pane to create a new project, open an existing project, and save the current project. These functions can also be selected in the **File** menu. See "Project toolbar options" on page 6 for more information.

Mobile Application Builder edit toolbar



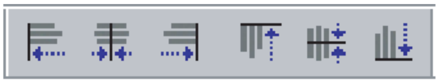
Use the edit toolbar icons above the Project pane to perform standard cut, copy, and paste actions. These functions can also be selected in the **Edit** menu. See "Edit toolbar options" on page 6 for more information.

Mobile Application Builder control order toolbar



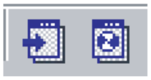
Use the control order icons above the Project pane to move a selected control to the front or the back of the form. These functions can also be selected in the **Format** menu. See "Control order toolbar options" on page 7 for more information.

Mobile Application Builder control alignment toolbar



Use the control alignment toolbar icons above the Form pane to align selected controls. These functions can also be selected in the **Format** menu. See "Control alignment toolbar options" on page 7 for more information.

Mobile Application Builder build toolbar



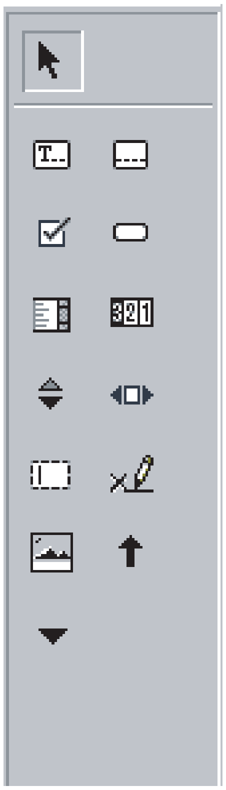
Use the build and test toolbar icons above the Form pane to build and test an application. These functions can also be selected in the **Build** menu. See "Build toolbar" on page 8 for more information.

Project pane
Use the Project pane, on the left side of the Mobile Application Builder interface, to display and work with project resources. See "Project pane options" on page 8 for more information.

Properties and Events pane
Use the Properties and Events pane, at the bottom of the Mobile Application Builder interface, to select and edit the properties and events of selected controls. On the Properties page, determine how a control looks and behaves. On the Events page, determine how a control reacts to a specific action. Each control has different possible actions. See "Properties and Events pane options" on page 9 for more information.

Form pane
The Form pane displays the current form. You can select a control from the palette and click the location on the form to place the control. See "Form pane options" on page 9 for more information.

Control Palette



The Control Palette displays all the form controls available for the specified device. The available controls will vary depending on the target platform. See "Control Palette options" on page 9 for more information.

Mobile Application Builder menu bar

The Mobile Application Builder menu bar contains the following menus:

File Use this menu to work with Mobile Application Builder projects and exit from Mobile Application Builder. Some of the functions in this menu are also available by clicking the icons in the project toolbar. See File menu options for more information.

Edit Use this menu to perform standard cut, copy, and paste functions. Some of the functions in this menu are also available by clicking the icons in the edit toolbar. See Edit menu options for more information.

Selected
Use this menu to display and select the available actions for the object that is selected in the Project pane or Form pane. See Selected menu options for more information.

Format
Use this menu to align and order controls on a form. Some of the functions in this menu are also available by clicking the icons in the control order toolbar and the control alignment toolbar. See Format menu options for more information.

Build Use this menu to build and test an application. The functions in this menu are also available by clicking the icons in the build toolbar. See Build menu options for more information.

Window
Use this menu to switch between open windows.

Help Use this menu to view the log file for the most recent build, view the product documentation, or view information about the version of Mobile Application Builder you are using. See Help menu options for more information.

File menu options

Use the **File** menu to work with Mobile Application Builder projects and exit from Mobile Application Builder. Some of the functions in this menu are also available by clicking the icons in the project toolbar. In addition to the menu items listed below, you can open recent projects, listed by project name, from the **File** menu.

From this menu, you can select these menu items:

New Project
Opens a new Mobile Application Builder project.

Open Project
Opens an existing Mobile Application Builder project.

Close Project
Closes the current Mobile Application Builder project.

Save Project
Saves the current Mobile Application Builder project without closing it.

Save Project As
Saves the current Mobile Application Builder project under a new name.

Preferences
Opens a window where you can set project and application preferences.

Exit Closes Mobile Application Builder.

Selected menu options

Use the **Selected** menu to display and select the available actions for the object selected in the Project pane or the Form pane.

If an object is selected in the Project pane or the Form pane, the **Selected** menu contains the actions for that object.

To select an object, click mouse button 1 on the object.

You can also display the actions for objects by clicking mouse button 2 directly on the objects. A pop-up menu opens containing the list of available actions.

Edit menu options

Use the **Edit** menu to perform general editing commands (Cut, Copy, Paste) on selected objects.

From this menu, you can select these menu items:

Cut Cuts the selected object and places it in the clipboard.

Copy Copies the object in the clipboard to the selected location.

Paste Pastes the object in the clipboard to the selected location.

Delete Deletes the selected object.

Some of these functions are also available from the edit toolbar.

Build menu options

Use this menu to build and test an application.

From this menu, you can select these menu items:

Build Builds the application.

Test Tests the application. You must successfully build an application before you can test it.

These functions are also available from the build toolbar.

Format menu options

Use this menu to align and order controls on a form.

From this menu, you can select these menu items:

Align Aligns the selected controls.

Order Moves the selected control to the front or the back of the form.

These functions are also available from the build toolbar.

Help menu options

Use the Help menu to view the message log, display online help, and display information about DB2.

View Log File
Opens the message log, containing information about each build that was done during the current session.

Documentation
Displays online help for Mobile Application Builder.

About Displays information about the product.

Mobile Application Builder toolbars

The Mobile Application Builder interface contains the following toolbars:

Project
Use the project toolbar icons above the Project pane to create a new project, open an existing project, and save the current project. See "Project toolbar options" for more information.

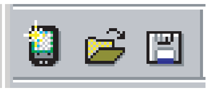
Edit
Use the edit toolbar icons above the Project pane to perform standard cut, copy, and paste actions. See "Edit toolbar options" for more information.

Control order
Use the control order toolbar icons above the Form pane to move a selected control to the front or the back of the form. See "Control order toolbar options" on page 7 for more information.

Control alignment
Use the control alignment toolbar icons above the Form pane to align selected controls. See "Control alignment toolbar options" on page 7 for more information.

Build
Use the build toolbar to build and test an application. See "Build toolbar" on page 8 for more information.

Project toolbar options



Use the project toolbar icons above the Project pane to create a new project, open an existing project, and save the current project.

New Project

Opens the Create New Project window where you can specify options for starting a new project.

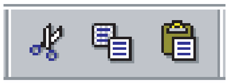
Open Project

Opens the Open Project window where you can select and open an existing project.

Save Project

Saves the current project without closing it.

Edit toolbar options



Use the edit toolbar icons above the Project pane to perform standard cut, copy, and paste actions.

The toolbar icons are:

Cut

Cuts the selected object and places it in the clipboard.

Copy

Copies the object in the clipboard to the selected location.

Paste

Pastes the object in the clipboard to the selected location.

Control order toolbar options

Use the control order toolbar icons above the Form pane to move a selected control to the front or the back of the form.

The toolbar icons are:

Bring to front

Moves the selected control to the front of the form.

Send to back

Moves the selected control to the back of the form.

Control alignment toolbar options

Use the control alignment toolbar icons above the Form pane to align selected controls.

The toolbar icons are:

Align left

Aligns the selected controls with the left edge of the anchor control.

Align center

Aligns the selected controls along the vertical axis of the anchor control.

Align right

Aligns the selected controls with the right edge of the anchor control.

Align top

Aligns the selected controls with the top edge of the anchor control.

Align middle

Aligns the selected controls along the horizontal axis of the anchor control.

Align bottom

Aligns the selected controls with the bottom edge of the anchor control.

Build toolbar

Use this toolbar to build and test an application.

The toolbar icons are:

Build

Builds the application.

Test

Tests the application.

Mobile Application Builder interface panes

The Mobile Application Builder interface contains the following panes:

Project
Use the Project pane to display and work with project resources. See “Project pane options” for more information.

Properties and events
Use the Properties and Events pane to view and edit the properties and events of selected controls. See “Properties and Events pane options” on page 9 for more information.

Form
The Form pane displays the forms that have been created. You can select a control from the palette and click the location on the form to place the control. You can select objects in the Form pane and invoke actions for them. See “Form pane options” on page 9 for more information.

Control palette
The Control Palette displays all the form controls available for the specified device. See “Control Palette options” on page 9 for more information.

Project pane options

Use the Project pane to display and work with project resources.

The Project pane displays the relationships between objects in a hierarchy. As you expand down from a particular object, the objects that reside, or are contained, in that object are displayed underneath.

Some objects do not contain other objects. Folders for these objects are displayed at the lowest level of the Project pane.

You can collapse the Project pane to display only those objects that you currently want to work with. To collapse the Project pane under a particular object, click on the minus sign (-) next to the object.

To invoke actions on an object in the Project pane, click on it with mouse button 2 to open a pop-up menu of available actions. Then select a menu choice. A window or notebook opens to guide you through the steps required to complete the action.

Another method of selecting an action for an object in the Project pane is to select it, then select the **Selected** menu, which contains the list of available actions. Then select the menu choice for the action that you want.

Properties and Events pane options

Use the Properties and Events pane to view and edit the properties and events of selected controls. On the Properties page, determine how a control looks and behaves. On the Events page, determine how a control reacts to a specific action. Each control has different possible actions.

Select a form or control in the Project pane or the Form pane, then view or edit the properties and events associated with that form or control in the Properties and Events pane.

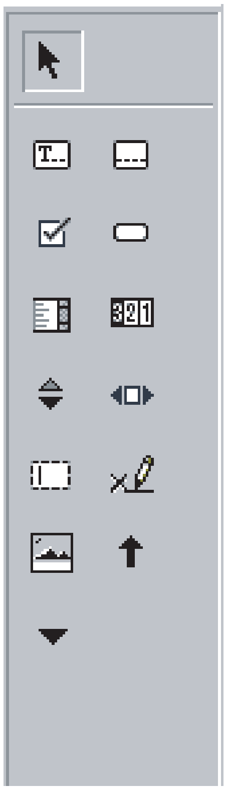
Form pane options

The Form pane displays the forms that have been created. You can select a control from the palette and click the location on the form to place the control. You can select objects in the Form pane and invoke actions for them.

To invoke actions for an object in the Form pane, select the object. Then click on the selected object with mouse button 2 to open a pop-up menu of available actions and select a menu choice. A window or notebook opens to guide you through the steps required to complete the action.

Another method of selecting an action for an object in the Form pane is to select it, then select the **Selected** menu, which contains the list of available actions. Then select the menu choice for the action that you want.

Control Palette options



The Control Palette displays all the form controls available for the specified device.

Click a control with mouse button 1 to select it. The cursor changes to a crosshair. Move the cursor over the desired location on the form, and click the form with mouse button 1 to drop the control. Hold mouse button 1 down and move the cursor to size the control.

Using keyboard shortcuts and accelerators

You can use keys or key combinations to perform operations that can also be done using a mouse.

Menu actions

Menu actions can be initiated from the keyboard in the following way:

- Press and hold the Alt key, then press the same letter as the one underlined in the name of the main menu option you want to select. For example, to select the **Documentation** menu option, press Alt+H, then press D.
- Some menu items have accelerators, which allow you to invoke the menu option without expanding the menu. For example, you can enter CTRL+S to save your project.
- To exit the main menu without selecting an option, press Esc.

Drag and drop actions

You can perform drag and drop actions, such as adding controls to a form, by enabling the numeric keypad accessibility option. To enable the numeric keypad accessibility option:

- Go to **Start —> Settings —> Control Panel**, and open **Accessibility Options**.
- On the Mouse page, select the **Use MouseKeys** check box, click **Apply**, then click **OK**.
- Enable **NumLock** on your keyboard.

After you enable the numeric keypad, use it to perform actions as described in the table below.

Table 1. Numeric keypad actions

Action	**Shortcut**
Move cursor left or right	Left or right arrow keys
Move cursor up or down	Up or down arrow keys
Select a control from the control palette or on a form	Center key (**Alternative**: Ctrl + Tab to gain focus in the **Selected Part** drop down list, then Shift + down arrow to select a resource or control from the drop down list)
Drop a control on a form	1. Move the cursor to the form and position it. 2. Click Insert to drop the control. 3. Click Insert + arrow keys to resize the control. 4. Click the center key to accept resize settings.
Resize a control on a form	1. Select the control. 2. Move the cursor to the edge of control until the cursor changes to a drag handle. 3. Click Insert + arrow keys to resize the control. 4. Click the center key to accept the resize settings.

Table 2. General keyboard shortcuts and accelerators

Action	**Shortcut**
Access the menu bar	Alt+menu shortcut character

Table 2. General keyboard shortcuts and accelerators (continued)

Action	Shortcut
Go to the next menu item	Arrow keys, or the underlined letter in the menu option
Go to the next field in a window	Tab
Go back to the previous field in a window	Shift+Tab
Create a new project	Ctrl+N
Open an existing project	Ctrl+O
Save project	Ctrl+S
Save project as...	Ctrl+A
Exit	Alt+F4

Table 3. Keyboard shortcuts for table actions

Action	Shortcut
Edit the next editable cell or give the next component focus	Tab
Edit the previous editable cell or give the previous component focus	Shift+Tab
Give the next field focus	Ctrl+Tab
Give the previous field focus	Ctrl+Shift+Tab
Edit the current cell, if it is editable	F2
Select item from pull-down menu in cell	F2, then Ctrl+Alt+DownArrow to open pull-down menu, then select item with down and up arrows and press Enter
Move to the cell above or below	Up or down arrows
Move to the cell to the left or right	Left or right arrows

Table 4. Keyboard shortcuts for tree navigation

Action	Shortcut
Gain focus in the project tree	Shift +Tab until focus is in the main menu, then down arrow key to move focus to the project tree
Navigate out forward	Tab
Navigate out backward	Shift+Tab
Expand entry	Right arrow
Collapse entry	Left arrow
Toggle expand/collapse for entry	Enter
Move up/down one entry	Up or down arrows
Move to first entry	Home
Move to last visible entry	End

Table 5. General editing actions

Action	Shortcut
Copy	Ctrl+C
Cut	Ctrl+X
Paste	Ctrl+V
Select All	Ctrl+A
Delete	Delete key

Table 6. Script editor actions

Action	Shortcut
Copy	Ctrl+C
Cut	Ctrl+X
Paste	Ctrl+V
Select All	Ctrl+A
Delete	Delete key
Find	Ctrl+F
Replace	Ctrl+H

MAB version 8.1.4 setup and configuration

It is only necessary to complete build and test setup for the mobile device platforms you are targeting.

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Testing a Mobile Application Builder application" on page 96

MAB version 8.1.4 Palm setup and configuration

Complete the following tasks to set up your Mobile Application Builder environment for building and testing Palm applications.

Mobile Application Builder version 8.1.4 Palm application development setup and configuration tasks::

- "Setting up the Palm C code building environment"
- "Setting up the Palm emulator for testing" on page 14
- "Setting up the Palm OS 5 Simulator for testing" on page 16
- "Setting up a Palm mobile device for testing" on page 17

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Testing a Mobile Application Builder application" on page 96

Setting up the Palm C code building environment

The tools below, except for the Palm OS SDK, can be downloaded from IBM on the DB2 Everyplace Web Site. They will also be on the DB2 Everyplace SDK CD. If you select the Mobile Application Builder Toolkits option during DB2 Everyplace SDK installation, the required tools are installed into the

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

directory. Most of the required tools for producing C-code applications for PalmOS are included. However, you must obtain the PalmOS SDK from Palm Computing. Read the instructions for each tool below before proceeding to the IBM download site by following the link for each tool. You will be required to enter your user ID and password for downloading from IBM sites. If you have not registered before, you will need to register before proceeding.

After installing the prerequisite tools, set preferences in Mobile Application Builder to indicate where the tools were installed.

This section provides information on installing the following required Palm C code build tools:

- Cygwin v1.3.22–1 (includes GNU PRC-tools and the PilRC Resource Compiler)
- Palm OS SDK 5

Cygwin v1.3.22–1 includes PRC-tools and the PilRC Resource Compiler for Palm OS, which are preselected for install if you obtain Cygwin from the DB2 Everyplace SDK. If you download Cygwin from the Palm Web site, you must select the following Palm OS-related packages from the Devel category during the installation:

- PRC-Tools
- PilRC
- Make component

To install Cygwin::

1. Uninstall any previous versions of Cygnus or Cygwin.
2. Run `...\SDK\ApplicationBuilder\Toolkits\Palm Development\MAB_prereqs\Cygwin\Setup.exe`.
3. On the Choose Installation Type page, select **Install from Local Directory** and click **Next**.
4. On the Choose Installation Directory page, keep the default directory (`c:\cygwin`), and all other default settings and click **Next**.
5. On the Select Local Package Directory page, keep the default directory and click **Next**.
6. On the Select Packages page, keep the defaults and click **Next**.
7. After the product installs, you can select to create icons on your desktop or the Start menu, then click **Finish** to exit the setup.

To install the Palm OS SDK 5 for PRC-Tools::

The SDK includes the Palm OS C libraries and header files. These SDKs have been modified to work with the GNU compiler. Use Palm SDK 5, for Windows, to compile applications. The Palm SDK 5, for Windows, must be downloaded directly from Palm computing. Use the download link below to get the SDK.

1. Select or create a directory to hold SDKs. This directory must match the directory you specified above for PRC-Tools 2.0.
2. Go to the Palm OS SDK version 5 page and click **I Agree** to comply with the license information. A Palm 5 SDK page will be displayed.

3. In the Palm OS SDK 5 R3 for PRC-Tools section, click **Tar**. Do *not* click **Zip**, because the zip archive does not contain the header files that are necessary for Palm development using MAB.
4.
5. Extract `palmos-sdk-5.0r3-1.tar.gz` to any directory using a program like WinZip which can extract files from ZIP files. If you do not have this type of program, you can obtain a free evaluation version from http://www.winzip.com.

Optional: To install the Palm SG SDK::

You must install this SDK to enable five-way navigation on Palm devices. To obtain this SDK, you must join the Palm PluggedIn developer program.

1. Download the `Palm SG SDK v1.0.zip` file.
2. Extract the zip file to the `\include` directory under the directory where you installed the Palm SDK 5 in the previous section. For example: `c:\PalmDev\sdk-5\include`.
3. You must open the Palm build tools preferences window as described in the next section and click **OK** for the new header files to be recognized by MAB.

To set Palm build tools preferences::

1. Start Mobile Application Builder. If this is the first time Mobile Application Builder 8.1.4 is started, a window opens reminding you to set tools preferences.
2. Open a new or existing Palm project.
3. Click **File —> Preferences**. The Preferences window opens.
4. Click **Palm Tools** to see Palm tools preferences.
5. For Cygwin, PRC tools, and PilRC, specify the path to the `\bin` subdirectory *under* the root directory where you installed Cygwin. For example, if you installed Cygwin in the `c:\cygwin` directory, specify `c:\cygwin\bin`.
6. For the Palm SDK, specify the full path to where the tool was installed. For example, if you installed `sdk-5r3` under `C:\PalmDev`, enter `C:\PalmDev\sdk-5r3`.
7. Click **OK**. The Preferences window closes.
8. Restart Mobile Application Builder so your changes take effect.

Return to Palm setup

Setting up the Palm emulator for testing

You can use the Palm OS Emulator (POSE) to test your Palm applications before downloading them to an actual Palm device. POSE 3.5 or higher is recommended. You can use this emulator to test applications for Palm OS 4.0 or earlier.

Note that the Emulator ″skins″ are not necessary unless you want to view a particular Palm OS device. The default skin is a generic Palm OS device.

To run the emulator, you must download ROM images for the level of PalmOS used on the devices you are developing applications for. There are several ways to do this:

- Inside the United States, you must join the Palm Alliance Program in order to obtain ROM images from the Palm download site. You can usually obtain the ROM images within a day or so.

- Outside the United States, you must join the Palm Developer Seeding Program in order to obtain ROM images. This process may take longer (approximately two to four weeks).
- For non-Palm manufactured but PalmOS-based devices, obtain the ROMs from the device licensee's developer program. Some of these can be downloaded from the Palm download site.

Please refer to the *Palm OS Development Tools Guide*, available on the Palm Web site, for detailed instructions on loading ROM images and using the emulator.

You must install the DB2 Everyplace database engine on the emulator to test a Mobile Application Builder application. You can also install sample applications included with Mobile Application Builder, including NurseInit, an application that initializes the DB2 Everyplace database with several tables used by the DB2 Everyplace engine and Mobile Application Builder samples.

To set up the Palm emulator for testing::

1. Download the Palm OS emulator.
2. Save the zip file to the Mobile Application Builder installation directory (for example: `C:\DB2EveryplaceSDK\SDK\ApplicationBuilder\bin`).
3. Unzip the file in this directory. The `emulator.exe` file must be located in the Mobile Application Builder installation directory in order for the user interface to find it when you test an application using the **Test** toolbar or menu item.
4. Install the DB2 Everyplace database engine on the Palm OS Emulator:
   - Start the emulator by double-clicking on `emulator.exe`. If you did not choose to obtain skin files during the emulator download, you will see a warning message to this effect when you start the emulator. You can ignore this message to use the default generic skin, or you can follow the instructions in this message to download and install skin files if you wish to view a specific Palm device.
   - Right-click anywhere on the emulator screen. Click **Install Application/Database**.
   - Install the following files:
     - `\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`

   Tip: You can also install these .prc files using Windows Explorer by dragging and dropping them onto the emulator screen.
5. After installing files on the emulator, reset the device by right-clicking anywhere on the emulator screen and clicking **Reset**.

Optional: To install NurseInit and CLP sample applications::

You can use the Command Line Processor (CLP) function to issue SQL statements against the tables.

1. Right-click anywhere on the emulator screen. Click **Install Application/Database**.
2. Install the sample applications as described in “DB2 Everyplace sample applications” on page 155.
3. After installing files on the emulator, reset the device by right-clicking anywhere on the emulator screen and clicking **Reset**.
4. Run the NurseInit application on your emulator. The NurseInit application inserts records in the tables.
5. Tap **Initialize** to create and insert records into the VNSCHEDULE, VNPERSON, VNMEDICALRECORD, VNCONTACT, and VNSIGNATURE tables.

Optional: To configure the emulator for synchronization using the Sync Server::

If your application includes synchronization capabilities, you must complete additional Sync Server configuration.

- Follow the instructions provided in the *IBM DB2 Everyplace Installation and User’s Guide* in the chapter called, ″Setting up and configuring a mobile device or emulator for synchronization.″

Optional: To change default emulator debug settings::

By default, the Palm emulator shows warnings for debugging purposes. Some of these warnings are not severe, and can be safely ignored. As an example, version 3.5x of the emulator shows warnings every time an application is closed.

1. Right-click anywhere on the emulator screen. Click **Settings —> Debugging**.
2. Clear the **MemMgr leaks** check box.

Return to Palm setup

Setting up the Palm OS 5 Simulator for testing

You can use the Palm OS 5 Simulator to test your Palm 5.x applications before downloading them to an actual Palm device. You can use this simulator only to test applications for Palm OS 5.x.

Please refer to the *Palm OS Development Tools Guide*, available on the Palm Web site, for detailed instructions on using the Simulator.

You must install the DB2 Everyplace database engine on the simulator to test a Mobile Application Builder application. You can also install sample applications included with Mobile Application Builder, including NurseInit, an application that initializes the DB2 Everyplace database with several tables used by the DB2 Everyplace engine and Mobile Application Builder samples.

You must manually save your simulator session after you have installed these files by following the steps below.

To set up the Palm OS 5 simulator for testing::

1. Download the Palm OS 5 Simulator.

2. Save the zip file to any directory.
3. Install the DB2 Everyplace database engine on the Palm OS simulator:
   - Start the simulator by double-clicking on `PalmSim.exe`.
   - Right-click anywhere on the simulator screen and click **Install —> Database** from the pop-up menu.
   - Install the following files:
     - `\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`

   Tip: You can also install these .prc files using Windows Explorer by dragging and dropping them onto the simulator screen.
4. After installing files on the simulator, reset the device by right-clicking anywhere on the simulator screen and clicking **Reset**.
5. Right-click anywhere on the simulator screen, and click **Storage —> Save** to save a snapshot of your simulator session.

Optional: To install NurseInit and CLP sample applications::

You can use the Command Line Processor (CLP) function to issue SQL statements against the tables.

1. Right-click anywhere on the simulator screen. Click **Install —> Database**.
2. Install the sample applications as described in “DB2 Everyplace sample applications” on page 155.
3. After installing files on the simulator, reset the device by right-clicking anywhere on the simulator screen and clicking **Reset**.
4. Run the NurseInit application on your simulator. The NurseInit application inserts records in the tables.
5. Tap **Initialize** to create and insert records into the VNSCHEDULE, VNPERSON, VNMEDICALRECORD, VNCONTACT, and VNSIGNATURE tables.

Optional: To configure the simulator for synchronization using the Sync Server::

If your application includes synchronization capabilities, you must complete additional Sync Server configuration.

- Follow the instructions provided in the *IBM DB2 Everyplace Installation and User’s Guide* in the chapter entitled, ″Setting up and configuring a mobile device for synchronization.″

Return to Palm setup

Setting up a Palm mobile device for testing

To set up a Palm mobile device for testing::

1. Follow the instructions provided with your mobile device to install the connection software (such as HotSync for Palm OS devices). Mobile Application Builder supports Palm OS 3.5 or higher.
2. Follow the instructions provided with the DB2 Everyplace engine, in *Installation and User's Guide*, to install the DB2 Everyplace engine on the mobile device. Install the following files:
   - `\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`

Optional: To install NurseInit and CLP sample applications::

You can use the Command Line Processor (CLP) function to issue SQL statements against the tables.

1. Install the sample applications as described in "DB2 Everyplace sample applications" on page 155.
2. Run the NurseInit application on your mobile device. The NurseInit application inserts records in the tables.
3. Tap **Initialize** to create and insert records into the VNSCHEDULE, VNPERSON, VNMEDICALRECORD, VNCONTACT, and VNSIGNATURE tables.

Optional: To configure the device for synchronization using the Sync Server::

If your application includes synchronization capabilities, you must complete additional Sync Server configuration.

- Follow the instructions provided in the *IBM DB2 Everyplace Installation and User's Guide* in Chapter 4, "Setting up and configuring a mobile device or emulator for synchronization."

Return to Palm setup

MAB version 8.1.4 Symbian Crystal setup and configuration

Complete the following tasks to set up your Mobile Application Builder environment for building and testing Symbian Crystal applications.

Mobile Application Builder version 8.1.4 setup and configuration tasks::

- "Setting up the Symbian Crystal Java code building environment" on page 19
- "Setting up a Symbian Crystal emulator for testing" on page 20
- "Setting up a Symbian Crystal mobile device for testing" on page 21

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Testing a Mobile Application Builder application" on page 96

Setting up the Symbian Crystal Java code building environment

After installing the prerequisite tools, set preferences in Mobile Application Builder to indicate where the tools were installed.

This section provides information on installing the following Symbian tool, required to build and test applications on Crystal devices such as the Nokia Communicator 92xx:

- Nokia 9200 Series SDK for Symbian OS

The Nokia 9200 Series SDK for Symbian OS, for Windows, must be downloaded directly from Forum Nokia. Use the download link below to get the SDK. You will have to register to have the SDK CD mailed to you or to download the SDK. The SDK includes a Windows desktop emulator.

The SDK includes a JRE. You can use this, or obtain JDKs from the IBM Java Technology Zone at http://www.ibm.com/developerworks/java or from Sun Microsystems at http://java.sun.com/products . If you use another JDK, set the Windows Environment PATH variable so the the JDK `\bin` directory is included.

The SDK includes Perl. If you have other versions of Perl installed, the application build process may fail if the other Perl version appears before the Symbian SDK's Perl in the Windows PATH environment variable. Verify that `C:\Perl\bin` appears before any other Perl version.

To install Nokia 9200 Series SDK for Symbian OS::

- Download the SDK from Forum Nokia. Save the Nokia 9200 Series SDK for Symbian OS zip file to any temporary directory.
- Run the Nokia 9200 Series SDK for Symbian OS executable (.exe) file. An installation wizard opens. The default installation directory is `C:\Symbian\6.x\NokiaJava`.
- Select all the components and keep the default directories for the components. Installing these components in a different drive than the SDK may result in build errors. If you already have JDK 1.2 or higher on your Windows desktop, you can choose to use it and not install the JRE.

To set Symbian build tools preferences::

If you are using both the Crystal and the UIQ SDKs simultaneously, make sure they are installed in the same root directory (example: `c:\Symbian`) and set the preferences as described below to indicate the directory where they are installed.

1. Start Mobile Application Builder. If this is the first time Mobile Application Builder 8.1.4 is started, a window opens reminding you to set tools preferences. You do not need to open or start a project.
2. Click **File —> Preferences**. The Preferences window opens.
3. Click **Symbian Tools** to see Symbian tools preferences. Specify the full path to the directory where the SDK was installed, including the drive. For example:`c:\temp`.

4. Click **OK**. The Preferences window closes.

Return to Crystal setup

Setting up a Symbian Crystal emulator for testing

The Nokia 9200 Series SDK for Symbian OS provides a Windows desktop emulator. Before you can set up the Symbian Crystal emulator for testing, you must set up the Symbian Crystal Java code building environment.

To set up the Symbian Crystal emulator for testing::

- Install the DB2 Everyplace database engine and JDBC driver on the emulator:
  - Copy the following files to

    `\Symbian\6.x\NokiaJava\Epoc32\Release\wins\urel`

    ```
    \Clients\Symbian6\database\wins\db2e.dll
    \Clients\Symbian6\database\wins\db2ejdbc.dll
    ```
  - Copy the following file to

    `\Symbian\6.x\NokiaJava\erj\ext`

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

Optional: To install and run the NurseInit application::

The NurseInit application, provided with the DB2 Everyplace SDK, is an application designed to create and populate the tables you need to complete the Symbian tutorials provided with the SDK.

1. Copy the `NurseInit.SIS` file to the `\Symbian\6.x\NokiaJava\erj` directory. The location of this file is described in "DB2 Everyplace sample applications" on page 155.
2. Open the Symbian emulator.
3. Open the Control Panel from the Extras directory.
4. Open the installation program.
5. Select **Install New**.
6. Browse to the `J:` drive, and install the `NurseInit.sis` file.
7. Close the installation program, and run the NurseInit application from **Extras**.

Optional: To improve start up time for the Symbian Crystal emulator::

Complete the following steps to improve the startup time of the Nokia emulator. If you do not complete these steps, the emulator can take up to a minute or so to open.

- Using Windows Explorer, go to

  ```
  \Symbian\6.x\NokiaJava\Epoc32\Release\wins\urel\z\system\LIBS
  ```

  .
- Create a directory named `hidden` in this location.
- Move the file `LINDASTART.DLL` from
  - `\Symbian\6.x\NokiaJava\Epoc32\Release\wins\urel\z\system\LIBS`

  to the `\hidden` directory.

- Move the file phonet.dll from

  ```
  \Symbian\6.x\NokiaJava\Epoc32\Release\wins\urel\
  ```

  to the \hidden directory.

Optional: To configure Symbian emulators for synchronization using the Sync Server::

If your application includes synchronization capabilities, you must complete additional Sync Server configuration.

- Follow the instructions provided in the *IBM DB2 Everyplace Installation and User's Guide* in the section called ″Setting up and configuring a mobile device or emulator for synchronization.″

Return to Crystal setup

Setting up a Symbian Crystal mobile device for testing

To set up a Symbian Crystal device for testing::

- Install the DB2 Everyplace database engine and JDBC driver on the device:
  - Install the following files:

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

Optional: To configure the device for synchronization using the Sync Server::

If your application includes synchronization capabilities, you must complete additional Sync Server configuration.

- Follow the instructions provided in the *IBM DB2 Everyplace Installation and User's Guide* in Chapter 4, ″Setting up and configuring a mobile device or emulator for synchronization.″

Optional: To install and run the NurseInit application::

The NurseInit application, provided with the DB2 Everyplace SDK, is an application designed to create and populate the tables you need to complete the Symbian tutorials provided with the SDK.

- Install the NurseInit.sis file on the device. The location of this file is described in "DB2 Everyplace sample applications" on page 155.
- Run the application on the device.

Return to Crystal setup

MAB version 8.1.4 Symbian UIQ setup and configuration

Complete the following tasks to set up your Mobile Application Builder environment for building and testing Symbian UIQ applications.

Mobile Application Builder version 8.1.4 setup and configuration tasks::

- "Setting up the Symbian UIQ Java code building environment"
- "Setting up a Symbian UIQ emulator for testing" on page 23
- "Setting up a Symbian UIQ device for testing" on page 24

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Testing a Mobile Application Builder application" on page 96

Setting up the Symbian UIQ Java code building environment

This section provides information on installing the following Symbian tool, required to build and test applications on UIQ devices such as the Sony Ericsson P800. After you install the tool, you will need to set build preferences in Mobile Application Builder to indicate where the tool was installed.

- Symbian OS v7.0 SDK for UIQ

The Symbian OS v7.0 SDK for UIQ, for Windows, must be downloaded directly from Ericsson Mobility World. Use the download link below to get the SDK. You will have to register to download the SDK. A Windows desktop emulator is also available for download.

The SDK includes a JRE. You can use this, or obtain JDKs from the IBM Java Technology Zone at http://www.ibm.com/developerworks/java or from Sun Microsystems at http://java.sun.com/products . If you use another JDK, set the Windows Environment PATH variable so the the JDK `\bin` directory is included.

The SDK includes Perl. If you have other versions of Perl installed, the application build process may fail if the other Perl version appears before the Symbian SDK's Perl in the Windows PATH environment variable. Verify that `C:\Perl\bin` appears before any other Perl version.

To install Symbian OS v7.0 SDK for UIQ::

1. Download the SDK from Ericsson Mobility World. Save the Symbian OS v7.0 SDK for UIQ zip files to any temporary directory. Follow the detailed instructions provided in the SDK readme.
2. Run the SDK executable (.exe) file. An installation wizard opens. The default installation directory is `X:\Symbian\UIQ_70`.
3. Select all the components and keep the default directories for the components. If you already have Active Perl and JDK installed, you can choose not to install them.
4. In the current version of the SDK, you must manually move the `aiftool.rh` file into the `Symbian\UIQ_70\epoc32\include` directory to avoid build errors. You can extract this file from com.symbian.api.GT-shared_0_0_70.sdkpkg IIRC, which is one of the SDK installation files. This file is just a normal zip file. Unzip it and extract the `aiftool.rh` file.

To set Symbian build tools preferences::

If you are using both the Crystal and the UIQ SDKs simultaneously, make sure they are installed in the same root directory (example: `X:\Symbian`) and set the preferences as described below to indicate the directory where they are installed.

1. Start Mobile Application Builder. If this is the first time Mobile Application Builder 8.1.4 is started, a window opens reminding you to set tools preferences. You do not need to open or start a project.
2. Click **File —> Preferences**. The Preferences window opens.
3. Click **Symbian Tools** to see Symbian tools preferences. Specify the full path to the directory where the SDK was installed, including the drive. For example:`c:\temp`.
4. Click **OK**. The Preferences window closes.

Return to Crystal setup

Setting up a Symbian UIQ emulator for testing

The Symbian OS v7.0 SDK for UIQ provides a Windows desktop emulator. Before you can set up the Symbian UIQ emulator for testing, you must set up the Symbian UIQ Java code building environment.

To set up the Symbian UIQ emulator for testing::

- Install the DB2 Everyplace database engine and JDBC driver on the emulator:
  1. Copy the following files to `\Symbian\UIQ_70\Epoc32\Release\winscw\urel`

     ```
     \Clients\Symbian7\database\wins\db2e.dll
     \Clients\Symbian7\database\wins\ECPKCS11.dll
     \Clients\Symbian7\database\wins\db2ejdbc.dll
     \Clients\Symbian7\database\wins\cryptoplugin.dll
     ```
  2. Copy the following file to `\Symbian\UIQ_70\erj\ext`

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

Optional: To install NurseInit and other sample applications::

The location of the Symbian UIQ sample files is described in “DB2 Everyplace sample applications” on page 155. There are two methods for installing sample applications on the Symbian UIQ emulator.

Primary method for installing sample applications::

1. Copy the .SIS files for each sample into the `\Symbian\UIQ_70\erj\examples` directory.
2. Install these files on the emulator. See the documentation provided with the emulator for detailed instructions on how to install files.

Alternative method for installing sample applications::

1. Create a project directory under the `Symbian\UIQ_70\epoc32\winscw\c\system\Apps` directory. For example, create the following directory for the PersonList application: `Symbian\UIQ_70\epoc32\winscw\c\system\Apps\PersonList`.
2. In the directory you created, copy all files *except* the .SIS file from

`Client\*platform*\database\Samples\*Language*`

The sample application is automatically installed on the emulator.

Return to Crystal setup

Setting up a Symbian UIQ device for testing

This section describes how to install the DB2 Everyplace libraries and sample applications on a Symbian OS Version 7 UIQ mobile device. Before you install, you need to perform the following steps:

- On your Windows workstation, install and configure the connection software that came with the Symbian UIQ mobile device.
- Connect the Symbian UIQ mobile device to the workstation. Use the mobile device's documentation to ensure that it is connected correctly.

To set up a Symbian UIQ device for testing::

To install the DB2 Everyplace libraries and sample application files manually, use the install tool from the connection software that is included with the Symbian OS UIQ mobile device.

- Install the DB2 Everyplace database engine and JDBC driver on the device:
  - `\Clients\Symbian7\database\armi\DB2e.sis`
  - `\Clients\Symbian7\database\armi\DB2eJDBC.sis`

Optional: To install NurseInit and other sample UIQ applications::

The Symbian UIQ sample files are located in `\Clients\Symbian7\database`. Install the following files on the device. Locations for these files are described in "DB2 Everyplace sample applications" on page 155.

Table 7. Symbian UIQ sample applications for the device

File Name	Description
`DB2eCLP.sis`	Command Line Processor (CLP) application
`PersonList.sis`	PersonList sample application
`NurseInit.sis`	NurseInit sample application
`DB2e_Symbian7.sis`	DB2 Everyplace package. This package includes: • DB2e.sis • DB2eJDBC.sis • SampleCLP.sis • NurseInit.sis • PersonList.sis

Optional: To configure Symbian devices for synchronization using the Sync Server::

If your application includes synchronization capabilities, you must complete additional Sync Server configuration. Install the following files on the device:

```
\Clients\Symbian7\sync\<locale>\processor\ISync.sis
\Clients\Symbian7\sync\<locale>\processor\upgrade\IUpgrade.app
\Clients\Symbian7\sync\<locale>\processor\upgrade\IUpgrade.rsc
```

To test synchronization on the device, make sure the the device is connected to the workstation using its connectivity software. Select ISync from the applications list. From the **File —>Settings** dialog, enter the IP, port, user, and password. Test synchronization by going to **File —>Synchronize**.

Return to Crystal setup

MAB version 8.1.4 WinCE setup and configuration

Complete the following tasks to set up your Mobile Application Builder environment for building and testing WinCE applications.

Mobile Application Builder version 8.1.4 setup and configuration tasks::

- "Setting up the WinCE JVM preferences"
- "Setting up for testing WinCE or Sharp Zaurus applications on the desktop" on page 27
- "Setting up a WinCE mobile device for testing" on page 28

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Testing a Mobile Application Builder application" on page 96

Setting up the WinCE JVM preferences

For WinCE/PocketPC devices, for which a Java Virtual Machine (JVM) does not come built-in, installed on the device, or in device ROM, MAB-built applications have been tested with the following vendor Java Virtual Machines:

- IBM J9 (as included with the DB2 Everyplace SDK or with WebSphere Studio Device Developer (WSDD)
- Insignia Corporation's Jeode PDA Edition version 1.9
- Sun Microsystem's Personal Java Runtime Environment 1.0
- CrEme Plus JVM, version 3.21 (required for barcode scanning applications)

Mobile Application Builder generates application installables for any device with a JVM. The WinCE JVM preferences should be set appropriately for the JVM you are using before the application is built. Preferences for Insignia, Sun, and CrEme Plus JVMs are built in to MAB. The following section contains information on obtaining and installing the following JVMs:

- IBM J9 Java Virtual Machine
- CrEme Plus JVM

See the *DB2 Everyplace Installation and User's Guide* for detailed information on obtaining J9 from WSDD and using it with MAB, or follow the steps below.

To obtain the IBM J9 Java Virtual Machine from WSDD and install on a device::

1. Download WSDD and related products from the WSDD web site
2. Copy the following files or directories to the device from the WSDD installation. The directory structure described below must be maintained for the J9 executable to work properly.
   - Install the following files, obtained from `<WSDD>\wsdd5.0\ive\lib\jclMax`, in the `\wsdd\lib\JCLMax` directory on the device:
     - `classes.zip`
     - `profile.far`
   - Install the following files, obtained from `<WSDD>\wsdd5.0\eclipse\plugins\com.ibm.ive.tomcatsupport_5.0.0`, in the `\wsdd\lib\JCLMax` directory on the device:
     - `iverel20.dll`
     - `j9.exe`
     - `j9dyn20.dll`
     - `j9max20.dll`
     - `j9midp20.dll`
     - `j9prt20.dll`
     - `j9thr20.dll`
     - `j9vm20.dll`
     - `j9w.exe`
     - `j9zlib20.dll`
     - `swt-win32-2104.dll`
   - For non-English projects, install the following files, obtained from the directories shown below, in the `\wsdd\lib` directory on the device:
     - `<WSDD>\wsdd5.0\ive\lib\charconv.zip`
     - `<WSDD>\wsdd5.0\ive\lib\jclMax\locale.zip`

To configure Mobile Application Builder to use IBM J9::

You must set WinCE JVM preferences and the **Database Location** property of the project to use the IBM J9.

1. Start Mobile Application Builder. If this is the first time Mobile Application Builder 8.1.4 is started, a window opens reminding you to set tools preferences. You do not need to open or start a project.
2. Click **File —> Preferences**. The Preferences window opens.
3. Click **Application —> WinCE JVM Preferences**.
4. Select **IBM J9 JVM** from the list of JVMs.
5. Replace any defaults with the following settings:
   - In the **JVM path** field, type `\WSDD\bin\j9.exe`.
   - In the **Arguments** field, type

     ```
     "-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
```

where

- `X` = project name
- `\WSDD` = Filesystem location on the device where J9 is installed

6. Click **OK**. The Preferences window closes.
7. Click WinCE in the Project pane.
8. On the Properties page of the Properties and Events pane, change the **Database Location** property from \ to \\.

To obtain the CrEme Plus JVM::

Use the CrEme Plus JVM if you are developing WinCE applications that use barcode scanning.

1. Download the CrEme Plus JVM from the NSIcom web site.
2. On the download page, select CrEme Plus in the **NSIcom Software** field, and specify other options depending on your target device.
3. After completing all download steps, run `setup.exe` with your device connected to install the JVM on your device.

To configure Mobile Application Builder to use CrEme Plus JVM::

You must set WinCE JVM preferences to use the CrEme Plus JVM.

1. Start Mobile Application Builder. If this is the first time Mobile Application Builder 8.1.4 is started, a window opens reminding you to set tools preferences.
2. Open a new WinCE project.
3. Click **File —> Preferences**. The Preferences window opens.
4. Click **Application —> WinCE JVM Preferences**.
5. Select **NSIcom — CrEme Plus JVM** from the list of JVMs.
6. Keep the default in the **Enter JVM Path** field.
7. In the Enter JVM Arguments field, you must append the database location path (the location of the database tables on the device) that is set for the current WinCE project to the list of arguments, separated by a space. You can see the database location property by clicking WinCE in the Project pane and viewing the **Database Location** property on the Properties page of the Properties and Events pane.

   For example: If the Database Location property is `\windows`, type a space after the last entry in the **Enter JVM Arguments** field, then type `\windows`.
8. Click **OK**. The Preferences window closes.

Return to WinCE setup

Setting up for testing WinCE or Sharp Zaurus applications on the desktop

WinCE/PocketPC applications cannot be tested on the WinCE emulator from Microsoft (eMbedded Visual Tools 3.0 product). A real device must be used, or, the application can be run on the Windows desktop (Win32). You can also use these instructions to set up Sharp Zaurus applications on the desktop.

To set up for testing WinCE or Sharp Zaurus applications on the desktop::

1. Set WinCE emulation preferences in Mobile Application Builder.
   a. Start Mobile Application Builder. If this is the first time Mobile Application Builder 8.1.4 is started, a window opens reminding you to set tools preferences. You do not need to open or start a project.
   b. Click **File —> Preferences**. The Preferences window opens.
   c. Click Win32 Emulation Requirements.
   d. Verify that the settings are correct and match where the DB2 Everyplace database engine and your tables are installed. The default for the database location is

      `c:\DB2EveryplaceSDK`, and the default for the tables location is

      ```
      C:\DB2EveryplaceSDK\SDK\ApplicationBuilder
      \Projects\Samples\Win32\Tables
      ```

      .
   e. Click **OK**. The Preferences window closes.

Optional: To create and populate tables used by WinCE sample applications::

`NurseInit.bat`, in the `\Projects\Samples\Win32` directory, is a sample script that uses the supplied `SchemaInit.java` program to create and populate database tables on Win32 based on row content in an ASCII file.

1. Modify `NurseInit.bat` to adjust the path information to where you installed DB2 Everyplace.
2. Run `NurseInit.bat`. The file creates the schema used by the PersonList and VNApp samples, on the desktop, at a directory of your choice. SchemaInit will drop the tables from the database if they already exist, and will also create the target directory if it does not exist.

Return to Crystal setup

Setting up a WinCE mobile device for testing

To test on a WinCE device, you must install the DB2 Everyplace database engine, JDBC driver, and a Java Virtual Machine (JVM) on the device.

To set up a WinCE device for testing::

1. Install the DB2 Everyplace database and JDBC files appropriate for your device's processor type and WinCE version from the `\Clients\Win32\database\x86` directory. See the *DB2 Everyplace Installation and User's Guide* for an explanation of the required files.
2. Obtain and install an appropriate JVM for WinCE devices. MAB-generated applications have been tested with JVMs from the following sources:
   - Insignia solutions at http://www.insignia.com for ARM processor, such as iPAQ 3630.

- Sun Microsystems at http://www.java.sun.com for MIPS and SH3 processors such as Compaq Aero or HP Jornada.

3. Install the JVM on the device following the directions provided with the JVM. When asked to choose a path during the installation, choose the default paths.

Return to Crystal setup

MAB version 8.1.4 Sharp Zaurus setup and configuration

Complete the following tasks to set up your Mobile Application Builder environment for building and testing Sharp Zaurus applications.

Mobile Application Builder version 8.1.4 Zaurus setup and configuration tasks::

- “Setting up the Sharp Zaurus building environment”
- “Setting up for testing Sharp Zaurus applications on the desktop” on page 30
- “Setting up a Sharp Zaurus device for testing” on page 30

Related tasks:

“Building a Mobile Application Builder application” on page 91
“Testing a Mobile Application Builder application” on page 96

Setting up the Sharp Zaurus building environment

Before you build applications for the Sharp Zaurus, you need to install Cygwin and add the Cygwin `\bin` directory to Windows environment user variable PATH. Cygwin installation for Sharp Zaurus is the same as Cygwin installation for Palm. However, for Zaurus you must manually add the `\bin` directory to the PATH variable instead of setting build preferences through Mobile Application Builder as you would do for Palm.

The Cygwin utilities used for Sharp Zaurus application builds are `tar.exe` and `gzip.exe`. Other versions of these utilities may be used, however, only the Cygwin versions of these utilities have been tested. If you are using other versions of these utilities you must also add them to the PATH variable as described below.

To set up the Sharp Zaurus building environment::

1. Install Cygwin as described in the section on ″Setting up the Palm C code building environment″.
2. Add *`<cygwin installation directory>`*`\cygwin-b20\h-i586-cygwin32\bin` to the beginning of the Windows 2000 environment user variable PATH, using the following method. Other Windows operating systems will require a similar method. See Windows Help for specific instructions if you are not using Windows 2000.
   a. Go to **Start —> Settings —> Control Panel**.
   b. Double-click the System icon to modify the System Properties.

c. Click on the **Advanced** tab.
d. Click the **Environment Variables** button.
e. Edit the PATH variable under either User or System Variables, depending on whether you want to use the builder under just the user who is currently logged in, or under any user. For User, you may have to add the PATH variable via the **New** button if PATH is not already defined.

“MAB version 8.1.4 Sharp Zaurus setup and configuration” on page 29

Setting up for testing Sharp Zaurus applications on the desktop

Sharp Zaurus applications cannot be tested on an emulator. A real device must be used, or, the application can be run on the Windows desktop (Win32). Follow the instructions provided in the section ″Setting up for testing WinCE applications on the desktop″, not including optional steps related to WinCE sample applications, to test Zaurus applications using Win32 desktop simulation.

“MAB version 8.1.4 Sharp Zaurus setup and configuration” on page 29

Setting up a Sharp Zaurus device for testing

This section describes how to install the DB2 Everyplace libraries and sample applications on a Sharp Zaurus mobile device. Before you install, you need to perform the following steps:

- On your Windows workstation, install and configure the connection software that came with the Sharp Zaurus mobile device.
- Connect the Sharp Zaurus mobile device to the workstation. Use the mobile device’s documentation to ensure that it is connected correctly.

To set up a Sharp Zaurus device for testing::

To install the DB2 Everyplace libraries and sample application files manually, use the install tool from the connection software that is included with the Sharp Zaurus mobile device. Install the following file: `\Clients\embeddedLinux\database\install\db2e-libs_8.1.2_arm.ipk`. This file installs the following items in the `/usr/bin` directory on your device:

- All required DB2 Everyplace database engine binaries
- All required synchronization binaries
- `db2ejdbc.jar`
- `isync4j.jar`

Optional: To install NurseInit and other sample Zaurus applications::

The location for the Zaurus sample files is described in “DB2 Everyplace sample applications” on page 155.

1. Install the following files on the device:

Table 8. Sharp Zaurus sample applications for the device

File Name	Description
`PersonList_1.0_arm.ipk`	PersonList sample application
`NurseInit_1.0_arm.ipk`	NurseInit sample application

2. After you install the NurseInit sample application, run it on the device. This application creates and insert records into the VNSCHEDULE, VNPERSON, VNMEDICALRECORD, VNCONTACT, and VNSIGNATURE tables, and places them in `/home/db2e/database` on the device.
3. Back in Mobile Application Builder, click **File —> Preferences** to open the Preferences window.

"MAB version 8.1.4 Sharp Zaurus setup and configuration" on page 29

Uninstalling Mobile Application Builder

Uninstall any previous versions before installing Mobile Application Builder.

To uninstall Mobile Application Builder::

1. From your Windows desktop, go to **Start -> Settings -> Control Panel -> Add/Remove Programs**.
2. Select **IBM DB2 Everyplace Software Development Kit** from the list of installed programs.
3. Click **Add/Remove** to uninstall the product.
4. Click **Yes** when it asks if you want to remove the product.
5. Click **OK** when uninstall is complete and close the Add/Remove Programs window.

Related tasks:

"MAB version 8.1.4 setup and configuration" on page 12

MAB projects

An MAB project contains a framework in which all the controls and resources to build an application can be stored and organized for development.

Related tasks:

"Application IDs" on page 42

Opening a project from the Welcome window

Each time Mobile Application Builder is started, it displays a welcome window. The welcome window is a starting point where you can create a new project, work with an existing project, or work on the last project you had open.

To open a project from the Welcome window::

In the Welcome window, complete the following tasks:

1. Specify what type of project you want to open.
   - Select **Start a new project** to open a window where you can create a new project.
   - Select **Open an existing project** to work on a project that was created in a previous session.
   - Select **Continue working on the last project (***project name***)** to open the last project. This option is only available if Mobile Application Builder detects a recent project.
2. Optional: Clear the **Show this window at startup** check box if you do not want this window to open at the next startup of Mobile Application Builder. To restore the window from within Mobile Application Builder, click **File —> Preferences**, click **General**, and check the **Show welcome dialog at startup** check box..
3. Click **OK**. The Welcome window closes and your selections are applied.

Related tasks:

"Creating a new project"
"Opening an existing project" on page 33

Creating a new project

A Mobile Application Builder project contains a framework in which all the controls and resources to build an application can be stored and organized for development.

Some initial project information needs to be entered to create a project.

To create a new project::

1. Open the New Project window using one of the following methods:
   - Click **New project** from the **File** menu.
   - Click from the toolbar.
2. In the **Project Name** field, type a name for the project. See "Naming conventions for projects and applications" on page 34 for naming conventions and restrictions. This name will be used as part of the project directory. The project name can be edited later.

3. In the **Project Directory** field, identify a directory to save the project in, or accept the default `..\ApplicationBuilder\Projects` directory. A directory with the same name as the project will be created below the directory you identify in this field. This is where the project file will be saved. See "Naming conventions for projects and applications" on page 34 for naming conventions and restrictions.
4. In the **Application Name** field, type a name for your application. See "Naming conventions for projects and applications" on page 34 for naming conventions and restrictions. This name will appear as the name of the application on the application picker screen of the target device.
5. In the **Target Device** field, select the mobile device the application will run on. Options in the fields below the **Target Device** field will change depending on the mobile device you select.
6. In the **Application ID** field, enter an ID for your application. If there is no **Application ID** field, the platform you are targeting does not require an application ID. See "Application IDs" on page 42 for more information.
7. In the **Form Size** field, specify a default form size for your application. All new forms for this project will be created with this default size. The default form size can be modified for individual forms within your project. This field is only editable if you select Generic Java in the **Target Device** field. For Generic Java applications, you can select the default size (1/4 VGA)or specify the width and height for the form.
8. Click **Finish**. Your project appears in the Project pane. If **Finish** is not available, make sure there are no spaces at the beginning or end of the project name, or that you have not violated any other naming requirements in the **Name** and **Application Name** fields.

Related tasks:

"Opening an existing project"
"Opening a project from the Welcome window" on page 32
"Renaming a project or an application" on page 35

Opening an existing project

A Mobile Application Builder project contains a framework in which all the controls and resources to build an application can be stored and organized for development.

To open an existing project, use one of the following methods::

-
  1. Click **File —>** 📂 **Open Project**, or click 📂 from the toolbar.
  2. Locate the project you want to open and click **Open**.
- Click **File —>** 📂 ***projectname*** from the main menu. The File menu contains entries for the most recently opened projects.

Related tasks:

"Creating a new project" on page 32
"Opening a project from the Welcome window" on page 32
"Renaming a project or an application" on page 35

Naming conventions for projects and applications

There are some restrictions on names for projects, project directories, and applications.

This table explains naming conventions and restrictions for projects.

Table 9. Project name conventions and restrictions

All platforms	Use only SBCS/ASCII characters for the project name. Do not use DBCS characters for the project name. The project name cannot be null. The project name cannot start with or end with a space or a period.
Java code platforms	The project name cannot contain any spaces.

This table explains naming conventions and restrictions for project directories.

Table 10. Project directory conventions and restrictions

All platforms	The project directory cannot be null. The project directory cannot end with a slash (\ ).
Symbian platforms	In addition to the restrictions for all platforms: The project directory cannot include DBCS characters or spaces.

This table explains naming conventions and restrictions for applications.

Table 11. Application name conventions and restrictions

All platforms	The application name cannot be null. The application name cannot start with or end with a space or a period.
Symbian platforms	In addition to the restrictions for all platforms: Do not use DBCS characters for the application name. DBCS characters *can* be used in the application name for other platforms. The application name cannot contain any spaces or periods, and cannot consist of a single character.

Related tasks:

"Creating a new project" on page 32
"Renaming a project or an application" on page 35

Related reference:

"Troubleshooting" on page 148

Renaming a project or an application

You can change the project name or the application name for the Mobile Application Builder project you are working on.

To rename a project or application::

1. Click the project name in the Project pane.
2. Click the **Name** field or the **Application Name** field in the Properties and Events pane. A window opens where you can specify a new name for the project or application.
3. In the **New name** field, type a new name for your project or application that conforms to the naming requirements for the current target platform. See "Naming conventions for projects and applications" on page 34 for details on naming requirements.
4. Click **OK**. the window closes and the new name appears in the **Name** field or the **Application Name** field. If the OK button is not available, make sure the name you typed conforms to the requirements for the current target platform.

Related tasks:

"Creating a new project" on page 32
"Opening a project from the Welcome window" on page 32

Configuring project properties

You must open a project before you can configure project properties. Project properties vary depending on your target platform.

Most project properties already have default values set.

To configure project properties::

1. Click <*Target OS*> in the Project pane.
2. Modify project properties on the Properties page of the Properties and Events pane. See "Project properties" on page 110 for detailed explanation of each project property.

Related reference information:

"Project properties" on page 110

Adding source files to a project

Use source files (.c and .h files) to compile as part of a Palm project. Source files can be added to any Palm project. Use the Source and Header Files window to add, edit, remove, or reorder source files to be used.

Support for source files varies by target platform.

To add source files to a project::

1. Click **Global Definitions** in the Project pane.
2. Click … in the Properties page in the Properties and Events pane next to Source files to open the Source and Header Files window.
3. Click **Add** to select the source file to include.
4. In the window that opens, type or locate the source file name in the **File** field, then click **OK**.
5. Click **OK**. The Source and Header Files window closes.

Related tasks:

"Adding library files to a project"

Adding library files to a project

Use library files to add shared libraries to an application. Shared libraries can provide more features and function to your application. Use the Shared Library Files window or the Java Library Files window to add, edit, remove, or reorder library files to be used. Locate the information to define the shared library in the documentation supplied with the library file.

Support for library files varies by target platform.

To add library files to a Palm C code project::

1. Click **Global Definitions** in the Project pane.
2. Click … in the Properties page in the Properties and Events pane next to Library files to open the Shared Library Files window.
3. Click **Add** to define the shared library file to include.
4. In the window that opens, enter the following information:
   - Pointer to the library name (`CharPtr`)
   - The global variable for storing the library reference (`UIntPtr`)
   - Type of library (`DWord`)
   - Creator of library (`DWord`)
5. Click **OK**.
6. Click **OK** again to close the Shared Library Files window.

To add library files to a Java code platform project::

You can add jar files to a Java platform project. You can also specify to have these files included in the build and installed onto the device along with the application. Note that other files that are *used by* these jar files (for example, .dll files) must be installed separately onto the device or emulator.

1. Click **Global Definitions** in the Project pane.
2. Click [...] in the Properties page in the Properties and Events pane next to Library files to open the Java Library Files window.
3. Click **Add** to define the shared library file to include.
4. In the window that opens, type or browse to a file name in the **File** field. Only jar files can be added.
5. Optional: Check the **Check to include file in installable** check box to have this file installed onto the device when the application is installed on the device.
6. Click **OK**.
7. Click **OK** again to close the Java Library Files window.

Related tasks:

"Adding library files to a project" on page 36

Saving and closing a project

Save and close your project when you are done. You can also save a project without closing it.

To save and close a project::

1. Click the project name in the Project pane.
2. Click **File —> Close Project**. If you have made changes to your project, a window opens prompting you to save your changes.
3. Click **Yes** to save your changes and close your project.

To save a project without closing::

1. Click 💾 on the toolbar, or click **File —>** 💾 **Save Project** from the main menu.

Related tasks:

"Creating a new project" on page 32
"Opening an existing project" on page 33

Setting preferences

You can customize application settings for your copy of Mobile Application Builder.

Preference options vary by target platform.

Related concepts:

"Scripting support" on page 136

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Adding print capabilities to an application" on page 85
"MAB projects" on page 31
"Scripting in Mobile Application Builder" on page 74
"MAB version 8.1.4 setup and configuration" on page 12

Setting general preferences

You can customize general settings for your copy of Mobile Application Builder.

To set general preferences::

1. Click **File —> Preferences**. The Preferences window opens.
2. Click the **General** node to see general preferences.
3. Optional: Select **Show welcome dialog at startup** to automatically launch the New Application wizard when Mobile Application Builder starts.
4. Optional: In the **Locate browser to display online help (full path and file name)** field, select a default browser to display the online help. Click the **Browse** button to locate your browser's executable file.
5. Click **OK**. The Preferences window closes.

Related tasks:

"Setting preferences"

Setting build preferences

You can customize build preferences for your copy of Mobile Application Builder.

Adding debug symbols to your application allows you to test your application and use GDB to debug it.

Palm OS has a code segment limitation of 32K per segment. If your application exceeds 32K, using a single code segment, you will receive an overflow error and the application will not compile. Multiple code segments are used, by default, to avoid this limitation. Multiple code segmenting separates code globally and by form so that each segment remains below 32K.

Disable multiple code segments to make a small application slightly more efficient. It is recommended that you disable multiple code segments before building your application to use with CodeWarrior.

To set build preferences::

1. Click **File —> Preferences**. The Preferences window opens.
2. Click the Build node to see build preferences.
3. Optional: Under **Build Mode**, select an option:
   - Select **Debug** to add debug symbols to your application at build time. If you select this option, intermediate files are saved to the project directory at build time to help you debug your application.
   - Select **Release** to optimize your application for performance by not including debugging symbols. If you select this option, no intermediate files are saved after a successful build.
4. Optional: Clear the **Compile with multiple code segments** check box to make a small application slightly more efficient.
5. Optional: Select the **Enable Five Way Navigation** check box to enable code generation for Palm five-way navigation buttons. See “Enabling Palm five-way navigation” on page 57 for more information.
6. Optional: Select a printing solution from the **Choose Print library for building** list.
7. Click **OK**. The Preferences window closes.

Related tasks:

“Setting preferences” on page 38

Setting application preferences

You can customize application settings for your copy of Mobile Application Builder.

To set application preferences::

1. Click **File —> Preferences**. The Preferences window opens.
2. Expand the **Application** node to see application preferences.
3. Optional: Click **Messages** to select whether to include confirmation or response message dialogs before or after database actions. Choose a default message or customize your own message.
4. Optional: Click **UI Color** to customize your application by choosing colors for UI elements, including objects (such as buttons and check boxes), menus, fields, forms, dialogs and alerts. Support for UI color options varies by target platform.

5. Optional: Click **Font** to customize your application by choosing font options. Support for font options varies by target platform.
6. Optional: Click **WinCE JVM Preferences** to set preferences for building WinCE applications.
7. Click **OK**. The Preferences window closes.

Related tasks:

"Setting preferences" on page 38

Setting scripting preferences

On the script page you can customize the script window to make reading and writing easier for you. Changes made here affect only the text that appears in the script window, not in the mobile device.

To set scripting preferences::

1. Click **File —> Preferences**. The Preferences window opens.
2. Expand the Script node to see scripting preferences.
3. Click the Script node to set the behavior of script editor elements.
4. Optional: Select an option under **Indentation** to customize the types of indentation to be used within the script window.
5. Optional: Select an option under **Find and Replace** to customize find and replace behavior.
6. Optional: Select **Include Mobile Application Builder generated functions in the project view** to display MAB functions in the Project pane with functions you have generated.
7. Click the Text Style node to customize the font appearance and size for text in the script window.
8. Optional: Select a font type from the **Font** list.
9. Optional: Select a font size from the **Size** list.
10. Optional: Select **Use color syntax highlighting** to display different types of text in different colors, then choose colors for Base text, Constants, Keywords, Comments, and Errors.

Related concepts:

"Scripting support" on page 136

Related tasks:

"Setting preferences" on page 38

Setting printing preferences

You can customize printing preferences for your copy of Mobile Application Builder.

Support for printing varies by target platform.

To set print preferences::

1. Click **File —> Preferences**. The Preferences window opens.
2. Click the **Build** node to see build preferences.
3. Select a printing solution from the **Choose Print library for building** list.
4. Click **OK**. The Preferences window closes.

Related tasks:

"Setting preferences" on page 38
"Adding print capabilities to an application" on page 85

Related reference information:

"Printing support" on page 137
"Printing API" on page 138

Setting preferences for prerequisite tools

For the initial setup of Mobile Application Builder, you must indicate path settings for prerequisite tools. You only need to indicate path settings for the platform you are targeting. You can install prerequisite tools in any directory.

To set prerequisite tools preferences::

1. Click **File —> Preferences**. The Preferences window opens.
2. Optional: Click **Palm Tools** to see Palm tools preferences. For each tool, specify the path to the root directory where the tool was installed.
3. Optional: Click **Symbian Tools** to see Symbian tools preferences. Specify the full path to the directory where the Symbian SDK was installed, including the drive letter. For example, `c:\temp`.
4. Optional: Click **WinCE Emulation Path Requirements** to see WinCE preferences. Specify the path to the root directory where the DB2 Everyplace database engine was installed, and specify the path to the Win 32 database tables on the desktop.
5. Click **OK**. The Preferences window closes.

Related tasks:

"MAB version 8.1.4 setup and configuration" on page 12
"Setting preferences" on page 38

Application IDs

Some mobile device operating systems use a unique identifier for each application running on the system. To prevent conflicts between your applications and other applications on the device, these mobile device operating systems maintain an official list of these application IDs.

When you are targeting a device with this requirement, a registered application ID needs to be obtained. The same application, with the same ID, can be subsequently downloaded to a device or emulator without any problems. If there is a problem, delete the application on the device or emulator before downloading it again.

Related tasks:

"Creating a new project" on page 32

Registering an application ID

Some mobile device operating systems use a unique identifier for each application running on the system. To prevent conflicts between your applications and other applications on the device, these mobile device operating systems maintain an official list of these application IDs.

To register a Palm application ID::

1. Search the Palm OS application ID database to ensure that the ID you want to use is not already taken.
2. Reserve the application ID using the online registration form.

To register a SymbianOS Unique Identification (UID) number::

- Symbian requires you to send them an e-mail requesting one or more UIDs. UIDs will be assigned to you by Symbian.
- See Symbian's web site for more information.

Related tasks:

"Setting or changing an application ID"

Setting or changing an application ID

This task is only necessary when you are targeting a device that requires a registered application ID. When you are targeting a device with this requirement, a registered application ID needs to be obtained before an application goes into the field.

Mobile Application Builder generates an application ID for you. If your target platform requires a registered application ID, complete the steps below to change your application ID to the ID you have registered.

To set or change an application's ID::

1. Click **<mobile device OS>** in the Project pane.
2. Edit the **Application ID** field on the Properties page of the Properties and Events pane. If there is no **Application ID** field on the Properties page, your target device does not require a registered application ID.

Related tasks:

"Registering an application ID" on page 42

Tables and data sources

Table definition files (*.ddl) are needed when developing your application. A table definition file needs to contain a CREATE TABLE statement. See the *Application Development Guide*, installed with the DB2 Everyplace engine, for more detailed information about CREATE TABLE statements.

Related examples:

"Example of a CREATE TABLE statement" on page 157
"Defining database queries" on page 157

Importing a table using a live connection

You can use a live connection to a remote database to obtain metadata for device-side tables and columns. DB2 and Oracle databases are supported.

Prerequisites:

Before you import a table using a live connection, you must set the Cygwin path in the Preferences window, as described in "Setting up the Palm C code building environment" on page 12. This step must be completed for all target platforms.

To import a table from a remote database table::

1. Provide the path to the JDBC driver on your workstation. You only need to complete this step once.
   a. Click **File —> Preferences**. The Preferences window opens.
   b. Expand the Application node to see application preferences.
   c. Click JDBC driver location.
   d. In the right panel, specify the DB2 or Oracle database version, and the directory where the JDBC driver is located.
   e. Click **OK**.
   f. Restart MAB so your changes take effect.

2. Right-click Tables in the MAB Project pane, and click **Import table from a remote database table** from the pop up menu. A Connection window opens.
3. In the Connection window, select the type of database you want to import, and fill in all other required fields.
4. Click **OK**. An Import window opens.
5. In the Import window, select tables and columns from the **Available tables** list, and click > to move them into the **Tables to import** list. You can select a table, which will import all of the columns in that table, or you can select individual columns. You can also select multiple tables or columns simultaneously.
6. Optional: Rename the table.
   a. Right-click the table in the **Tables to import** list. A window opens where you can specify a new name for the table.
   b. Specify a new name, and click **OK**.
7. Click **Import**. The table metadata will be written into a DDL file in the `Projects\` directory, and will also appear in the MAB Project pane under the Tables node.

Related tasks:

"Creating and importing a table definition file"
"Updating a table definition file" on page 45
"Deleting a table definition file" on page 45
"Creating a new form" on page 48

Creating and importing a table definition file

In order for information collected in your application to be populated into tables correctly, table definitions must be imported into the project. Importing a table definition (*.ddl) provides the project and its resources with a blueprint of columns and data types allowed.

If changes are made to a table definition file (*.ddl) after it has been imported into a project it will need to be removed and imported again in order for the changes to be available in the project. After a table has been removed and imported again, you must reset all existing references to the table.

If you are using tables that exist on another database, such as DB2 Universal Database (UDB), you can sometimes generate the definition into a file. For DB2 UDB, use the **Generate DDL** menu function for the table in the Control Center. You may have to edit the file to make minor changes, but it will be more accurate than re-creating the .ddl from scratch.

To create and import a table definition file::

1. Create a *.ddl file, using a text editor, containing a CREATE TABLE statement along with field definitions. See the *Application Development Guide*, installed with the DB2 Everyplace engine, for more detailed information about CREATE TABLE statements.
2. Right-click Tables in the Project pane.
3. Click **Import table** from the pop-up menu.

4. Locate the appropriate .ddl file, and click **Open**.

Related tasks:

“Updating a table definition file”
“Deleting a table definition file”
“Creating a new form” on page 48

Related examples:

“Example of a CREATE TABLE statement” on page 157
“Defining database queries” on page 157

Updating a table definition file

Modify the table definition file as needed. Once changes are made, import the table definition file again to make the changes available in your project.

After a table has been removed and imported again, you must reset all existing references to the table.

To update a table definition file that has already been imported into a project::

1. Right-click the table name in the Project pane, and click **Delete**.
2. Right-click ▦ Tables in the Project pane.
3. Click **Import table** from the pop-up menu.
4. Locate the appropriate .ddl file and click **Open**.
5.

Related tasks:

“Creating and importing a table definition file” on page 44
“Deleting a table definition file”

Related examples:

“Example of a CREATE TABLE statement” on page 157
“Defining database queries” on page 157

Deleting a table definition file

Delete a table definition file to remove the table structure from the project.

To delete a table definition file::

1. Right-click the table that you want to delete in the Project pane.
2. Click **Delete** from the pop-up menu.

Related tasks:

"Creating and importing a table definition file" on page 44
"Updating a table definition file" on page 45

Related examples:

"Example of a CREATE TABLE statement" on page 157
"Defining database queries" on page 157

Defining data sources for a list control

Use the List data sources window to define the table definition files and fields to display in your list.

You must import a table into the project before you can define data sources for a list control.

Data source and Data field identify the database table and column associated with a field. All of the Data source and Data field associations on a form are used to build SQL queries for that form. When you run your application, these queries are executed and the retrieved data is displayed on the form. If you have associated database update events (Create record, Update record, Delete record) with the form, then SQL statements are executed that update the database with the current contents of the form when the events are triggered.

Link source and Link field identify the database table and column used to build a SQL query that joins information from two database tables.

For example, the first form of the Visiting Nurses application has a list of appointment times and patient IDs. This information is obtained by joining the VNSchedule and VNPerson tables. The table columns that define how to join the tables are VNSchedule.PatientID and VNPerson.ID. Selecting the following options builds a query for the form that joins the contents of the VNSchedule and VNPerson tables where VNSchedule.PatientID equals VNPerson.ID.

- Data source: VNSchedule
- Data field: PatientID
- Link source: VNPerson
- Link field: ID

To define data sources for a list control::

1. Select a list control.
2. Click the Properties tab in the Properties and Events pane.
3. Click … in the field next to Data sources. The List Data Sources window opens.
4. Optional: In the **Data source** field, select a table definition from those that have been imported into the project. Only the tables imported into the project appear in the **Data source** pull-down menu. If a table is missing, make sure you have imported the table into the project.

5. Optional: In the **Data field** field, select a column from the selected data source table. The columns displayed in this pull-down menu come from the data source you selected. If nothing appears, make sure a data source is selected.
6. Optional: In the **Link source** field, select a table definition from those that have been imported into the project. Only the tables imported into the project appear in the Link source pull-down menu. If a table is missing, make sure you imported the table into the project.
7. Optional: In the **Link field** field, select a field from the selected link source table. The fields displayed in this pull-down menu come from the link source you selected. If nothing appears, make sure a link source is selected.
8. Optional: In the **Usable** field, select USABLE to display the data to the user or UNUSABLE to hide it.
9. Optional: In the **Width** field, specify the width, by number of characters, for a column in the list. By default width equals the width set in the table. If the **Width** field is left blank, the column sizes automatically based on the content.
10. Optional: In the **Separator** field, enter symbols and spaces to separate information in each column. Enter the separator exactly as you want to see. Enter the separator on the row of the column you want it to appear after.
11. Optional: In the **Align** field, choose how you want the text aligned in its column. Select left, right, or center from the pull-down list.
12. Optional: Change the row order.
13. Click **OK**.

Related tasks:

"Creating and importing a table definition file" on page 44

Related examples:

"Example of a CREATE TABLE statement" on page 157
"Defining database queries" on page 157

Changing the row order

You must define data sources for a list control before you can change the row order.

These steps are part of the larger task of defining the data sources for a list. When you complete the steps for changing the row order, return to Defining the data sources for a list.

To change the row order::

1. In the List Data Sources window, Select a row and right-click on the row number to display a menu.
2. Choose the action you want:
   - **Insert new row** - Inserts a blank row above the selected row.
   - **Delete row** - Deletes the selected row.
   - **Move row up** - Moves the selected row up one row.
   - **Move row down** - Moves the selected row down one row.

Return to Defining the data sources for a list

Forms

Add forms to your application.

Related tasks:

"Controls" on page 50
"Tables and data sources" on page 43

Creating a new form

When you create a new project, a blank form is automatically created for you. There are a couple of ways to create additional forms.

The Form Creation wizard creates a form that includes labels and fields for each of the database columns selected. The wizard also allows you to add buttons or menu items with specific events associated with them. You can use the wizard to create a form that associates actions with a command button array (CBA), if the device you are targeting supports this option.

The controls created by the wizard can be modified outside of the wizard.

To create a new form using the wizard::

1. In the Project pane, click the table that contains the columns you want to appear on the new form.
2. Right-click on the table and click **Launch form creation wizard** to start the wizard. All columns and database operations are selected by default.
3. Optional: On the Database columns page, complete the following steps:
   - In the **Column** column, check the fields that you want to appear on the form. All columns are selected by default. The **Type** column shows the type of data that can be accepted in the field and on the form.
   - In the **Label** column, type the text that you want to appear as the label for each field.
4. Click **Next**.
5. Optional: On the Database Operations page, complete the following steps:
   - In the **Action** column, select the actions you want added to the form from the list of possible actions.
   - In the **Text** column, type the text to appear on the button labels.
   - In the **Command Button Array** column, select up to four database operations to be associated with the device's command button array. If you do not associate an operation with the CBA, the operation will be assigned to a menu. If there is no **Command Button Array** column, this option is not available for the current target device.
6. Click **Finish** to create the form.

To create a new form manually::

1. Right-click Forms in the Project pane.
2. Click **Add new form**. A new form appears in the Form pane. If the current target platform supports a Command Button Array (CBA), this control will be added to the new form in a fixed location. The CBA cannot be moved or deleted.

Related tasks:

"Tables and data sources" on page 43
"Adding a control to a form" on page 51
"Setting the application initial form"
"Resizing a form" on page 50
"Changing the zoom percentage of a form"
"Deleting a form" on page 50

Setting the application initial form

When an application is launched, it always goes to an initial form. The initial form is a starting point for the application. The order of the rest of the forms will be determined based on the form navigation used in the application.

To set the application initial form::

Perform one of the following tasks.

- Modify the properties of the target OS.
  1. Click <*Target OS*> in the Project pane.
  2. On the Properties page of the Properties and Events pane, select a form from the **Startup Form** field.
- Or, right-click a form and click **Set as initial form**.

Related tasks:

"Tables and data sources" on page 43
"Creating a new form" on page 48
"Resizing a form" on page 50
"Changing the zoom percentage of a form"
"Deleting a form" on page 50

Changing the zoom percentage of a form

The default form zoom, when a project is first opened, varies depending on which platform you are targeting. Each form can be individually resized.

To change the zoom percentage for a form::

1. Select a form in the Form pane.
2. Click the down arrow at the top of the form.
3. Select the desired form zoom percentage from the list.

Related tasks:

"Resizing a form"

Resizing a form

The default form size, when a project is first opened, varies depending on which platform you are targeting. Mobile Application Builder chooses the optimal default size for forms; however, each form can be individually resized.

To resize a form::

1. Select a form in the Form pane.
2. On the Properties page of the Properties and Events pane, specify the Width and Height properties for the form.

Related tasks:

"Changing the zoom percentage of a form" on page 49

Deleting a form

You can delete a form from a project.

To delete a form::

1. Right-click a form in the Form pane or the Project pane.
2. Click **Delete** from the pop-up menu. A confirmation window opens.
3. Click **OK** to close the window and delete the form.

Related tasks:

"Creating a new form" on page 48

Controls

The Palette contains controls to be used for constructing your application. Mobile Application Builder comes with a standard palette of controls.

For each control there are many attributes that can be customized to fit your applications needs. Most controls have a Name and Caption property. The Name property is a unique name for the control that is system-generated, and cannot be edited. The Name will not be visible to users. Use the Caption property for labeling of controls. The Caption is visible to users, except in the case of a Form bitmap control.

The controls that are available in the palette vary by target platform.

Related tasks:

"Creating a new form" on page 48

Related reference information:

"Control properties" on page 110

Adding a control to a form

The Palette contains controls to be used for constructing your application. Mobile Application Builder comes with a standard palette of controls.

To add a control to a form::

1. Click on the control icon in the Palette to select it. This loads the cursor with the control, and changes the cursor to a crosshair.
2. Move the cursor over the desired location on the form.
3. Click the form to drop the control. If a control is selected but you do not want to use it, click another control or click the same control again to de-select it.

Related tasks:

"Configuring control properties" on page 54
"Controls" on page 50

Related reference information:

"Control properties" on page 110

Deleting a control from a form

You can delete controls from a form.

To delete a control from a form::

1. Right-click a control in the Form pane.
2. Click **Delete**, or click **Edit -> Delete** from the main menu.

Related tasks:

“Adding a control to a form” on page 51

Related reference information:

“Control properties” on page 110

Aligning controls

Use the Align feature to move one or more controls based on the position of another control. The control selected first is used as an anchor and the other selected controls are aligned based on the anchor’s position. The anchor control has solid color handles and all other selected control handles are white. Refer to the chart below for descriptions of how each alignment feature works.

To align controls::

1. Click the control to serve as an anchor.
2. Hold the Ctrl key down and click another control to align to the anchor control. To select multiple controls, hold the Ctrl key while clicking the controls you want to select.
3. Click **Format —> Align —>** *direction* from the main menu, use the icons on the toolbar, or right-click on the selected items you want to align and choose from the list that appears.

Table 12. Align options. Description

Left		Aligns the selected controls with the left edge of the anchor control.
Center		Aligns the selected controls along the vertical axis of the anchor control.
Right		Aligns the selected controls with the right edge of the anchor control.
Top		Aligns the selected controls with the top edge of the anchor control.
Middle		Aligns the selected controls along the horizontal axis of the anchor control.
Bottom		Aligns the selected controls with the bottom edge of the anchor control.

Related tasks:

“Moving controls” on page 53
“Re–ordering controls” on page 54

Related reference information:

“Control properties” on page 110

Moving controls

You can move individual or multiple controls on a form by using nudge positioning. You can also move a control by changing the properties of the control on the Properties page of the Properties and Events pane.

To move a control using nudge positioning::

1. Select one or more controls on a form.
2. Use the arrow keys, on the keyboard, to move one or more selected controls one pixel at a time.

To move a control by changing its properties::

1. Select a control.
2. On the Properties page of the Properties and Events pane, adjust the Left property to specify the distance from the left edge of the control to the screen edge. Adjust the Top property to specify the distance from the top edge of the control to the screen edge.

Related tasks:

"Aligning controls" on page 52
"Re–ordering controls" on page 54
"Re–sizing controls"

Re–sizing controls

When you select a control in the Form pane, sizing handles appear at the corners and along the edges of the control. You can resize a control by dragging its sizing handles, or you can resize it more precisely by specifying the control's Height and Width properties.

To re–size a control using sizing handles::

1. Click the control that you want to resize in the Form pane. Sizing handles appear at the corners and edges of the control.
2. Drag the sizing handles to change the size of the control.

To re–size a control by adjusting its properties::

1. Select the control that you want to resize.
2. On the Properties page of the Properties and Events pane, specify the width of the control in the **Width** field, and specify the height of the control in the **Height** field.

Related tasks:

"Moving controls" on page 53
"Re–ordering controls"

Related reference information:

"Control properties" on page 110

Re–ordering controls

Use the Order feature to move one control to the front or back of the form. Order is more obvious if controls are layered. Refer to the chart below for descriptions of how each order feature behaves.

To re–order controls::

1. Click the control that you want to move. This feature works only when one control is selected.
2. Click **Format —> Order —>** *direction* from the main menu, use the icons on the toolbar, or right-click on the selected items to order and choose from the list that appears.

Table 13. Order options. Description

Bring to Front		Moves control to the front of the form.
Send to Back		Moves control to the back of the form.

Related tasks:

"Moving controls" on page 53
"Re–ordering controls"

Related reference information:

"Control properties" on page 110

Configuring control properties

You must add a control to a form before you can configure control properties.

Most control properties already have default values set. For check boxes, fields, and lists, the Data source and Data field properties must be defined, to read and write to a specific table. For form bitmap controls, a Bitmap resource name must be defined.

To configure control properties::

1. Select the control in the Form pane or the Project pane.

2. Modify control properties on the Properties page of the Properties and Events pane. See “Control properties” on page 110 for detailed explanation of each control property.

Related tasks:

“Events, actions, and targets” on page 70
“Configuring device control properties”

Related reference information:

“Control properties” on page 110

Configuring device control properties

Mobile Application Builder provides support for many application controls. Each control has properties associated with it. Device control configuration allows you to change the default values for each control’s property. The changes affect all controls in Mobile Application Builder, and are not limited to a project.

To configure control properties for a device::

1. Right-click <*Target OS*> in the Project pane, and click **Open device configuration**.
2. Expand the Palettes node to see the available Control Palette options.
3. Expand a Control Palette node to see the available controls.
4. Click the name of a control, and modify the default properties in the right panel.
5. Click **File —> Close**. Your changes are applied.

Related tasks:

“Configuring control properties” on page 54

Related reference information:

“Control properties” on page 110

Changing the font for a control

You can change the font that will be used to display the caption of a control in an application. You can set the font at an application level, and you can set the font for individual controls. Default fonts are used unless you specify a different font.

Changes to font settings work in the following ways:

- If you make changes to an individual control using the control’s property settings, the changes are applied to that control only.

- If you make changes to the default settings in the application preferences, the changes are applied to all controls, *except* any controls that were previously changed individually using the control's properties.
- You can reset controls that you have changed individually back to the preference settings.

Font support varies by target platform.

To change the default font settings at an application level::

1. Click **File —> Preferences**. The Preferences window opens.
2. Expand the **Application** node to see application preferences.
3. Click the Font node. In the pane on the right, specify font settings, or click **Restore Defaults** to return to default Mobile Application Builder settings. Setting options vary by target platform.
4. Click **OK**. The changes you made are applied to all controls, *except* any controls that you have previously changed using the control's properties.

To change the font for an individual control::

1. Click the control that you want to change the font for.
2. On the Properties page of the Properties and Events pane, perform one of the following tasks. The options available vary by target platform.
   - Select an option in the **Font** field. This option is available if the target platform is Palm OS.
   - Or,
     - Click the … to open the Font Options window. If there is no … in the Font field, this option is not available for the current target platform.
     - Optional: Specify a font in the **Font** field.
     - Optional: Specify a style in the **Style** field.
     - Optional: Specify a size in the **Size** field.
     - Click **OK**. The Font Options window closes and your selections are applied to the control.

Related tasks:

"Configuring control properties" on page 54

Changing the color for a control

You can change the colors that will be used to display a control in an application. You can set the color at an application level, and you can set the color for individual controls. Default colors are used unless you specify a different color. Support for color options varies by target platform.

Changes to color settings work in the following ways:

- If you make changes to an individual control using the control's property settings, the changes are applied to that control only.

- If you make changes to the default settings in the application preferences, the changes are applied to all controls, *except* any controls that were previously changed individually using the control's properties.
- You can reset controls that you have changed individually back to the preference settings.

Color support varies by target platform.

To change the default color settings at an application level::

1. Click **File —> Preferences**. The Preferences window opens.
2. Expand the **Application** node to see application preferences.
3. Expand the UI Color node, then click the control or resource that you want to set color options for.
4. In the pane on the right, specify color settings, or click **Restore Defaults** to return to default Mobile Application Builder settings. Setting options vary by target platform.
5. Click **OK**. The changes you made are applied to all controls, *except* any controls that you have previously changed using the control's properties.

To change the color for an individual control::

1. Click the control that you want to change the color for.
2. On the Properties page of the Properties and Events pane, perform the following task. The options available vary by target platform.
   - Click the … next to the **Background** or **Foreground** property to open the Color Options window. If there is no **Background** or **Foreground** property, this option is not available for the current target platform.
   - In the Color Options window, specify a color, or click **Use Default** to return to the project default color settings. Click **OK**.

Related tasks:

"Configuring control properties" on page 54

Enabling Palm five-way navigation

You can enable five-way navigation for Palm applications. This allows users to move among controls on a form using the five-way navigation button on a Palm device.

For MAB-generated applications, navigation is mapped as follows:

Button direction	Action
Right	Move to next field
Left	Move to previous field
Up	Move to previous list item, if a list is on the current form.

Button direction	Action
Down	Move to next list item, if a list is on the current form.
Center	Select the highlighted item in a list. This is equivalent to a Tap event in a list.

If there are multiple lists on a form, the first list is enabled for navigation actions when the form is opened. If the user Taps on another list, that list is enabled for navigation actions.

Prerequisites:

In order to enable this function, you must install the Palm SG SDK, as described in "Setting up the Palm C code building environment" on page 12. If the Palm SG SDK is not installed properly, you will receive an installation-related error when you build the application.

To enable five-way navigation in a Palm application::

1. Open a Palm project.
2. Click **File —> Preferences**. The Preferences window opens.
3. Click the Build node to see build preferences.
4. Select the **Enable Five Way Navigation** check box.
5. Click **OK**.

Related tasks:

"Setting build preferences" on page 38

Using pop-up triggers

A pop-up trigger allows the user to make a selection from multiple choices that drop down in a list when the control is tapped. The contents of the list are defined by linking a list control with the pop-up trigger. The order of actions starts when the pop-up trigger is clicked. Next the list is displayed and the user makes a selection. Before the pop-up trigger label changes to display the selection all events associated with the control are executed.

Support for this control varies by target platform.

To use a pop-up trigger::

1. Add a list control to the form you want the pop-up trigger on.
2. Set the following list properties:
   - Define the data sources or list items. These are the choices that are available when the pop-up trigger is clicked.

- The Visible items property needs to be set to the number of items that will appear in the pop-up.
- Set the usable property to NONUSABLE. The list is not visible until it is called by the pop-up trigger.

3. Add a pop-up trigger to the same form. Typically the pop-up trigger is positioned above the List. When the pop-up trigger is clicked the list becomes visible to the user.
4. Set the following pop-up trigger properties:
   - Specify the caption that will appear by default on the pop-up trigger.
   - Specify the name of the list created above next to the ListID property.

Related tasks:

"Configuring control properties" on page 54

Related reference information:

"Pop-up trigger properties" on page 123

Creating radio buttons

Radio buttons can be created for Java platforms by configuring the **Group ID** and **Checked** properties for two or more check box controls.

Support for this control varies by target platform.

To create radio buttons::

1. Add two or more check box controls to the form you want the radio buttons on.
2. Set the **Group ID** property for all check box controls to the same value, with the following restrictions:
   - You must set a value other than blank.
   - The Group ID must be a Java group name with no spaces.
   - The first character must be alphanumeric.
3. Set the **Checked** property to `True` for the control you want to appear as selected by default, and set the **Checked** property for all other controls to `False`. When you build and test the application, these controls are rendered as radio buttons, with the default control that you specified selected.

Related tasks:

"Configuring control properties" on page 54

Related reference information:

"Check box properties" on page 114

Defining the order that fields display in a list

Use the Order By window to define the order that fields will display in your list.

You must define data sources for a list control before you can define the order that fields will display in the list.

To define the order that fields display in a list::

1. Select a list control.
2. On the the Properties page of the Properties and Events pane, click ... next to the **Order By** field to open the Order By window. You must define data sources for the list before you can open the Order By window.
3. In the **Data Source** column, select a table definition from those that have been defined as data sources for the list control. If a table is missing, make sure you have defined the data sources for the list.
4. In the **Data Field** column, select a column from the selected data source table. The columns displayed come from the data source you selected in the previous step. If nothing appears, make sure a data source is selected.
5. In the **Order** field, select the order in which you want data to appear. You can select `ASCENDING` or `DESCENDING` order.
6. Click **OK**. The Order By window closes.

Related tasks:

"Adding a control to a form" on page 51

Related reference information:

"List properties" on page 121

Associating a unique value with a field control

Mobile Application Builder can optionally generate a unique ID that can be associated with an ID field, such as CustomerID or MedicalRecordID. This ID will be generated as a unique number that is only used by the database for record tracking purposes, and does not necessarily need to be accessed by the end user.

Unique IDs consist of a minimum of a timestamp, which requires 26 characters. You can further customize the unique ID to contain a prefix string, all or a portion of the current user ID, or a postfix string, depending on your needs and the available characters in the field.

If you want to add a user ID to the unique value, you must first ensure that the user ID information is available by allowing the user to enter the information *before* the form that contains the field control with the unique ID is opened. Then, you must associate the user ID with a global variable that Mobile Application Builder generates automatically for this purpose.

For Palm C code applications, Mobile Application Builder generates a global variable called `MAB_USERID`. For all Java code platforms, the name of this global variable is `MABGlobalScripts.MAB_USERID`.

Field controls that meet the following requirements can be associated with a unique ID.

- The column that is specified as the data field for the control is of the data type VARCHAR or CHAR.
- The maximum number of characters available for the field is 26 or greater.

To associate a unique value with a field control::

1. Select a field control.
2. On the Properties page of the Properties and Events pane, select a table from the **Data Source** list. If a table is missing, make sure you have created and imported a table definition into your project.
3. Select a column from the **Data Field** list. The data type of the column you selected is shown in the Data Field Type field. The column must have a data type of VARCHAR or CHAR. The maximum number of characters available is shown in the Max Chars field. This value must be 26 or greater.
4. Click [...] next to the Use Unique Value field to open a window where you can specify the characteristics of the unique value.
5. Select the **Set unique value to this field** option.
6. Optional: In the Enter prefix string field, type a string to add to the beginning of the unique value.
7. Optional: Specify a number in the next field to add all or a portion of the current user ID information to the unique ID. The number specifies how many characters of the current user ID will be added to the unique value.
8. Optional: In the Enter postfix string field, type a string to add to the end of the unique value.
9. Click **OK**. If the parameters for the unique value are valid, the window closes and your changes are applied. If the parameters are invalid, a message window will inform you that you need to adjust your settings.

Related tasks:

"Creating and importing a table definition file" on page 44
"Defining a global variable" on page 76

Related reference information:

"Field properties" on page 117

Project resources

Mobile Application Builder provides many resources to use in your application. Set properties and events for the resources and use them in scripts.

The resources available for your application vary by target platform.

Related tasks:

“MAB projects” on page 31

Related reference information:

“Resource properties” on page 128

Creating a new alert

Alerts are used to prompt a user for a response. An alert must be acknowledged by the user before they can proceed. Use scripting to reference the alert.

Support for alerts and alert properties varies by target platform.

To create a new Palm alert::

1. Right-click ⚠ Alerts in the Project pane. If there is no Alert node in the Project pane, alerts are not supported for the current target platform.
2. Click **Add new Alert**. A new alert appears under Alerts in the Project pane.
3. Click the new alert in the Project pane.
4. On the Properties page of the Properties and Events pane, enter a Caption, Alert Type (for Palm applications only), Message, Button text, and Default Button ID (for Palm applications only).
5. Click the form or control you want the alert associated with and click **Selected —> Create a new script**. A script window opens.
6. Enter the following script in the script window. Replace ***alertName*** with the actual system-generated name for the alert.

```
FrmAlert(alertName);
```

7. Click **File —>** 💾 **Save Script**.
8. Click **File —> Close**. The script window closes, and the script is added as an event to the form or control you created the script under.

To create a new alert for Java platforms::

You can create a custom alert, or you can use standard alerts.

- To create a custom alert:
  1. Right-click ⚠ Alerts in the Project pane. If there is no Alert node in the Project pane, alerts are not supported for the current target platform.
  2. Click **Add new Alert**. A new alert appears under Alerts in the Project pane.
  3. Click the new alert in the Project pane.
  4. On the Properties page of the Properties and Events pane, enter a Caption, Message, and Button text.
  5. Click the form or control you want the alert associated with and click **Selected —> Create a new script**. A script window opens.

6. Enter the following script in the script window. Replace ***alertName*** with the actual system-generated name for the alert. The value returned by the showAlert method is the index of the button on the alert which is clicked, the first button having an index of 0.

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

7. Click **File —>** 💾 **Save Script**.
8. Click **File —> Close**. The script window closes, and the script is added as an event to the form or control you created the script under.

- To use a standard alert:
  1. Click the form or control you want the alert associated with and click **Selected —> Create a new script**. A script window opens.
  2. Enter the following script in the script window, customizing italic bold text as follows:

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

where

- *alertType* and *buttonOption* are integers
- *title* and *message* are strings

alertType	Specify one of the following alert types. This will be used to determine the image shown on the alert: • CONFORMATION_ALERT • INFORMATION_ALERT • ERROR_ALERT • WARNING_ALERT
buttonOption	Specify one of the following options for the buttons and text that will appear on the alert: • DEFAULT_OPTION • OK_CANCEL_OPTION • YES_NO_CANCEL_OPTION • YES_NO_OPTION
String title	Specify the title of the alert.
String message	Specify the message text that will appear on the alert.

Related concepts:

"Scripting support" on page 136

Related tasks:

"Scripting in Mobile Application Builder" on page 74

Related reference information:

"Alert properties" on page 130

Creating and displaying bitmaps

You can display bitmaps in your application on forms or button controls. A bitmap image can be created using any external drawing program that allows you to save bitmap images in black and white or direct color format (supported in Palm OS 4.0 or higher). Store bitmaps in your application's project directory. Display a bitmap by adding a form bitmap control to a form, or display a bitmap on a button, push button, or repeating button by configuring the control's properties.

The display characteristics of a bitmap will vary depending on the screen resolution of your target device. For example, if you supply a 16-bit bitmap and the target device only supports up to 4-bit bitmaps, the image will display as a 4-bit bitmap. This may have a negative effect on the appearance of the bitmap. Always test bitmaps on an emulator or mobile device.

Support for bitmaps varies by target platform.

Mobile Application Builder bitmap tasks::

"Creating a new bitmap"
"Displaying a bitmap on a form" on page 65
"Displaying a bitmap on a control" on page 65

Related reference information:

"Bitmap properties" on page 132
"Form bitmap properties" on page 111

Creating a new bitmap

Before you can display a bitmap on a form or control, you must create the bitmap, store it in the project directory, then add a bitmap resource to your project.

Support for bitmaps varies by target platform.

To create a new bitmap::

1. Right-click Bitmaps, in the Project pane. If there is no Bitmap node in the Project pane, bitmaps are not supported for the current target platform.
2. Click **Add new Bitmap**. A new bitmap appears under Bitmaps.
3. Select the new bitmap in the Project pane.
4. On the Properties tab in the Properties and Events pane, type the name of the bitmap file in the **Bitmap File Name** field. The bitmap file should be in the current project directory.

Related tasks:

“Displaying a bitmap on a form”
“Displaying a bitmap on a control”

Related reference information:

“Bitmap properties” on page 132
“Form bitmap properties” on page 111

Displaying a bitmap on a form

Before you can display a bitmap on a form, you must create the bitmap and add it as a resource to your project.

Support for bitmaps varies by target platform.

To display a bitmap on a form::

1. Select a form.
2. Add a form bitmap control to the form from the Control Palette.
3. On the Properties page in the Properties and Events pane, select a bitmap from the **Bitmap Resource Name** list. If there are no bitmaps available in the **Bitmap Resource Name** list, make sure you have created a bitmap and added it as a resource to your project.

Related tasks:

“Creating a new bitmap” on page 64
“Displaying a bitmap on a control”

Related reference information:

“Bitmap properties” on page 132
“Form bitmap properties” on page 111

Displaying a bitmap on a control

You can display a bitmap on a button, push button, or repeating button. Before you can display a bitmap on a control, you must create the bitmap and store it in the project directory. You must specify a bitmap for both the selected and unselected state of the control or it will not display properly. These bitmaps can be the same or different.

Support for bitmaps varies by target platform.

To display a bitmap on a control::

1. Select a button, push button, or repeating button control.
2. On the Properties page in the Properties and Events pane, select a bitmap from the **Bitmap** list. If there are no bitmaps available in the **Bitmap** list, make sure you have created a bitmap and stored it in the project directory. This property will designate which bitmap will be displayed on the control while it is in an unselected state.

3. On the Properties page in the Properties and Events pane, select a bitmap from the **Selected Bitmap** list. If there are no bitmaps available in the **Bitmap** list, make sure you have created a bitmap and stored it in the project directory. This property will designated which bitmap will be displayed on the control while it is in a selected state. You can choose to specify the same bitmap that you specified for the **Bitmap** property.

Related tasks:

"Displaying a bitmap on a form" on page 65
"Creating a new bitmap" on page 64

Related reference information:

"Bitmap properties" on page 132
"Form bitmap properties" on page 111

Creating a new icon

Each Palm application has an icon that is displayed in the Palm application pane.

Mobile Application Builder provides a default icon or you can create one of your own, with an external icon editor of your choice. Some target platforms require an icon and a mask.

Support for icons varies by target platform.

The display characteristics of an icon will vary depending on the screen resolution of your target device. Always test icons on an emulator or mobile device.

When you create an icon, use the following guidelines:

For Palm applications (icon only):

- The icon size should be 32x32, 32x22, or 22x22 pixels, in .bmp format. Other sizes may not display properly.
- The icon can be in color or black and white format.

For SymbianOS/Crystal applications (icon and mask):

- The icon size should be either 64x50 and 25x20 pixels, in .bmp format. Color depth is 16 bit.
- The icon can be in color or black and white format.
- The mask should be white for the transparent areas where the background behind the icon should show through, and black for all other regions.

For SymbianOS/Quartz applications (icon and mask):

- The icon size should be either 32x32 or 20x16 pixels, in .bmp format. Color depth is 16 bit.
- The icon can be in color or black and white format.
- The mask should be white for the transparent areas where the background behind the icon should show through, and black for all other regions.

To create an icon::

1. Click Icons, in the Project pane.
2. In the Icon File Name on the Properties page of the Properties and Events pane, enter a path and file name for the icon. For Symbian platforms, enter multiple file names for icons and masks separated by a space.

Related concepts:

"Scripting support" on page 136

Related tasks:

"Scripting in Mobile Application Builder" on page 74

Related reference information:

"Icon properties" on page 132

Creating a new menu item

Menus or menu items are the individual functions that are accessible from the menubar and are not found anywhere else on the screen.

Menus appear in a drop-down list off of a menubar topic. Menu items are only usable when they are associated with a menubar. After you have created a menu, create a menubar and associate the menu with it.

To create a new menu item::

1. Right-click **Menus**, in the Project pane.
2. Click **Add new Menu**. A new menu appears under Menus.
3. On the Properties page in the Properties and Events pane, type the text and shortcut character for the menu item in the **Menu Item Text** and **Accel Char** fields.
4. Identify an event and action to execute when the menu item is selected in your application.
   a. Click the Events tab in the Properties and Events pane.
   b. Select the event type on the left. See Event descriptions for more information.
   c. Click the ... to the right of an event description. The Event Action and Target Selection window opens.
   d. In the Event Action and Target Selection window, select the action that you want from the **Action** list. See Action descriptions for more information. An action is performed when a control's event executes. The **Action** list displays possible actions associated with the selected event.
   e. Select a corresponding target for the action from the **Target** list. A target is a table, form, or control you want the action to be performed on. The **Target** list displays possible targets to choose from based on the action selected.
5. Create a menubar to display the menu items.

Related tasks:

"Creating a new menubar"

Related reference information:

"Menu properties" on page 131
"Menubar properties" on page 131

Creating a new menubar

Menubars provide access to additional functions (menu items) that are not available anywhere else on the screen. The menubar is available when the Palm silk-screened Menu button is tapped. Each form in your application can have multiple menubars.

Menubars contain topics that drop-down a list and display menu items. Menu items must be created first to be associated with a menubar. Once a menubar is created it must be associated with a specific form for it to be available in your application.

To create a new menubar::

1. Right-click MenuBars, in the Project pane.
2. Click **Add new Menubar**. A new menubar appears under Menubars.
3. Go to the new menubar's properties and type the text for the menubar drop-down in the **PullDowns** field.
4. Enter the names of the menu items you want to appear under the menubar drop-down next to the Menu Item IDs property.
5. To add or delete Menu Item IDs, right-click the Menu Item ID, and click **Add MenuItem**, or click **Delete** from the pop-up list.
6. To add additional drop-down menus:
   a. Right-click **PullDowns**, and click **Add PullDown**. An additional PullDowns item appears.
   b. Type the text for the new drop-down menu.
   c. To add a menu item to this drop-down menu, right-click the last existing PullDown item, and click **Add MenuItem**.
   d. Enter the names of the menu items you want to appear under the new Menubar drop-down next to the Menu Item IDs property.
7. Activate the menubar by associating it with a form:
   - Select a form.
   - On the Properties page of the Properties and Events pane, select a menubar from the list in the Menu ID property.

Related tasks:

"Creating a new menu item" on page 67

Related reference information:

"Menu properties" on page 131
"Menubar properties" on page 131

Creating a new string

Strings are used to display a string of characters. Use strings to display a message about the application, such as version number. Use scripting to reference the string.

Support for strings varies by target platform.

To create a new string::

1. Right-click Strings in the Project pane. If there is no String node in the Project pane, strings are not supported for the current target platform.
2. Click **Add new String**. A new string appears under Strings in the Project pane.
3. Click the new string in the Project pane.
4. On the Properties page of the Properties and Events pane, type the string text in the **String** field, or enter the name of the file containing the text.
5. Add a field to a form to display the string.
6. Select the field, and click **Selected —> Create a new script**. A script window opens.
7. Enter the sample script below. Customize the italicized bold text to work with your application. Paste Part is required to set the text to display in the field. See comments in code below.

```
CharPtr startString;
 VoidHand rscH;
/* retrieve a String resource */
 rscH = DmGetResource (strRsc, stringName);
 startString = MemHandleLock (rscH);
/* use paste part */
/* select display field String and set text */
 /* change CharPtr to same as defined above */
 PAB_setFieldDrawPtr(fieldName, startString);
 /* release String resource */
MemHandleUnlock(rscH);
 DmReleaseResource(rscH);
```

8. Click **File —> Save Script**.
9. Click **File —> Close**. The script window closes, and the script is added as a Tap event to the form or control you created the script under.

Related concepts:

"Scripting support" on page 136

Related tasks:

"Scripting in Mobile Application Builder" on page 74

Related reference information:

"String properties" on page 130

Events, actions, and targets

Events are actions in your application that are triggered by pen strokes or key presses. Forms, controls, and resources have many events available to be used with them. For example, to run a script and go to another form when a button control is pushed is two events. Events occur in the order in which they are listed.

An action is performed when a control's event executes. The action field in the Event Action and Target Selection window displays a list of possible actions associated with the selected event. Select an action, from the list, to take place when an event executes. For example, a Tap event on a button control can have an associated action (Show) to open another form.

A target is a table, form, dialog, or control that you want the action to be performed on. The target field displays a list of possible targets to choose from based on the action selected. For example, the Show action above needs to have a target form associated with it.

Related tasks:

"Controls" on page 50

Related reference information:

"Event descriptions" on page 133
"Action descriptions" on page 135

Configuring control events

You must add a control to a form before you can configure control events.

To configure control events::

1. Select a control in the Form pane or the Project pane.
2. Click the Events tab in the Properties and Events pane.
3. Select the event type on the left. See Event descriptions for more information.
4. Click the … to the right of an event description. The Event Action and Target Selection window opens.
5. Select the action that you want from the **Action** list. See Action descriptions for more information.
6. Optional: Click **Options** to specify options for certain actions. If the **Options** button is not available, there are no options available for the action you selected.
7. Select a corresponding target for the action from the **Target** list. The **Target** list displays possible targets to choose from based on the action selected.

8. Optional: For a Create record, Update record, or Delete record action, add a confirmation dialog to the event.
9. Click **OK**. The action and target you specified is added to the event.
10. Optional: Right-click the Events page of the Properties and Events pane and click **Add new event connection** from the pop-up menu to add additional events.

Related tasks:

"Adding a control to a form" on page 51

Related reference information:

"Control properties" on page 110
"Event descriptions" on page 133
"Action descriptions" on page 135

Adding a confirmation dialog for a database action

You must configure control events before you can add a confirmation dialog for a Create record, Update record, or Delete record action.

Choose a default message or customize your own message. Adding a dialog to an event action overrides any confirmation dialog settings set in Preferences.

These steps are part of the larger task of configuring control events. When you complete the steps for adding a confirmation dialog for a Create record, Update record, or Delete record action, return to Configuring control events.

To add a confirmation dialog for a Create record, Update record, or Delete record action::

1. Select a Create record, Update record or Delete record action from the Event Action and Target Selection window.
2. Click **Set Message...** at the bottom of the Event Action and Target Selection window to open the Confirmation and Response window.
3. In the Confirmation and Response window, define the confirmation dialog characteristics for the selected action.
   - Select **Message** to activate a dialog before or after the selected action.
   - Click in the field next to the **Message** check box to select from a list of messages, or type in customized message text.
4. Click **OK**.

Return to Configuring control events

Reordering control events

You can change the order in which events for a control are executed.

To reorder control events::

1. Select a control in the Form pane or the Project pane.
2. Click the Events tab in the Properties and Events pane.
3. Right-click an event, and click **Reorder Events**.
4. In the window that opens, click an event, and click the up or down arrows to move the event up or down in the list.
5. Click **OK**.

Related tasks:

"Adding a control to a form" on page 51

Related reference information:

"Control properties" on page 110
"Event descriptions" on page 133
"Action descriptions" on page 135

Adding an event to a control to display a database preferences dialog

Mobile Application Builder supports DB2 Everyplace databases that reside on secondary storage devices such as Compact Flash, IBM Microdrive, or Palm Expansion Card slot. Encrypted DB2 Everyplace databases are also supported. Mobile Application Builder will generate the necessary code to display a dialog that will allow the user to select the location of the database and enter a user ID and password before attempting to access encrypted data, from the running application. You can manually choose to have this dialog accessed from an application menu, button, or any control that can have an associated event action. This dialog can be configured to contain only the database location , or both the database location and user ID and password fields. To access data on a secondary storage device, use the dialog with only the database location field. To access encrypted data, use the dialog with all fields.

For Palm applications, the location dialog contains a drop-down list that users can use to select either Device RAM (primary storage) or secondary storage cards. If the user selects a secondary storage card, he must also enter a directory path in an additional field indicating which directory contains the database tables. In this field, the user should type the directory path, using forward slashes, without a starting slash. Ending slashes are acceptable, but not mandatory. For example:

- Correct location entry: `db2e/database/tables`
- Correct location entry: `db2e/database/tables/`
- Incorrect location entry: `/db2e/database/tables`

In some cases, this location dialog will automatically be generated by Mobile Application Builder. For example, if an application accesses encrypted data, and valid user ID and password information has not previously been saved, the dialog will come up automatically so the user can enter the appropriate information.

The user ID and password information, once entered, is stored within the application until the application is closed. The user can also choose to have the password information saved indefinitely, by selecting a check box on the dialog. If this option is not selected, the password information is destroyed when the application is closed, and the user ID is saved as the default user ID.

If the user changes the database location to a location that is different from the current connection location, the user will be prompted to restart the application so a new connection can be established. This will ensure the consistency of the data already obtained in the application.

To add an event to a control to display a database preferences dialog::

1. Select the control to which you want to add the dialog.
2. Click the Events tab in the Properties and Events pane.
3. Click ... on the right to open the Event Action and Target Selection window if there are no existing actions associated with this control. Or, right-click on the last event shown and click **Add new event connection**, then click ... .
4. Click Show in the Action list.
5. In the Target list, choose one of the following options.
   - To generate a dialog containing database location and user ID and password fields, click Database Authentication Dialog .
   - To generate a dialog containing only database location fields, click Database Location Dialog.

Related tasks:

"Adding a control to a form" on page 51

Related reference information:

"Control properties" on page 110
"Event descriptions" on page 133
"Action descriptions" on page 135

Adding a physical delete option to a Delete record action

You can add an option to a Delete record action that will cause the record deletion to occur on the client (mobile) database only. This option is applicable for users who want to delete a record from the mobile database, but do not want to reflect the delete action on the server (enterprise) database when they synchronize.

To add a physical delete option to a Delete record action::

1. Select the control to which you want to add the delete action.
2. Click the Events tab in the Properties and Events pane.

3. Click ... on the right to open the Event Action and Target Selection window if there are no existing actions associated with this control. Or, right-click on the last event shown and click **Add new event connection**, then click ... .
4. Click Delete record in the Action list.
5. Click **Options**. An Options window opens.
6. Select the **Delete record only from the device (physical delete)** check box.
7. Click **OK**. The Options window closes.

Related reference information:

"Control properties" on page 110
"Event descriptions" on page 133
"Action descriptions" on page 135

Scripting in Mobile Application Builder

Mobile Application Builder generates many code functions associated with some of the resources and controls. In addition, you can use the integrated scripting function to add many of your own functions. The scripting function allows you to write your own C code using the mobile device's API calls.

Related concepts:

"Scripting support" on page 136

Related tasks:

"Setting scripting preferences" on page 40

Accessing MAB-generated code using Java scripts

This topic describes how to extend the default behavior of a generated Java application using scripting. The developer is responsible for syntax and semantic errors as well as the logic in the scripts written using the following information.

Generated MAB objects:

The following table describes how MAB objects are generated, and how they can be accessed in scripts.

Table 14. Generated MAB objects

MAB object	Generated as	Type	Accessing in scripts
host variables	static members of the class *MABGlobalScripts*	*java.lang.String*	Can be accessed as `MABGlobalScripts.hv` where *hv* is a host variable.

Table 14. Generated MAB objects (continued)

MAB object	Generated as	Type	Accessing in scripts
global scripts	member methods of the class *MABGlobalScripts*	method signature is defined by the user	Can be accessed as `new MABGlobalScripts().gs()` where *gs()* is a global script.
forms	• Generated as a class with the same name as the value of the form's **Name** property in MAB. • All forms are created and added to a *Hashtable* named *panelTable* in the *MABAppFrame* class. Each of the classes representing a form has a variable named *parent*, which refers to *MABAppFrame*	*java.awt.Panel*	Text in a field *Field1Form1* , in *Form1* can be accessed from a script on a button on *Form2* (or any form) as `((Form1)parent.panelTable .get("Form1")) .Field1Form1.getText();`
controls	members of the class with the same as the value of the **Name** property of the control in MAB	list control: *java.awt.List* field control: *java.awt.TextField*	The scripts defined on a form (or any control on a form) can access the controls on that form by referring to their name. For example, in a script on a button , you can get the text from the field *Field1Form1* as *Field1Form1.getText()*
scripts defined on a form, or scripts defined on any control on the form	member functions of the class for the form		First script on *Button1Form1* is generated as `void Button1Form1Script1 (ANTEvent ae)`

Accessing data:

During application execution, when a row is selected from a table, the values of the columns in the row are stored in an internal buffer. This buffer is updated with each data fetch from DB2 Everyplace. Therefore, for any column, the buffer will have the most recently selected value for that column.

This buffer is implemented in the *DataStore* class, and the *getValue()* method will return the current value of the column. For example: *Object getValue(String key)*, where *key* is the string obtained by concatenating the table name and the column name. The returned value is the most recent value selected for the column, or *null* if a row including the column has not yet been fetched from the database.

So, from a script on any form, the current value of *Column1* of *Table1* can be obtained as *parent.datastore.getValue("Table1Column1"))*.

The database connection in the application is generated as a *public static* variable named *con* in the *MABDataAccess* class and can be accessed from scripts as *MABDataAccess.con*.

Related concepts:

"Scripting support for Java code platforms" on page 136

Related tasks:

"Scripting in Mobile Application Builder" on page 74

Creating a global script

Global scripts allow you to create general functions that can be called by other scripts. Global scripts are recommended if you want to use a script in more than one place. Global scripts can be called only from event scripts.

New global scripts are added under Global Definitions in the Project pane and are available throughout the project.

To create a global script::

1. Click **Global Definitions** in the Project pane.
2. Click **Selected —> Create a new script** from the main menu. A script window opens.
3. Add your script. Leave the return value and any parameters as is.
4. Optional: Add helper functions to the script window.
5. Optional: Paste a control into the script window.
6. Click the 💾 Save button and close the script window.

Related concepts:

"Scripting support" on page 136

Related tasks:

"Setting scripting preferences" on page 40
"Creating an event script" on page 77
"Adding helper functions to the script window" on page 79
"Pasting a control into the script window" on page 80

Defining a global variable

Use global variables to define variables that you want visible to all scripts. These variables can be called by global and event scripts. Global variables is one window and one file that all global variables for the project are stored in.

Support for global variables varies by target platform.

To define a global variable::

1. Click **Global Definitions** in the Project pane.
2. Click ... in the Properties page in the Properties and Events pane next to Global Variables to open the Global Variables window.
3. Define your application's global variables.
4. Click **OK** to close the Global Variables window.

Related concepts:

"Scripting support" on page 136

Related tasks:

"Setting scripting preferences" on page 40

Editing or deleting a global script

All global scripts are located under **Global Definitions** in the Project pane.

To edit or delete a global script::

1. Select the global script you want to edit in the Project pane.
2. Click **Selected —> Edit script** or **Selected —> Delete script** from the main menu.

Related concepts:

"Scripting support" on page 136

Related tasks:

"Setting scripting preferences" on page 40
"Creating a global script" on page 76

Creating an event script

Event scripts allow you to run a specific script when a certain event occurs, like on a button click or form initialization. Follow the steps below to create an event script. It is recommended that you define scripts under Global Definitions and then call those scripts from event scripts.

All event scripts are associated with specific forms and controls. If you want a script to be available for all forms or controls, create a global script.

To create an event script::

1. Select either a form or a control.
2. Select **Selected -> Create a new script** from the main menu. A script window opens. Depending on your target platform, you will either see a C function skeleton or a standard Java signature already in the window. The C function definition at the top of the script window will include the name of the control you are adding script to. Characters in this name that are not in the allowable character set for the GNU C-compiler are changed to an underscore character.
3. Add your script. For C code scripts, leave the return value and any parameters as is.
4. Optional: For C code scripting, Add helper functions to the script window.
5. Optional: For C code scripting, Paste a control into the script window.
6. Optional: Rename the event script. If you do not rename an event script it is automatically named for you.
7. Click the 💾 Save button and close the script window.
8. Set the event that will cause the script to be invoked.
   a. With the form or control still selected, click the Events tab in the Properties and Events pane. Your new script appears in the list of events.
   b. Select an event from the list of events in the left-hand column next to the new script.

Related concepts:

"Scripting support" on page 136

Related tasks:

"Setting scripting preferences" on page 40
"Creating a global script" on page 76
"Editing or deleting an event script"
"Renaming an event script" on page 79
"Adding helper functions to the script window" on page 79
"Pasting a control into the script window" on page 80

Related reference information:

"Event descriptions" on page 133

Editing or deleting an event script

Event scripts are only associated with forms and controls. Follow the steps below to locate and edit specific event scripts.

To edit or delete an event script::

1. Select a form or control with a script associated with it.
2. Click **Selected —> Edit script —>** *Script Name* or **Delete script —> Script —>** *Script name* from the main menu.

Related concepts:

"Scripting support" on page 136

Related tasks:

"Setting scripting preferences" on page 40
"Creating an event script" on page 77
"Renaming an event script"
"Pasting a control into the script window" on page 80
"Adding helper functions to the script window"

Renaming an event script

Due to the way scripts are handled in Mobile Application Builder it is necessary to perform a script renaming action if you want to change the name of an event script. If you do not rename an event script it is automatically named for you.

To rename an event script::

1. In the script window, click **Edit —> Rename script**.
2. Type the new script name. Do not use spaces or invalid characters.
3. Click **OK**. If an invalid name is entered, the name is not saved. Correct the name and click **OK** again.

Related concepts:

"Scripting support" on page 136

Related tasks:

"Setting scripting preferences" on page 40
"Creating an event script" on page 77
"Editing or deleting an event script" on page 78
"Pasting a control into the script window" on page 80
"Adding helper functions to the script window"

Adding helper functions to the script window

The Paste Helper window, which is available when scripting, allows you to add functions to script. Helper functions provided with Mobile Application Builder and project user-created global scripts are available for pasting. Helper functions perform all the Palm API calls needed to perform the specified task.

Support for helper functions varies by target platform.

Currently two helper functions are provided with the product:

MAB_getDateString
Gets current system date. Formatted for the database used by the Visiting Nurses sample application.

MAB_getTimeString
Gets current system time. Formatted for the database used by the Visiting Nurses sample application.

Note that if you want to use the provided helper functions, you must use the Paste Helper window to select the function call you want and have the code inserted into the editor for you. If you manually insert a call to the helper without using the window, the generation step fails. After the function call is inserted in the script window, you can modify the code.

To add helper functions to the script window::

1. In the script window, position the cursor where you would like the function to start.
2. Click **Edit —>** **Paste Helper** from the main menu or from the toolbar. If the icon and menu item are not available, this function is not supported for the current target platform.
3. Optional: Use the radio buttons provided to toggle between Mobile Application Builder helper functions and user-created global scripts.
4. Select the helper function to use. A description of the selected function is provided in the Description area.
5. Click **OK**. The selected function is placed in the script starting where the cursor was when paste helper was invoked.

Related concepts:

"Scripting support" on page 136

Related tasks:

"Creating a global script" on page 76
"Creating an event script" on page 77
"Pasting a control into the script window"

Pasting a control into the script window

You can use the Paste Part button on the script window add some helper functions to access certain properties of controls that have been dropped in a form. Currently this window supports only getting and setting text with a field control that has valid values in the Data source and Data field properties.

Support for helper functions varies by target platform.

To paste a control into the script window::

1. In the script window, position the cursor where you would like the function referring to the part to start.

2. Click **Edit —>** **Paste Part** from the main menu or from the toolbar. The Paste Part window opens. If the icon and menu item are not available, this function is not supported for the current target platform.
3. Select the part to use. Only parts from the current form will appear in the list.
4. Select the action to associate with the part. Only applicable actions for each part will appear.
5. Click **OK**. The function is added to the script window.

Related concepts:

"Scripting support" on page 136

Related tasks:

"Creating a global script" on page 76
"Creating an event script" on page 77
"Adding helper functions to the script window" on page 79

Modifying the SQL for a form event

A form may have a SQL statement associated with it. When a form has controls with defined properties for Data source and Data field, there is an associated SELECT statement that retrieves data to populate the form. If a database action (CREATE/DELETE/UPDATE record) has been associated with a form, there is a corresponding SQL statement that modifies the target database table. This feature allows you to customize SQL statements beyond the basic function available in Mobile Application Builder. For example, you may want to add relational operators or modify search conditions used in the SQL statements generated by Mobile Application Builder.

Notes about manually modifying SQL statements::

- After you edit and save a SQL statement using the **Use edited SQL** option (as described in the following steps), the SQL will *not* be regenerated by MAB with subsequent project saves.
- To flag MAB to generate the SQL statement with the next project save, select the **Use default generated SQL option** for this SQL statement.
- SQL that has been manually edited will remain in the project even if controls or forms that are using this SQL are deleted. You can manually delete modified SQL.

Follow the steps below to edit the SQL for a form.

To modify the SQL for a form event::

1. Click a form with SQL associated with it.
2. From the main menu, click **Selected -> Edit SQL**. Or, right-click on the form and then click **Edit SQL** from the pop-up menu.
3. Select the SQL you want to edit. A window opens displaying the SQL statement.

4. Click the **Use edited SQL** radio button. If you do not select this option, your changes will not be saved, and the SQL will be regenerated by MAB with the next project save.
5. Make changes as necessary. Do not change the column names or the order of the columns. Changing column information will result in an error.
6. Click **OK** to save changes and close the window.

Related tasks:

"Deleting the SQL for a form event"

Deleting the SQL for a form event

SQL that has been manually edited will remain in the project even if controls or forms that are using this SQL are deleted. Complete the following steps to delete modified SQL from your project. You cannot delete MAB-generated SQL.

To delete the SQL for a form event::

1. Click a form with SQL associated with it.
2. From the main menu, click **Selected -> Delete SQL**. Or, right-click on the form and then click **Delete SQL** from the pop-up menu. The **Delete SQL** menu item is only available if there is SQL associated with the selected form that has been manually edited.
3. Select the SQL you want to delete.
4. Click **OK** in the confirmation dialog to delete the selected SQL.

Related tasks:

"Modifying the SQL for a form event" on page 81

Using host variables

A host variable is a variable that is referred to by embedded SQL statements in an application host program. Host variables are programming variables in the application program and are the primary mechanism for transmitting data between tables in the database and application program work areas.

Use host variables to allow data extraction based on user input and linking on tables based on multi-column condition checking.

Note: If you are using defined host variables in scripts, the host variables must be cast to the required data type.

To use host variables::

1. Define the host variable at the project level.

2. Set the Update host variable property for a field, a list control, or a choice list control.
3. Set the Selection criteria property for a form, a list control, or a choice list control.

Related reference information:

"Control properties" on page 110

Defining host variables

These steps are part of the larger task of using host variables. When you complete the steps for defining host variables, return to Using host variables.

To define host variables::

1. Click **Global Definitions** in the Project pane.
2. On the Properties page of the Properties and Events pane, click ... next to the **Host Variables** field to open the Host Variables window.
3. Click **Add** to define a new host variable.
4. In the window that opens, type in the name of a host variable. All variables must be of the following SQL data types:
   - CHAR for Palm
   - STRING for Java

   Implementation will be hidden. Host variable names cannot contain any DBCS (double-byte) characters.
5. Click **OK**. The new host variable is added to the list of host variables.
6. Optional: Repeat steps 3 and 4 to add more host variables.
7. Click **OK** to close the Host Variables window.

Return to Using host variables.

Setting the Update host variable property for a field control:

These steps are part of the larger task of using host variables. When you complete the steps for Setting the **Update host variable** property for a field control, return to Using host variables.

To set the Update host variable property for a field control::

1. Select a field control.
2. On the Properties page of the Properties and Events pane, select a host variable name from the **Update Host Variables** list to define which host variable will be updated when information is entered into the field.

Return to Using host variables.

Setting the Update Host Variable property for a list control or a choice list control:

These steps are part of the larger task of using host variables. When you complete the steps for Setting the **Update Host Variable** property for a list control or a choice list control, return to Using host variables.

To set the Update Host Variable property for a list control or a choice list control::

1. Select a list control or a choice list control.
2. On the Properties page of the Properties and Events pane, click ... next to Update Host Variables to open a window where you can define which host variable will be updated with the list or choice list selection results.
3. In the **Data Source** column, select a table definition from those that have been defined as data sources for the list control or choice list control. Only the tables defined as data sources appear in the **Data Source** column. If a table is missing, make sure you have defined the data sources for the list or choice list.
4. In the **Data Field** column, select a column from the selected data source table. The columns displayed in this column come from the data source you selected in the previous step.
5. Select the host variable you want to associate with the above data source and data field. If nothing appears, make sure you have selected a data source and data field, and that you have defined a host variable.
6. Click **OK** to close the window.

Return to Using host variables.

Setting the Selection criteria for a form

These steps are part of the larger task of using host variables. When you complete the steps for Setting the **Selection criteria** for a form, return to Using host variables.

To set the Selection criteria for a form::

1. Select a form.
2. On the the Properties page of the Properties and Events pane, click ... next to the **Selection criteria** field to open a window where you can specify which database columns are associated with which host variables. This will generate the parameterized SQL query for the form. If any join SELECT statement is produced it will be added based on the column associated with the host variables.

3. In the **Data Source** column, select a table definition from those that have been imported into your project. If a table is missing, make sure you have imported it into your project.
4. In the **Data Field** column, select a column from the selected data source table. The columns displayed in this column come from the data source you selected in the previous step.
5. Select the host variable you want to associate with the above data source and data field. If nothing appears, make sure you have selected a data source and data field, and that you have defined a host variable.
6. Click **OK** to close the window.

Return to Using host variables.

Setting the Selection criteria for a list control or choice list control

These steps are part of the larger task of using host variables. When you complete the steps for Setting the **Selection criteria** for a list control or choice list control, return to Using host variables.

To set the Selection criteria for a list control or choice list control::

1. Select a list control or a choice list control.
2. On the Properties tab of the Properties and Events pane, click ... next to the **Selection criteria** field to open a window where you can specify which database columns are associated with which host variables. This will modify the SQL query for the list control or choice list control.
3. In the **Data Source** column, select a table definition from those that have been defined as data sources for the list control or choice list control. Only the tables defined as data sources appear in the **Data Source** column. If a table is missing, make sure you have defined the data sources for the list or choice list.
4. In the **Data Field** column, select a column from the selected data source table. The columns displayed in this column come from the data source you selected in the previous step.
5. Select the host variable you want to associate with the above data source and data field. If nothing appears, make sure you have selected a data source and data field, and that you have defined a host variable.
6. Click **OK** to close the window.

Return to Using host variables.

Adding print capabilities to an application

Mobile Application Builder provides limited printing support for Palm OS devices. You can access printing capabilities by setting a default print solution for each project in **Preferences**.

Related reference information:

"Printing support" on page 137
"Printing API" on page 138

Installing PalmPrint

The PalmPrint solution must be obtained from Stevens Creek Software before Mobile Application Builder print capability can be accessed using this solution.

Support for printing varies by target platform.

To install PalmPrint::

1. Obtain PalmPrint .prc files from Stevens Creek Software.
2. Install the PalmPrint .prc files on your emulator or mobile device. Follow the instructions provided by Stevens Creek Software.

For more information, see the Stevens Creek Software Web site.

Related reference information:

"Printing support" on page 137
"Printing API" on page 138

Adding a print button to a form

After choosing the printing solution you wish to implement, add printing capability to a form by adding a print button.

Support for printing varies by target platform.

To add a print button to a form::

1. Add a button to a form using the control palette.
2. Click the new button on the form pane, and click the **Properties** tab in the Properties and Events pane.
3. Type the Caption for this button as `Print` (or customize).
4. On the **Events** page, add a Print event to the button.
   a. Click … next to the **No action** label to open the Event Action and Target Selection window.
   b. Click **PrintForm** in the Action list. If the PrintForm action is not available, printing is not supported for the current target platform.
   c. Click **OK**. The Event Action and Target Selection window closes.

Related tasks:

"Installing PalmPrint" on page 86
"Adding a print menu item to a form"

Related reference information:

"Printing support" on page 137
"Printing API" on page 138
"Button properties" on page 113

Adding a print menu item to a form

After choosing the printing solution you wish to implement, add printing capability to a form by adding a print menu item.

Support for printing varies by target platform.

To add a print menu item to a form::

1. Right-click the **Menus** entry in the Project pane and click **Add new menu**. A new menu entry is created in the **Menus** section of the Project pane.
2. Click the new menu item in the Project pane, and click the **Properties** tab in the Properties and Events pane.
3. Type the Menu Item Text as `Print` (or customize), and the Accel Char as `p` (or customize).
4. On the **Events** page, add an event to the Print menu.
   a. Click … on the right to open the Event Action and Target Selection window.
   b. Click **PrintForm** in the Action list. If the PrintForm action is not available, printing is not supported for the current target platform.
   c. Click **OK**. The Event Action and Target Selection window closes.
5. Right-click the **MenuBars** entry in the Project pane and click **Add new menu bar**.
6. Click the new menu bar item in the Project pane, and click the **Properties** tab in the Properties and Events pane.
7. Type the text in the **Pull Downs** as `Main Menu` (or customize).
8. Set the **Menu Item IDs** property as the name of the menu you created in steps 1 through 4.
9. Select the form you want to add a print menu to, and click the **Properties** tab in the Properties and Events pane.
10. Set the **Menu ID** property for the form as the name of the menu bar you created in steps 5 through 8.
11. Optional: Repeat steps 8 through 10 to add the print menu item to menubars on other forms.

Related tasks:

“Installing PalmPrint” on page 86
“Adding a print button to a form” on page 86

Related reference information:

“Printing support” on page 137
“Printing API” on page 138
“Menu properties” on page 131
“Menubar properties” on page 131

Adding barcode scanning capabilities to an application

Mobile Application Builder supports hard or soft barcode scanning on devices that support this option. Hard scan devices have a scan button integrated on the device. Soft scan devices require you to create a scan button. Barcode data can be scanned into a field on a form by designating the field as scan-aware. To scan barcode information into a form for Palm applications, users will park the cursor in the desired field on a scan-aware form and perform a hard or soft scan. For WinCE applications, users need only perform the scan and do not need to park the cursor. After the scan is successfully executed, the data string will be copied to that field.

For WinCE barcode scanning applications, you must obtain the CrEme Plus JVM and select it in MAB preferences before building your application. Due to a bug in the CrEme Plus v 3.21 JVM, the initiation of a barcode ScannerClose event in a WinCE application causes the application to close. To avoid this problem until it is fixed in the JVM, do not add this event to your application.

Applications that contain barcode scanning implementation cannot be tested on emulators. You must test these applications using an actual mobile device.

To add scan capability to your application, add events and actions to the form that will receive the barcode data. The ScannerDecode event is mandatory. Other events are optional, and can be added to further customize barcode scanning behavior.

Support for barcode scanning varies by target platform.

Mobile Application Builder barcode scanning tasks::

“Adding soft-scan barcode scanning to a form” on page 89
“Adding hard-scan barcode scanning to a form” on page 90

Related reference information:

“Barcode Scanning API” on page 141
“Supported 1-dimensional barcode types for Palm applications” on page 147
“Event descriptions” on page 133
“Action descriptions” on page 135

Adding soft-scan barcode scanning to a form

Support for soft barcode scanning varies by target platform. Due to a bug in the CrEme Plus v 3.21 JVM, the initiation of a barcode ScannerClose event in a WinCE application causes the application to close. To avoid this problem, do not add this event to your application

To add soft-scan capability to a form::

1. Select the form on which you want to receive barcode scan data. The form must contain a field control.
2. On the Events page, add the first scan event to the form.
   a. Select the ScannerInit event on the left. The Event Action and Target Selection window opens.
   b. Select InitializeScanner as the action for this event.
   c. Click **OK** to close the Event Action and Target Selection window.
3. On the Events tab, add the second scan event to the form.
   a. Select the ScannerDecode event on the left. The Event Action and Target Selection window opens.
   b. Select DecodeScanTo as the action for this event. If the DecodeScanTo action is not available, make sure the form you selected contains a field control.
   c. Select the field or fields on this form that should be scan-aware as the Targets for this action.
   d. Click **OK** to close the Event Action and Target Selection window.
4. Optional, for Palm applications only: On the Events page, add a NoDecode event to the form. Select an action for this event, or specify No Action to generate a default error message.
5. On the Events tab, add the final scan event to the form.
   a. Select the ScannerClose event on the left. The Event Action and Target Selection window opens.
   b. Select DeInitScanner as the action for this event.
   c. Click **OK** to close the Event Action and Target Selection window.
6. Add a new button to the form using the control palette.
7. On the Properties page, type the Caption for this button as `Scan` (or customize).
8. On the Events tab, add a scan event to the new button.
   a. Click ... next to the No Action label to open the Event Action and Target Selection window.
   b. Select SoftScan as the action for the event.
   c. Click **OK** to close the Event Action and Target Selection window.

Related tasks:

"Adding hard-scan barcode scanning to a form" on page 90

Related reference information:

"Barcode Scanning API" on page 141
"Supported 1-dimensional barcode types for Palm applications" on page 147
"Event descriptions" on page 133
"Action descriptions" on page 135

Adding hard-scan barcode scanning to a form

Support for barcode scanning varies by target platform. Due to a bug in the CrEme Plus v 3.21 JVM, the initiation of a barcode ScannerClose event in a WinCE application causes the application to close. To avoid this problem, do not add this event to your application

To add hard-scan capability to a form::

1. Select the form on which you want to receive barcode scan data.
2. On the Events page, add the first scan event to the form.
   a. Select the ScannerInit event on the left. The Event Action and Target Selection window opens.
   b. Select InitializeScanner as the action for this event.
   c. Click **OK** to close the Event Action and Target Selection window.
3. On the Events page, add the second scan event to the form.
   a. Select the ScannerDecode event on the left. The Event Action and Target Selection window opens.
   b. Select DecodeScanTo as the action for this event.
   c. Select the field or fields on this form that should be scan-aware as the Targets for this action.
   d. Click **OK** to close the Event Action and Target Selection window.
4. Optional, for Palm applications only: On the Events page, add a NoDecode event to the form. Select an action for this event, or specify No Action to generate a default error message.
5. On the Events tab, add the final scan event to the form.
   a. Select the ScannerClose event on the left. The Event Action and Target Selection window opens.
   b. Select DeInitScanner as the action for this event.
   c. Click **OK** to close the Event Action and Target Selection window.

Related tasks:

"Adding soft-scan barcode scanning to a form" on page 89

Related reference information:

"Barcode Scanning API" on page 141
"Supported 1-dimensional barcode types for Palm applications" on page 147
"Event descriptions" on page 133
"Action descriptions" on page 135

Adding synchronization capabilities to an application

You can add synchronization capabilities to an application using DB2 Everyplace Sync Server functions. You must install and configure the Sync Server before you can add synchronization capabilities to an application.

When a synchronization is initiated by the user, all the enabled subscription sets in the configuration store will be synchronized. The location of the DB2 Everyplace database on the device is used as the location of the configuration store.

See the *Sync Server Administration Guide* for more information on Sync Server subscription sets and the configuration store.

Some input parameters must be obtained from the user at runtime before synchronization can occur. It is strongly recommended that you explicitly add a synchronization preferences dialog to your application in order to obtain these parameters from the user before synchronization is attempted. If a synchronization action is attempted and these preferences have not been previously set, Mobile Application Builder will automatically invoke this dialog once. However, if the dialog is not also explicitly added to the application, the user will not have the opportunity to change or correct these preferences.

To add synchronization capabilities to a control::

1. Select the control that you want to add synchronization function to.
2. Add a synchronization event to the control.
   a. On the Events page, select an event on the left. The Event Action and Target Selection window opens.
   b. Select Sync Application as the action for this event.

To add an event to a control to display a synchronization preferences dialog::

1. Select a control.
2. On the Events page, select an event on the left. The Event Action and Target Selection window opens.
3. Select Show as the action for this event, and select Sync Preferences dialog as the target for the action.

Related tasks:

"Adding a control to a form" on page 51

Related reference information:

"Event descriptions" on page 133
"Action descriptions" on page 135

Building a Mobile Application Builder application

Build your application to compile all the controls and resources into one file that can be read by the target mobile device.

Related tasks:

"Testing a Mobile Application Builder application" on page 96

Related reference information:

"Code generation/build files descriptions" on page 94

Defining preprocessor statements

Use preprocessor statements to define statements to be used during code generation. The statements entered into the Preprocessor Statements window will be inserted in at the top of the code.

Support for preprocessor statements varies by target platform.

To define preprocessor statements::

1. Click Global Definitions in the Project pane.
2. Click ... in the Properties page in the Properties and Events pane next to Preprocessor statements to open the Preprocessor Statements window.
3. Define the preprocessor statements to be used during code generation.
4. Click **OK** to close the window.

Related reference information:

"Code generation/build files descriptions" on page 94

Building applications

Build your application to compile all the controls and resources into one file that can be read by the target mobile device.

Before building an application, ensure that the Mobile Application Builder environment is set up properly. See Setup and Configuration for more information.

For target platforms that require it, make sure the application has a unique application ID, otherwise the application may not behave properly.

To build an application::

1. Click **Build ->** **Build** from the menu, or from the toolbar. A Building Application status dialog displays the build in progress. This step generates your application. A build completion dialog displays the status of the completed build. When complete, a PRC file is created in your project subdirectory, along with several intermediate files.

2. If you have any errors, refer to Troubleshooting to find a possible solution.

Related tasks:

"Testing a Mobile Application Builder application" on page 96
"Viewing and clearing the message log"
"Saving the message log"

Related reference information:

"Code generation/build files descriptions" on page 94

Viewing and clearing the message log

The message log contains information about each build that was done during the current session. You can also clear the message log manually. The most current information is located at the bottom of the log.

To view the message log::

1. Click **Help -> View log file** from the main menu. The Message Log window opens.

To clear the message log::

1. In the Message Log window, click **Clear**.

Related tasks:

"Building applications" on page 92
"Saving the message log"

Related reference information:

"Code generation/build files descriptions" on page 94
"Troubleshooting" on page 148

Saving the message log

If you encounter an error that cannot be resolved, save the message log file. It is important that the log is saved before ending the session where the error occurred because the log is cleared each time Mobile Application Builder is closed.

To save the message log::

1. Locate the `db2emab.log` log file in the `../ApplicationBuilder/bin/` directory.
2. Create a copy of the log file in the same directory.

Related tasks:

"Building applications" on page 92
"Viewing and clearing the message log" on page 93

Related reference information:

"Code generation/build files descriptions"
"Troubleshooting" on page 148

Code generation/build files descriptions

Mobile Application Builder generates intermediate C code or Java code, depending on the target platform you have chosen, to create the final application executable file. The table below provides a general description of which code is generated for each supported target platform.

Table 15. Mobile Application Builder Code Generation

Target platform	Generated code
Palm	C
Symbian	Java
WinCE	Java
Sharp Zaurus	Java
Generic Java	Java

When you build your application many other intermediate files are created in order to create the final executable file. Any manual changes to these intermediate files will be lost if you build your application again using Mobile Application Builder.

Some intermediate files are not saved after a build if you select the **Release** option in Build preferences. The tables below provide descriptions of generated files for each supported target platform.

The following files are generated for Palm applications.

Table 16. Palm C Code Application Build Files

File(s)	Description	Deleted in Release mode?
*.prc	Palm application executable. This is the final output created when a project is built successfully.	No
*.c	Source code for the application.	Yes
sql*.h	Header files your application needs to access the DB2 Everyplace engine functions.	Yes
*.h	Header file used to generate application.	Yes
*.makefile	Used to build executable application for Palm.	Yes

Table 16. Palm C Code Application Build Files (continued)

File(s)	Description	Deleted in Release mode?
*.rcp	Definition of Palm resources in application.	Yes
MAB-sections.*	Related to multiple code segment support.	No
*.bin	Executable Palm resources in application.	Yes (always deleted after a successful build)
*.s, *.ld, *.stamp, MAB*.o	These intermediate files are *always* deleted after a successful build. Although they may be saved if a build is unsuccessful, they are not useful for debugging.	Yes (always deleted after a successful build)
MAB_Icon.bmp, *.def	These files are deleted after a successful build *only* if you select the **Release** option in Build preferences.. Saving them may be useful for debugging.	Yes

The following files are generated for *all* target platforms that require Java code generation.

Table 17. Java Application Build Files (All Platforms)

File(s)	Description	Deleted in Release mode?
*.java	The generated Java source files.	Yes
*.class	The class files generated by the java compiler.	Yes
*.jar	The application jar generated from the class files.	No
*.bat	These files are generated and run to compile the source and package the application.	No

In addition to the files in Table 5 above, the following files are generated for Symbian applications.

Table 18. Symbian Application Build Files

File(s)	Description	Deleted in Release mode?
*.sis	Symbian application executable.	No
.bmp,.mbm	These files are related to icons for the application. The *.bmp file is supplied by the user and the *.mbm file is generated by the Symbian tool chain.	No
.app,.aifb,*.aif, *.txt,*.pkg	Intermediate files related to code generation. Not useful for debugging.	No

In addition to the files in Table 5 above, the following files are generated for WinCE applications.

Table 19. WinCE Application Build Files

*.cab	WinCE application executable.	No
.inf,.DAT,*.exe	Intermediate files related to code generation. Not useful for debugging.	No
File(s)	**Description**	**Deleted in Release mode?**

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Setting build preferences" on page 38

Testing a Mobile Application Builder application

Use an emulator or the actual mobile device to test applications.

You must build an application before you can test it on an emulator or mobile device.

To use applications generated in Java, the device or emulator must support a Java Virtual Machine (JVM). Although other device platforms capable of supporting JVMs are supported, a device with a minimum of 16MB of RAM is recommended. Some newer devices will have JVMs built into the device. Other devices have a JVM packaged on a CD that comes with the device. Or, you can purchase and download other JVMs on the web.

Related tasks:

"Building a Mobile Application Builder application" on page 91
"MAB version 8.1.4 setup and configuration" on page 12

Related reference information:

"Code generation/build files descriptions" on page 94

Populating tables using a live connection

You can use a live connection to obtain application test data for device-side tables and columns. Using this method, MAB creates an initialization application, similar to the NurseInit sample application, that you can run on the emulator or device to populate the tables you need for your application.

These steps are part of the larger task of testing an application on an emulator or mobile device. When you have finished the steps for creating and populating tables, return to the appropriate section in the testing document:

To populate tables using a live connection::

1. Click Build — Populate from the main menu, or click the Populate icon on the toolbar. A window opens asking if you want to create a new connection.
   - If you have an existing connection to the database you want to use to populate the device or emulator, click No. The Population dialog opens, with the tables you imported into the project listed in the Available tables list.
   - If you need to create a new connection, click Yes, and the Connection window opens. In the Connection window, select the type of database you want to support, and fill in all other required fields. You must provide the path to the JDBC driver on your workstation. Click OK. The Population dialog opens, with the tables you imported into the project listed in the Available tables list.
2. In the Available tables list, select one or more tables. You can view the data from the selected tables in the Population list.
3. Click OK. An application named *projectname*Init is created in a new `\data` folder under your project folder.
4. Install and run this application to initialize the tables used in your main application.

Return to "Testing a Palm application on an emulator" on page 99
Return to "Testing a Palm application on a mobile device" on page 100
Return to "Testing a Symbian Crystal application on an emulator" on page 101
Return to "Testing a Symbian Crystal application on a mobile device" on page 102
Return to "Simulating a WinCE application on the desktop" on page 105
Return to "Testing a WinCE application on a mobile device" on page 105

Creating and populating tables manually

Although you can import the definition of tables into the Mobile Application Builder program for use in building applications, the tables used by your application must be created and populated on either the emulator or a mobile device before you run the application the first time. If your application populates its tables when it runs (i.e. starts with empty tables), you still need to create the tables one time on the database.

The preferred method for populating tables is through the Populate function of MAB. However, there are several alternative ways to do so, selection of which depends on the use of your application and the amount of data that you want to load into the tables.

These steps are part of the larger task of testing an application on an emulator or mobile device. When you have finished the steps for creating and populating tables, return to the appropriate section in the testing document:

Creating tables for WinCE applications by using SchemaInit::

NurseInit.bat, in the , is sample script that uses the supplied SchemaInit.java program to create and populate database tables based on row content in an ASCII file. Modify NurseInit.bat to adjust path information to where you installed DB2 Everyplace, then run it to create the schema used by the PersonList and VNApp samples, on the desktop, at a directory of your choice. SchemaInit will drop the tables from the database if they already exist, and will also create the target directory (Windows) if it does not exist.

To use this script to create and populate tables for your application, modify tables.txt and data.txt, in the ..\Projects\Samples\Win32 directory, to reflect the tables and data that will be accessed for your application.

Creating tables by using CLP::

For application prototyping or testing, when only a small amount of table data is needed, you can use the function within the supplied CLP application running on a device or the emulator. Enter CREATE TABLE statements to create tables, and use INSERT INTO to add rows to the tables. See the *IBM DB2 Everyplace Application Development Guide* for details about SQL statements. See *IBM DB2 Everyplace Installation and User's Guide* for information about using the CLP application.

Creating tables by importing from Comma Separated Value (CSV) files::

For applications involving an amount of data for which it would be inappropriate to enter by hand, you can use the DB2 Everyplace Import tool to create tables and load the data from CSV files. Those files can be built using various desktop tools. You can cause the data definition language (DDL) statement for a table to be placed in the file system on your workstation by using the Generate DDL... menu function against a table in the DB2 Control Center. You can also produce CSV files (use the "Delimited ASCII format" from the Export menu against a table) to produce a file containing row data, which can then be used with the PalmImport.bat program on Windows and the DB2eImport application on the device or emulator to load the data.

Creating tables by using the IMPORT command from DB2e CLP::

DB2 Everyplace includes the DB2eCLP application. From DB2e CLP, you can issue the IMPORT command, which is available for 7.2 and above for the Palm platform. This was previously available for the other non-Palm platforms. After entering a CREATE TABLE command, you can then issue the IMPORT command to cause the table rows to be populated from data you have placed into a Palm Memo application. The first line of the memo is its Palm name, and the same name is specified to the IMPORT command. The rest of the memo should contain the row values, separated by commas (CSV format). The contents can be created on the device, emulator, or Palm Desktop software (then HotSync). See *DB2 Everyplace Installation and User's Guide*, in the chapter entitled: "The CLP application."

Creating tables by writing an application::

You can write an application whose function is to create and populate any tables your application needs. You would then run this initializing application once before using your main application. The supplied sample program NurseInit functions this way. NurseInit uses DB2 CLI functions, and was constructed using the CodeWarrior code development tool. It would be possible to write CLI code (script) to use within a MAB built application, but MAB is not currently designed to easily do so. See *IBM DB2 Everyplace Application Development Guide* for details about CLI programming. The C source code and related files are installed in the Samples directory where you installed the DB2 Everyplace Database component.

Creating tables by synchronizing with a backend database using the Sync Server::

If you have installed the IBM DB2 Everyplace Sync Server component, you can, after defining synchronization subscriptions, use the IBM Sync application ″refresh″ setting on the device or emulator to create and initialize (populate) the tables you need on your device or emulator from tables that already exist on backend databases. See *IBM DB2 Everyplace Sync Server Administration Guide* for more information.

Return to “Testing a Palm application on an emulator”
Return to “Testing a Palm application on a mobile device” on page 100
Return to “Testing a Symbian Crystal application on an emulator” on page 101
Return to “Testing a Symbian Crystal application on a mobile device” on page 102
Return to “Simulating a WinCE application on the desktop” on page 105
Return to “Testing a WinCE application on a mobile device” on page 105

Testing Palm applications

You can test Palm applications on an emulator or a device.

Mobile Application Builder Palm application testing tasks:

- “Testing a Palm application on an emulator”
- “Testing a Palm application on a mobile device” on page 100

Testing a Palm application on an emulator

You can use an emulator to test applications. Most mobile devices have emulators or simulators that allow you test the application in the mobile environment on your development machine. Emulators allow you to upload applications without having to have the actual mobile device.

Prerequisites to Palm emulator testing:

- Proper installation and configuration of a Palm emulator on your workstation. See Palm setup and configuration for details.
- Successful build of your application in Mobile Application Builder.

To test a Palm application on an emulator::

1. Click **Build->** Test from the Mobile Application Builder menu bar, or from the toolbar. An emulator session launches with your application installed.
2. Create and populate the tables needed by your application.
3. Tap the application icon to launch it.

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Testing a Palm application on a mobile device"

Related reference information:

"Code generation/build files descriptions" on page 94

Testing a Palm application on a Palm OS 5 Simulator

You can use a Palm OS 5 Simulator to test Palm OS 5.x applications. The simulator allows you to upload applications without having to have the actual mobile device.

Prerequisites to Palm OS 5 Simulator testing:

- Proper installation and configuration of a Palm OS 5 Simulator on your workstation. See Palm setup and configuration for details.
- Successful build of your application in Mobile Application Builder.

To test a Palm OS 5.x application on a Palm OS 5 Simulator::

1. Start the simulator by double-clicking on `PalmSim.exe`.
2. Open the simulator session that you saved after you completed the simulator setup steps:
   a. Right-click anywhere on the simulator screen, and click **Storage —> Load**.
   b. Browse to the appropriate simulator session and click **Open**.
3. Right-click anywhere on the simulator screen and click **Install —> Database** from the pop-up menu.
4. Browse to your application and click **Open**. The application is installed on the simulator.
5. Create and populate the tables needed by your application.
6. Tap the application icon to launch it.

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Testing a Palm application on a mobile device"

Related reference information:

"Code generation/build files descriptions" on page 94

Testing a Palm application on a mobile device

Use an emulator or the actual mobile device to test applications. Although testing on an emulator works well, you should confirm your application functions on the actual mobile device.

Prerequisites to Palm mobile device testing:

- Proper configuration of a Palm mobile device for testing. See Mobile Application Builder Setup and Configuration for details.
- Successful build of your application in Mobile Application Builder.

To test a Palm application on a mobile device::

1. Install the application on the device using the Install function of the Palm Desktop software.
2. Create and populate the tables needed by your application.
3. The next Palm HotSync adds the application.
4. Tap the application icon to launch it.

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Testing a Palm application on an emulator" on page 99

Related reference information:

"Code generation/build files descriptions" on page 94

Testing Symbian Crystal applications

You can test Symbian Crystal applications on an emulator or a device.

Mobile Application Builder Symbian Crystal application testing tasks:

- "Testing a Symbian Crystal application on an emulator"
- "Testing a Symbian Crystal application on a mobile device" on page 102

Testing a Symbian Crystal application on an emulator

You can use an emulator to test applications. Most mobile devices have emulators or simulators that allow you test the application in the mobile environment on your development machine. Emulators allow you to upload applications without having to have the actual mobile device.

Applications developed using Mobile Application Builder look for the database tables in the default location on the emulator. For Symbian Crystal, this is `\system\data\isync\`. Before the applications are used, make sure that the required database tables exist in this location.

When you test a Symbian Crystal application from Mobile Application Builder, Mobile Application Builder puts the built application files onto your desktop in the

```
x:\Symbian\6.0\NokiaJava\Epoc32\Wins\c\System\Apps\<application name>
```

directory. To set up other .SIS files you wish to install, copy them to the

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

directory.

Prerequisites to Symbian Crystal emulator testing:

- Proper installation and configuration of a Symbian Crystal emulator on your workstation. See Mobile Application Builder Setup and Configuration for details.
- Successful build of your application in Mobile Application Builder.

To test a Symbian Crystal application on an emulator::

1. Click **Build->** Test from the Mobile Application Builder menu bar, or from the toolbar. An emulator session launches, with your application installed.
2. Create and populate the tables needed by your application.
3. Click the application icon on the emulator screen to launch your application.

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Testing a Symbian Crystal application on a mobile device"

Related reference information:

"Code generation/build files descriptions" on page 94

Testing a Symbian Crystal application on a mobile device

Mobile Application Builder generates the device installable .SIS file (*`ApplicationIconName`*`.SIS`) in the project directory.

Applications developed using Mobile Application Builder look for the database tables in the default locations on the device. For Symbian Crystal, this is`\system\data\isync\`. Before the applications are used, make sure that the required database tables exist in this location.

Prerequisites to Symbian Crystal mobile device testing:

- Proper installation and configuration of a Symbian Crystal device. See Mobile Application Builder Setup and Configuration for details.
- Successful build of your application in Mobile Application Builder.

To test a Symbian Crystal application on a mobile device::

1. Install the *`ApplicationIconName`*`.SIS` file on the mobile device. The application appears in the **Extras** bar on the mobile device.
2. Create and populate the tables needed by your application.
3. Select the application icon and start it to launch the application.

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Testing a Symbian Crystal application on an emulator" on page 101

Related reference information:

"Code generation/build files descriptions" on page 94

Testing Symbian UIQ applications

You can test Symbian UIQ applications on an emulator or a device.

Mobile Application Builder Symbian UIQ application testing tasks:

- "Testing a Symbian UIQ application on an emulator"
- "Testing a Symbian UIQ application on a mobile device" on page 104

Testing a Symbian UIQ application on an emulator

You can use an emulator to test applications. Most mobile devices have emulators or simulators that allow you test the application in the mobile environment on your development machine. Emulators allow you to upload applications without having to have the actual mobile device.

Applications developed using Mobile Application Builder look for the database tables in the default location on the emulator. For Symbian UIQ, this is `\system\data\isync\`. Before the applications are used, make sure that the required database tables exist in this location.

When you test a Symbian UIQ application from Mobile Application Builder, Mobile Application Builder puts the built application files onto your desktop in the

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

directory. To set up other .SIS files you wish to install, copy them to the

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

directory.

Prerequisites to Symbian UIQ emulator testing:

- Proper installation and configuration of a Symbian UIQ emulator on your workstation. See Mobile Application Builder Setup and Configuration for details.
- Successful build of your application in Mobile Application Builder.

To test a Symbian UIQ application on an emulator::

1. Click **Build->** Test from the Mobile Application Builder menu bar, or from the toolbar. An emulator session launches, with your application installed.
2. Create and populate the tables needed by your application.
3. Click the application icon on the emulator screen to launch your application.

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Testing a Symbian UIQ application on a mobile device"

Related reference information:

"Code generation/build files descriptions" on page 94

Testing a Symbian UIQ application on a mobile device

Mobile Application Builder generates the device installable .SIS file (`*ApplicationIconName*.SIS`) in the project directory.

Applications developed using Mobile Application Builder look for the database tables in the default locations on the device. For Symbian UIQ , this is`\system\data\isync\`. Before the applications are used, make sure that the required database tables exist in this location.

Prerequisites to Symbian UIQ mobile device testing:

- Proper installation and configuration of a Symbian UIQ device. See Mobile Application Builder Setup and Configuration for details.
- Successful build of your application in Mobile Application Builder.

To test a Symbian UIQ application on a mobile device::

1. Install the `*ApplicationIconName*.SIS` file on the mobile device.
2. Create and populate the tables needed by your application.
3. Select the application icon and start it to launch the application.

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Testing a Symbian UIQ application on an emulator" on page 103

Related reference information:

"Code generation/build files descriptions" on page 94

Testing WinCE applications

You can test WinCE applications on a device or simulate applications on the desktop.

Mobile Application Builder WinCE application testing tasks:

- "Simulating a WinCE application on the desktop" on page 105
- "Testing a WinCE application on a mobile device" on page 105

Simulating a WinCE application on the desktop

You can run WinCE applications on the desktop using Win32 emulation capability. Although this is not true emulation, it allows you to see what an application will look like on a device.

Applications developed using Mobile Application Builder look for the database tables on the default locations on the device or emulator. For WinCE/PocketPC , this is \. Before the applications are used, make sure that the required database tables exist in this location.

Prerequisites to WinCE desktop simulation:

- Proper installation and configuration of the Win32 emulator on your workstation. See Mobile Application Builder Setup and Configuration for details.
- Successful build of your application in Mobile Application Builder.

To simulate a WinCE application on the desktop::

1. Click **Build->** Test from the Mobile Application Builder menu bar, or from the toolbar. A Win 32 emulator session launches, with your application installed.
2. Create and populate the tables needed by your application.

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Testing a WinCE application on a mobile device"

Related reference information:

"Code generation/build files descriptions" on page 94

Testing a WinCE application on a mobile device

Applications developed using Mobile Application Builder look for the database tables on the default locations on the device. For WinCE\PocketPC this is \. Before the applications are used, make sure that the required database tables exist in this location.

Prerequisites to testing an application on a mobile WinCE mobile device:

- Proper configuration of the Win32 mobile device. See Mobile Application Builder Setup and Configuration for details.
- Successful build of your application in Mobile Application Builder.

To test a WinCE application on a mobile device::

Mobile Application Builder generates application installables for any device with a JVM. The WinCE Build Preferences should be set appropriately for the JVM before the application is built.

1. Copy *projectname*`.cab` to any directory on the device. Tap the file on the device to install the application.
2. Create and populate the tables needed by your application.
3. Tap the application icon in the **Start** menu to launch the application.

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Simulating a WinCE application on the desktop" on page 105

Related reference information:

"Code generation/build files descriptions" on page 94

Testing Sharp Zaurus applications

You can test Sharp Zaurus applications on a device or simulate applications on the desktop.

Mobile Application Builder Sharp Zaurus application testing tasks:

- "Simulating a Sharp Zaurus application on the desktop"
- "Testing a Zaurus application on a mobile device" on page 107

Simulating a Sharp Zaurus application on the desktop

You can run Sharp Zaurus applications on the desktop using Win32 emulation capability. Although this is not true emulation, it allows you to see what an application will look like on a device.

Prerequisites to Zaurus desktop simulation:

- Proper installation and configuration of the Win32 emulator on your workstation. See Mobile Application Builder Setup and Configuration for details.
- Successful build of your application in Mobile Application Builder.

To simulate a Zaurus application on the desktop::

1. Click **Build->** Test from the Mobile Application Builder menu bar, or from the toolbar. A Win 32 emulator session launches, with your application installed.
2. Create and populate the tables needed by your application.

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Testing a Zaurus application on a mobile device"

Related reference information:

"Code generation/build files descriptions" on page 94

Testing a Zaurus application on a mobile device

Mobile Application Builder generates the device installable .IPK file in the project directory.

Prerequisites to Zaurus mobile device testing:

- Proper configuration of the Sharp Zaurus mobile device. See Mobile Application Builder Setup and Configuration for details.
- Successful build of your application in Mobile Application Builder.

To test a Zaurus application on a mobile device::

1. Install the `.IPK` file on the mobile device. The application appears in the **Documents** location on the mobile device, with a shortcut under Jeode.
2. Create and populate the tables needed by your application.
3. Tap the application icon to launch the application.

Related tasks:

"Building a Mobile Application Builder application" on page 91
"Simulating a Sharp Zaurus application on the desktop" on page 106

Related reference information:

"Code generation/build files descriptions" on page 94

Chapter 2. Concepts/Reference

This section provides concept and reference information for Mobile Application Builder.

Palette control icons

The Control Palette contains controls to be used for constructing your application. Mobile Application Builder comes with a standard palette of controls. The controls that are available for your application vary by target platform.

Table 20. Palette control icons. Description

Icon	Description
	Form Bitmap
	Button
	Check Box
	Field
	Graffiti Shift
	Ink
	Blob
	Label
	List
	Pop-up Trigger (Palm platforms), Choice List (Java platforms)
	Push Button
	Repeat Button
	Scrollbar
	Selector Trigger

Related tasks:

“Controls” on page 50

Related reference information:

"Control properties"

Project properties

Set properties for your MAB project. Project properties vary depending on your target platform.

Table 21. Project properties

Property	Description
Name	The name used to identify the target platform. You cannot modify this property.
Startup Form	The form that will display first when the application is run.
Application ID	The unique identifier for this application. See "Application IDs" on page 42 for more information about application IDs.
Category	The application category for this application on a Palm OS emulator or device. This category will be created, if it does not already exist, in the Palm application launcher.
Database Location	The location of the database tables for this application.

Related tasks:

"MAB projects" on page 31
"Application IDs" on page 42

Control properties

The Control Palette contains controls to be used for constructing your application. For each control there are many attributes that can be customized to fit your applications needs.

The controls that are available for use vary by target platform.

Mobile Application Builder control properties::

"Form bitmap properties" on page 111
"Button properties" on page 113
"Check box properties" on page 114
"Choice list properties" on page 115
"Command button array (CBA) properties" on page 117
"Field properties" on page 117
"Graffiti shift properties" on page 119
"Ink properties" on page 120
"Label properties" on page 121
"List properties" on page 121
"Pop-up trigger properties" on page 123

"Push button properties" on page 124
"Repeat button properties" on page 125
"Scrollbar properties" on page 126
"Selector trigger properties" on page 127

Related tasks:

"Controls" on page 50

Related reference information:

"Palette control icons" on page 109

Form bitmap properties

Displays a bitmap image on a form. A bitmap image can be created using any external drawing program that allows you to save images in black and white or direct color (supported in Palm OS 4.0 or higher) bitmap format. Store bitmaps in your application's project directory.

Support for this control varies by target platform.

Table 22. Form bitmap properties

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Caption	The text that will appear on the bitmap in the Mobile Application Builder Form pane. This caption text will not be visible to users.
Left	The distance from the left edge of the control to the screen edge. Use this property to manually set the position of the control.
Top	The distance from the top edge of the control to the screen edge. Use this property to manually set the position of the control.
Bitmap resource name	The name of the Bitmap file. You must define a bitmap resource in the project tree view before you can select it here.
Width	The width of the control in pixels. Use this property to manually set the size of the control.
Height	The height of the control in pixels. Use this property to manually set the size of the control.

Related tasks:

"Controls" on page 50
"Creating and displaying bitmaps" on page 64

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

Blob properties

Use this control to save an image as a BLOB data type. Use of this control requires blob support on the device Java Virtual Machine, which is a Java 1.2 feature. If you are using a custom Java Virtual Machine, you must ensure that the SQL package has blob support.

Support for this control varies by target platform.

Table 23. Blob properties

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Data sources	The name of the table that contains the Data field or column you want the control to read and write to. The list of available Data sources is based on the tables that have been imported into the project. See "Creating and importing a table definition file" on page 44 for more information.
Data field	The name of the column you want the control to read and write to. The list of available Data fields is based on the table selected in the Data source property.
Usable	A USABLE control is visible to the user. USABLE is the default setting for this property. A NONUSABLE control does not draw on the form, but can be programmed to draw during execution.
Left	The distance from the left edge of the control to the screen edge. Use this property to manually set the position of the control.
Top	The distance from the top edge of the control to the screen edge. Use this property to manually set the position of the control.
Width	The width of the control in pixels. Use this property to manually set the size of the control.
Height	The height of the control in pixels. Use this property to manually set the size of the control.

Related tasks:

"Controls" on page 50

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

Button properties

Executes an event. The event performs an action. The action may have a target. For example, the button Tap event performs the RunScript action that executes the target script.

Support for this control varies by target platform.

Table 24. Button properties

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Caption	The text that will appear on the button.
Font	The font used in the caption for the control.
Bitmap	The bitmap image that will appear on the control in an unselected state. You must add a bitmap image to your project before you can specify it here. In order for the bitmap to appear on the control, you must specify an image for both the Bitmap and Selected Bitmap properties. These images can be the same. Support for bitmaps varies by target platform.
Selected Bitmap	The bitmap image that will appear on the control in a selected state. You must add a bitmap image to your project before you can specify it here. In order for the bitmap to appear on the control, you must specify an image for both the Bitmap and Selected Bitmap properties. These images can be the same. Support for bitmaps varies by target platform.
Usable	A USABLE control is visible to the user. USABLE is the default setting for this property. A NONUSABLE control does not draw on the form, but can be programmed to draw during execution.
Background	The color of the background for this control. Support for this property varies by target platform.
Foreground	The color of the foreground for this control. Support for this property varies by target platform.
AnchorLeft	If LEFTANCHOR is selected, the left bound of the control is fixed. LEFTANCHOR is the default setting for this property. If RIGHTANCHOR is selected, the right bound of the control is fixed. Support for this property varies by target platform.
Frame	If FRAME is selected, a rectangular frame with rounded corners is drawn around the control. If NOFRAME is selected, no border is drawn around the control. If BOLDFRAME is selected, a bold rectangular frame with rounded corners is drawn around the control. Support for this property varies by target platform.
Left	The distance from the left edge of the control to the screen edge. Use this property to manually set the position of the control.
Top	The distance from the top edge of the control to the screen edge. Use this property to manually set the position of the control.

Table 24. Button properties (continued)

Property	Description
Width	The width of the control in pixels. Use this property to manually set the size of the control.
Height	The height of the control in pixels. Use this property to manually set the size of the control.

Related tasks:

"Controls" on page 50

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

Check box properties

Boolean control with 2 states, checked and cleared. Can execute a Tap event to run a script to verify state. Initial state may be set by the Data Source and Data Field properties. Checking or clearing can execute an Update event to Data Source and Data Field.

Table 25. Check box properties

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Caption	The text displayed to the right of the control.
Data sources	The name of the table that contains the Data field or column you want the control to read and write to. The list of available Data sources is based on the tables that have been imported into the project. See "Creating and importing a table definition file" on page 44 for more information.
Data field	The name of the column you want the control to read and write to. The list of available Data fields is based on the table selected in the Data source property.
Font	The font used in the caption for the control.
Usable	A USABLE control is visible to the user. USABLE is the default setting for this property. A NONUSABLE control does not draw on the form, but can be programmed to draw during execution.
AnchorLeft	If LEFTANCHOR is selected, the left bound of the control is fixed. LEFTANCHOR is the default setting for this property. If RIGHTANCHOR is selected, the right bound of the control is fixed. Support for this property varies by target platform.

Table 25. Check box properties (continued)

Property	Description
GroupID	A unique number for each check box that maps to the same column in a table. Mobile Application Builder automatically assigns the unique value for each check box that links to the same column. A non-visible control is not considered part of the application and does not draw.
Checked	If True, the initial selection state of the check box is checked. If False, the initial selection state of the check box is not checked.
Background	The color of the background for this control. Support for this property varies by target platform.
Foreground	The color of the foreground for this control. Support for this property varies by target platform.
Left	The distance from the left edge of the control to the screen edge. Use this property to manually set the position of the control.
Top	The distance from the top edge of the control to the screen edge. Use this property to manually set the position of the control.
Width	The width of the control in pixels. Use this property to manually set the size of the control.
Height	The height of the control in pixels. Use this property to manually set the size of the control.

Related tasks:

"Controls" on page 50

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

Choice list properties

Allows the user to make a selection from multiple choices that drop down in a list when the control is tapped. The order of actions starts when the choice list control is clicked. Next the list is displayed and the user makes a selection. Before the choice list label changes to display the selection all events associated with the control are executed.

This control is only available for Java platforms.

Table 26. Choice list properties

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.

Table 26. Choice list properties (continued)

Property	Description
Usable	A USABLE control is visible to the user. USABLE is the default setting for this property. A NONUSABLE control does not draw on the form, but can be programmed to draw during execution.
Data source	The name of the table that contains the data field or column you want the control to read and write to. The list of available data sources is based on the tables that have been imported into the project. See Table definition files for more information.
Data field	The name of the column you want the control to read and write to. The list of available data fields is based on the table selected in the Data source property.
Selection criteria	Click on the ellipse to the right of the **Selection criteria** field to open a window where you can associate a column from a database table with a host variable that you have already defined. See Using host variables for more information. Support for this property varies by target platform.
Update host variable	Click on the ellipse to the right of Update host variable to open a window where you can define which previously-defined host variables will be updated with the choice list selection results. Support for this property varies by target platform.
AnchorLeft	If LEFTANCHOR is selected, the left bound of the control is fixed. LEFTANCHOR is the default setting for this property. If RIGHTANCHOR is selected, the right bound of the control is fixed. Support for this property varies by target platform.
Background	The color of the background for this control. Support for this property varies by target platform.
Foreground	The color of the foreground for this control. Support for this property varies by target platform.
Left	The distance from the left edge of the control to the screen edge. Use this property to manually set the position of the control.
Top	The distance from the top edge of the control to the screen edge. Use this property to manually set the position of the control.
Width	The width of the control in pixels. Use this property to manually set the size of the control.
Height	The height of the control in pixels. Use this property to manually set the size of the control.

Related tasks:

"Controls" on page 50

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

Command button array (CBA) properties

The CBA is a set of four hardware buttons on the Nokia communicator. These buttons are usually assigned to the most frequent user interaction options for the current view. The CBA is used in all the applications on the communicator and is an essential feature for this device.

Mobile Application Builder adds four CBA controls to any new forms that are created for the Symbian Crystal target platform. These controls correspond to the four hardware buttons on the device. You can configure properties and events for the CBA controls, but you cannot move them to a different location on the form, remove them from the form, or resize them.

Table 27. Command button array (CBA) properties

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Caption	The text displayed for the control.
Usable	A USABLE control is visible to the user. USABLE is the default setting for this property. A NONUSABLE control does not draw on the form, but can be programmed to draw during execution.
Left	The distance from the left edge of the control to the screen edge. Cannot be edited.
Top	The distance from the top edge of the control to the screen edge. Cannot be edited.
Width	The width of the control in pixels. Cannot be edited.
Height	The height of the control in pixels. Cannot be edited.

Related tasks:

"Controls" on page 50

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

Field properties

Displays or allows entry of data. Initial state may be set using the Data Source and Data Field properties. Modifying the field can execute an Update event to Data Source and Data Field. May also be updated using strings.

Table 28. Field properties

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Data source	The name of the table that contains the data field or column you want the control to read and write to. The list of available Data sources is based on the tables that have been imported into the project. See Table definition files for more information.
Data field	The name of the column you want the control to read and write to. The list of available data fields is based on the table selected in the Data source property.
Data field type	Displays the data type for the column you selected for the Data field property. Cannot be edited.
Use Unique Value	Click ... to the right of the Use Unique Value field to open a window where you can specify whether to assign a unique value to the field. You can also specify parameters to include in the unique value. Only available if the Data field type is VARCHAR or CHAR. Only available if the Max Char property for this control is 26 or greater.
Background	The color of the background for this control. Support for this property varies by target platform.
Foreground	The color of the foreground for this control. Support for this property varies by target platform.
Link source	The name of the table to link to.
Link field	The name of the column that contains the key to link to.
Update host variable	Select which previously-defined host variables will be updated when the field is changed. Support for this property varies by target platform.
Font	The font used in the caption for the control.
Usable	A USABLE control is visible to the user. USABLE is the default setting for this property. A NONUSABLE control does not draw on the form, but can be programmed to draw during execution.
Editable	If EDITABLE is selected, changes to the contents of a Field are allowed. If NONEDITABLE is selected, changes to the contents of a Field are not allowed.
SingleLine	If SINGLELINE is selected, only one line of text is visible. A single line field does not scroll horizontally or allow text beyond the end of the field. If MULTIPLELINES is selected, the Field can consist of multiple lines. The size of the Field needs to be adjusted to accommodate the multiple lines. The size of the Field will not automatically change. Each line of text is underlined. Support for this property varies by target platform.
Justification	Aligns the text to the left or right. Support for this property varies by target platform.

Table 28. Field properties (continued)

Property	Description
Max Chars	The maximum number of characters allowed to be entered into the Field. Users will not be able to enter more characters than the maximum specified. Support for this property varies by target platform.
Left	The distance from the left edge of the control to the screen edge. Use this property to manually set the position of the control.
Top	The distance from the top edge of the control to the screen edge. Use this property to manually set the position of the control.
Width	The width of the control in pixels. Use this property to manually set the size of the control.
Height	The height of the control in pixels. Use this property to manually set the size of the control.

Related tasks:

"Controls" on page 50

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

Graffiti shift properties

Allows text cases changes during data input using upward pen strokes. One upward stroke performs a case shift. Two upward strokes performs a shift lock. A third upward stroke disables the shift lock.

Support for this control varies by target platform.

Table 29. Graffiti Shift properties

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Left	The distance from the left edge of the control to the screen edge. Use this property to manually set the position of the control.
Top	The distance from the top edge of the control to the screen edge. Use this property to manually set the position of the control.
Width	The width of the control in pixels. Use this property to manually set the size of the control.
Height	The height of the control in pixels. Use this property to manually set the size of the control.

Related tasks:

"Controls" on page 50

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

Ink properties

Use this control to save a signature as a BLOB data type.

Support for this control varies by target platform.

Table 30. Ink properties

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Data sources	The name of the table that contains the Data field or column you want the control to read and write to. The list of available Data sources is based on the tables that have been imported into the project. See "Creating and importing a table definition file" on page 44 for more information.
Data field	The name of the column you want the control to read and write to. The list of available Data fields is based on the table selected in the Data source property.
Left	The distance from the left edge of the control to the screen edge. Use this property to manually set the position of the control.
Top	The distance from the top edge of the control to the screen edge. Use this property to manually set the position of the control.
Width	The width of the control in pixels. Use this property to manually set the size of the control.
Height	The height of the control in pixels. Use this property to manually set the size of the control.

Related tasks:

"Controls" on page 50

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

Label properties

A single-line read only text label.

Table 31. Label properties

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Caption	The text displayed in this control.
Font	The font used in the caption for the control.
Usable	A USABLE control is visible to the user. USABLE is the default setting for this property. A NONUSABLE control does not draw on the form, but can be programmed to draw during execution.
Background	The color of the background for this control. Support for this property varies by target platform.
Foreground	The color of the foreground for this control. Support for this property varies by target platform.
Left	The distance from the left edge of the control to the screen edge. Use this property to manually set the position of the control.
Top	The distance from the top edge of the control to the screen edge. Use this property to manually set the position of the control.
Width	The width of the control in pixels. Use this property to manually set the size of the control.
Height	The height of the control in pixels. Use this property to manually set the size of the control.

Related tasks:

"Controls" on page 50

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

List properties

Displays static table information. Initially populated by the Data Sources and Data Fields defined in the List Data Sources window. The Tap event may be used to perform database actions. The contents of the list can come from a table or a separately defined list.

Table 32. List properties

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.

Table 32. List properties (continued)

Property	Description
Data sources	The name of the table that contains the Data field or column you want the control to read and write to. The list of available Data sources is based on the tables that have been imported into the project. See Table definition files for more information.
List items	Click on the ellipse to the right of List items to open a window to enter custom values for the list. This property is often used to set the contents of the Pop-up trigger. Only one source is permitted for the List. The Data sources property cannot be used with the List items property.
Order By	Click on the ellipse to the right of Order By to open a window that will allow you to define the Data Source, Data Field, and ascending or descending order.
Distinct	Select YES to retrieve distinct rows, select NO to retrieve all rows. Setting the Distinct property makes the list control for reference only. A row selected in the list should not be used for displaying or updating the row (e.g. in a record details form displayed by tapping on a record displayed in the list control), because Mobile Application Builder cannot determine the exact row(s) to display or update when Distinct is used.
Limit	Define the number of rows that will be retrieved.
Selection criteria	Click on the ellipse to the right of the **Selection criteria** field to open a window where you can associate a column from a database table with a host variable that you have already defined. See Using host variables for more information. Support for this property varies by target platform.
Update host variable	Click on the ellipse to the right of Update host variable to open a window where you can define which previously-defined host variables will be updated with the list selection results. Support for this property varies by target platform.
Usable	A USABLE control is visible to the user. USABLE is the default setting for this property. A NONUSABLE control does not draw on the form, but can be programmed to draw during execution.
Background	The color of the background for this control. Support for this property varies by target platform.
Foreground	The color of the foreground for this control. Support for this property varies by target platform.
Left	The distance from the left edge of the control to the screen edge. Use this property to manually set the position of the control.
Top	The distance from the top edge of the control to the screen edge. Use this property to manually set the position of the control.
Width	The width of the control in pixels. Use this property to manually set the size of the control.
Height	The height of the control in pixels. Use this property to manually set the size of the control.

Related tasks:

"Controls" on page 50

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

Pop-up trigger properties

Allows the user to make a selection from multiple choices that drop down in a list when the control is tapped. The contents of the list are defined by linking a list control with the pop-up trigger. The order of actions starts when the pop-up trigger is clicked. Next the list is displayed and the user makes a selection. Before the pop-up trigger label changes to display the selection all events associated with the control are executed.

Support for this control varies by target platform.

Table 33. Pop—up trigger properties

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Caption	The title displayed to the right of the control.
Font	The font used in the caption for the control.
Usable	A USABLE control is visible to the user. USABLE is the default setting for this property. A NONUSABLE control does not draw on the form, but can be programmed to draw during execution.
AnchorLeft	If LEFTANCHOR is selected, the left bound of the control is fixed. LEFTANCHOR is the default setting for this property. If RIGHTANCHOR is selected, the right bound of the control is fixed. Support for this property varies by target platform.
ListID	Name of the List that is associated with the pop-up trigger. Select the List from the drop-down list.
Left	The distance from the left edge of the control to the screen edge. Use this property to manually set the position of the control.
Top	The distance from the top edge of the control to the screen edge. Use this property to manually set the position of the control.
Width	The width of the control in pixels. Use this property to manually set the size of the control.
Height	The height of the control in pixels. Use this property to manually set the size of the control.

Related tasks:

"Controls" on page 50
"Using pop-up triggers" on page 58

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

Push button properties

Allows the user one button choice from all push buttons within the same group ID. Push buttons inside the same group ID are mutually exclusive. The same events and actions are supported as for the button control.

Support for this control varies by target platform.

Table 34. Push button properties

Property	**Description**
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Caption	The text for the control.
Bitmap	The bitmap image that will appear on the control in an unselected state. You must add a bitmap image to your project before you can specify it here. In order for the bitmap to appear on the control, you must specify an image for both the Bitmap and Selected Bitmap properties. These images can be the same.
Selected Bitmap	The bitmap image that will appear on the control in a selected state. You must add a bitmap image to your project before you can specify it here. In order for the bitmap to appear on the control, you must specify an image for both the Bitmap and Selected Bitmap properties. These images can be the same.
Font	The font used in the caption for the control.
Usable	A USABLE control is visible to the user. USABLE is the default setting for this property. A NONUSABLE control does not draw on the form, but can be programmed to draw during execution.
AnchorLeft	If LEFTANCHOR is selected, the left bound of the control is fixed. LEFTANCHOR is the default setting for this property. If RIGHTANCHOR is selected, the right bound of the control is fixed. Support for this property varies by target platform.
GroupID	A unique number for each Push button that maps to the same column in a table. Mobile Application Builder automatically assigns the unique value for each Push button that links to the same column. A non-visible control is not considered part of the application and does not draw.
Left	The distance from the left edge of the control to the screen edge. Use this property to manually set the position of the control.

Table 34. Push button properties (continued)

Property	Description
Top	The distance from the top edge of the control to the screen edge. Use this property to manually set the position of the control.
Width	The width of the control in pixels. Use this property to manually set the size of the control.
Height	The height of the control in pixels. Use this property to manually set the size of the control.

Related tasks:

"Controls" on page 50

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

Repeat button properties

This button is selected repeatedly on a Pen Down event until the pen is lifted. The same events and actions are supported as for the button control.

Support for this control varies by target platform.

Table 35. Repeat button properties

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Caption	The text for the control.
Bitmap	The bitmap image that will appear on the control in an unselected state. You must add a bitmap image to your project before you can specify it here. In order for the bitmap to appear on the control, you must specify an image for both the Bitmap and Selected Bitmap properties. These images can be the same.
Selected Bitmap	The bitmap image that will appear on the control in a selected state. You must add a bitmap image to your project before you can specify it here. In order for the bitmap to appear on the control, you must specify an image for both the Bitmap and Selected Bitmap properties. These images can be the same.
Font	The font used in the caption for the control.
Usable	A USABLE control is visible to the user. USABLE is the default setting for this property. A NONUSABLE control does not draw on the form, but can be programmed to draw during execution.

Table 35. Repeat button properties (continued)

Property	Description
AnchorLeft	If LEFTANCHOR is selected, the left bound of the control is fixed. LEFTANCHOR is the default setting for this property. If RIGHTANCHOR is selected, the right bound of the control is fixed. Support for this property varies by target platform.
Frame	If FRAME is selected, a rectangular frame with rounded corners is drawn around the control. If NOFRAME is selected, no border is drawn around the control. If BOLDFRAME is selected, a bold rectangular frame with rounded corners is drawn around the control.
GroupID	A unique number for each Push button that maps to the same column in a table. Mobile Application Builder automatically assigns the unique value for each Push button that links to the same column. A non-visible control is not considered part of the application and does not draw.
Left	The distance from the left edge of the control to the screen edge. Use this property to manually set the position of the control.
Top	The distance from the top edge of the control to the screen edge. Use this property to manually set the position of the control.
Width	The width of the control in pixels. Use this property to manually set the size of the control.
Height	The height of the control in pixels. Use this property to manually set the size of the control.

Related tasks:

"Controls" on page 50

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

Scrollbar properties

Use with Fields or Lists.

Table 36. Scrollbar properties

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Usable	A USABLE control is visible to the user. USABLE is the default setting for this property. A NONUSABLE control does not draw on the form, but can be programmed to draw during execution.
CurrValue	Current value of the top of the scrollable car.

Table 36. Scrollbar properties (continued)

Property	Description
MinValue	The scroll car position when it is at the top of the Scrollbar. For most applications the default is zero.
MaxValue	The scroll car position when it is at the bottom of the Scrollbar.
PageSize	The number of lines to scroll at one time.
Left	The distance from the left edge of the control to the screen edge. Use this property to manually set the position of the control.
Top	The distance from the top edge of the control to the screen edge. Use this property to manually set the position of the control.
Width	The width of the control in pixels. Use this property to manually set the size of the control.
Height	The height of the control in pixels. Use this property to manually set the size of the control.

Related tasks:

"Controls" on page 50

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

Selector trigger properties

Displays a label outlined by a rectangular frame. It invokes Run Script events that allow the user to change the text displayed.

Table 37. Selector trigger properties

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Caption	The text displayed for the control.
Usable	A USABLE control is visible to the user. USABLE is the default setting for this property. A NONUSABLE control does not draw on the form, but can be programmed to draw during execution.
AnchorLeft	If LEFTANCHOR is selected, the left bound of the control is fixed. LEFTANCHOR is the default setting for this property. If RIGHTANCHOR is selected, the right bound of the control is fixed. Support for this property varies by target platform.
Left	The distance from the left edge of the control to the screen edge. Use this property to manually set the position of the control.

Table 37. Selector trigger properties (continued)

Property	Description
Top	The distance from the top edge of the control to the screen edge. Use this property to manually set the position of the control.
Width	The width of the control in pixels. Use this property to manually set the size of the control.
Height	The height of the control in pixels. Use this property to manually set the size of the control.

Related tasks:

"Controls" on page 50

Related reference information:

"Palette control icons" on page 109
"Control properties" on page 110

Resource properties

Mobile Application Builder provides many resources to use in your application. Set properties and events for the resources and use them in scripts.

Mobile Application Builder resource properties::

"Form properties" on page 129
"Alert properties" on page 130
"String properties" on page 130
"Menu properties" on page 131
"Menubar properties" on page 131
"Icon properties" on page 132
"Bitmap properties" on page 132

Related tasks:

"Project resources" on page 61

Related reference information:

"Control properties" on page 110

Form properties

Table 38. Form properties. Description

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Caption	The text that will appear on the form tab. This text will also appear in the Project pane.
Left	The distance from the left edge of the form to the screen edge.
Top	The distance from the top edge of the form to the screen edge.
Width	The width of the form in pixels. A typical Palm form width is 160.
Height	The height of the form in pixels. A typical Palm form height is 160.
Modal	If true, the user must respond to this form before continuing. The form will remain on top of all other forms until closed.
Menu ID	Enter the name of the menubar you want associated with the form.
Data source	The name of the DB2 table containing the information.
Data field	The name of the column that this form relates to.
Link source	The name of the DB2 table to be linked to.
Link field	The name of the column that contains the key to link to.
Selection criteria	Click … to the right of Selection criteria to open the Associate Columns with Host Variables window. Use the Associate Columns with Host Variables window to associate a column from a database table with a host variable that you have already defined. See Host variables for more information.
Help ID	The ID of the string to use when displaying help for a modal form. Support for this property varies by target platform.
Background	The color of the background for this control.
Foreground	The color of the foreground for this control.

Related tasks:

“Project resources” on page 61
“Forms” on page 48

Related reference information:

“Control properties” on page 110

Alert properties

Alerts are used to prompt a user for a response. An alert must be acknowledged by the user before they can proceed. Use scripting to reference the alert.

Support for alerts varies by target platform.

Table 39. Alert properties. Description

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Caption	The text displayed in the title. This text also appears in the Project pane.
Alert Type	Select one of the following alert types. Each type displays a different icon on the dialog. Support for this property varies by target platform. • CONFIRMATION • WARNING • INFORMATION • ERROR
Message	The message displayed in the alert.
Button Text	The text displayed on the alert button.
Default Button ID	The alert default button ID. The number of the button you want to display on the Alert. ID=0 means you want to display one button. ID=1 means two buttons, and so on. Support for this property varies by target platform.

Related tasks:

"Project resources" on page 61
"Creating a new alert" on page 62

Related reference information:

"Control properties" on page 110

String properties

Strings are used to display a string of characters. Use strings to display a message about the application, such as version number. Use scripting to reference the string.

Support for strings varies by target platform.

Table 40. String properties. Description

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
String	Text for the string. This text also appears in the Project pane.

Table 40. String properties (continued). Description

Property	Description
String File Name	Name of the file containing the string. Store the string file in the project directory.

Related tasks:

"Project resources" on page 61
"Creating a new string" on page 69

Related reference information:

"Control properties" on page 110

Menu properties

Menus or menu items are the individual functions that are accessible from the menubar and are not found anywhere else on the screen.

Table 41. Menu properties. Description

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Menu Item Text	The text for the menu item. This text will appear in the drop-down list of the menubar it is associated with. This text will also appear in the Project pane.
Accel Char	The shortcut keyboard character for the menu item. Enter one letter to represent the shortcut. Make sure that the shortcut is not used for any other menu item available at the time.

Related tasks:

"Project resources" on page 61
"Creating a new menu item" on page 67

Related reference information:

"Control properties" on page 110

Menubar properties

Menubars provide access to additional functions (menu items) that are not available anywhere else on the screen. The menubar is available when the menu button is tapped. Each form in your application can have multiple menubars.

Table 42. Menubar properties. Description

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
PullDowns	The text or caption that appears on the menu bar located at the top of each menu pull-down. This text also appears in the Project pane.
Menu Item IDs	Names of the menus to list under the menubar. Menus appear on the menubar pull-down list in the same order they are displayed on the properties page.

Related tasks:

"Project resources" on page 61
"Creating a new menubar" on page 68

Related reference information:

"Control properties" on page 110

Icon properties

Each Palm application has an icon that is displayed in the Palm application pane.

Support for icons varies by target platform.

Table 43. Icon properties. Description

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Icon File Name	The path and file name of the icon. It is recommended that you store the icon in the project directory.

Related tasks:

"Project resources" on page 61
"Creating a new icon" on page 66

Related reference information:

"Control properties" on page 110

Bitmap properties

Displays a bitmap image on a form. A bitmap image can be created using any external drawing program that allows you to save Bitmap images in black and white or direct color format (supported in Palm OS 4.0 or higher). Store bitmaps in your application's project directory. Display the bitmap by adding a form bitmap control to a form.

Support for bitmaps varies by target platform.

Table 44. Bitmap properties. Description

Property	Description
Name	A unique, system-generated name for the control. Not visible to users. Cannot be edited. Update scripts to use this control name.
Bitmap File Name	The name of the file that contains the bitmap. Store bitmaps in the project directory.
Compression Type	Select one of the following bitmap compression types. • NOCOMPRESS: do not compress bitmaps • COMPRESS • FORCECOMPRESS: force bitmap compression

Related tasks:

"Project resources" on page 61
"Creating and displaying bitmaps" on page 64

Related reference information:

"Control properties" on page 110

Event descriptions

Support for Mobile Application Builder events varies by control and by target platform.

Mobile Application Builder events

Events

Table 45. Event descriptions.

Event	Corresponding Palm event	Description
Control events		
Pen down	ctlEnterEvent	Pen down on control.
Exited	ctlExitEvent	Pen down in the control and lifted outside the control.
Repeat	ctlRepeatEvent	Pen down and will get another while pen is down. Used with Repeat button.
Tap	ctlSelectEvent	Pen down and lifted in control.
Field events		
Scroll	fldChangedEvent	Field scrolled as result of drag selecting.
Select	fldEnterEvent	Pen down in boundaries of the field.

Table 45. Event descriptions (continued).

Event	Corresponding Palm event	Description
Control events		
Height changed	fldHeightChangedEvent	Field height changed. Not normally handled by developer.
Form events		
Open	Form init	This event is executed each time a form is opened.
Barcode scanning form events		
Initialize scanning	ScannerInit	This event is associated with the InitializeScanner action described below. The event occurs just after the form Open event.
Decode Scanning	ScannerDecode	This event is associated with the DecodeScanTo action. The event occurs after a hard or a soft scan.
No Decode	NoDecode	This event occurs when the laser is triggered on a barcode and the barcode could not be successfully decoded. Any action can be associated with this event. If no action is specified, a default error message will be delivered to the user when this event occurs.
Close scanning	ScannerClose	This event is associated with the DeInitScanner action. The event occurs just after the form Close event.
List events		
Pen down	lstEnterEvent	Pen down in list.
Exited	lstExitEvent	Pen down in list and lifted outside of list.
Tap	lstSelectEvent	Pen down in list and lifted in same list.
Pop-up trigger events		
Pop-up	popSelectEvent	Item selected from a pop-up list.
Scroll events		
Pen down	sclEnterEvent	Pen down in control.
Exited	sclExitEvent	Pen down in the control and lifted outside.
Scroll	sclRepeatEvent	Pen down and held continuously.
Menu events		
Menu select	menuEvent	Pen down highlights menu item, lift pen in highlighted menu item.

Related tasks:

"Events, actions, and targets" on page 70

Related reference information:

"Action descriptions"

Action descriptions

Support for Mobile Application Builder actions varies by event and by target platform.

Mobile Application Builder actions

Actions

Table 46. Action descriptions.

Action	Description
No action	No action will occur.
Show	Displays the target form or dialog.
Clear form	Clears all of the fields, ink controls, or blob controls on a target form.
Snyc application	Initializes a synchronization with backend databases using the Sync Server.
Print form	Prints the contents of a form.
Return to previous form	Opens previous form.
Clear	Selectively clears target fields, blob controls, or ink controls on a form.
Choose Image	Opens a file dialog that lets the user choose an image from the file system, and displays this image on a Blob control. This makes the current image the active image, and if the use inserts the record then the chosen image is inserted.
Save Image	Allows the user to save the Blob image associated with a Blob control to the file system on the device.
Show Date Picker	Opens a dialog that allows the end user to choose a date using a date picker. The date is inserted into the target field.
Show Time Picker	Opens a dialog that allows the end user to choose a time using a time picker. The time is inserted into the target field.
Show in current state	Support for this action varies by target platform. Brings the target form into focus (i.e., makes the form visible) but the form will not be populated again. It will be displayed as it is.
Close application	Support for this action varies by target platform. Closes the application. On some target platforms, an application closes automatically when the user switches to a new application, so this action is not necessary.
Initialize Scanner	Initializes the scanner hardware for barcode scanning.
Soft Scan	This action should be used to fire the laser at the tap of a button or menu item. It is mandatory to include this action if the target device does not have a hardware button that triggers the laser beam to perform barcode scanning.
Close Scanner	Closes the barcode scanning awareness for the form.
Decode Scan To	Decodes the scanned data to a field.

Table 46. Action descriptions (continued).

Action	Description
Create record	Creates a record in the target table.
Delete record	Deletes a record in the target table.
Update record	Updates a record in the target table.
Move to next record	The next record in the target table is displayed.
Move to previous record	The previous record in the target table is displayed.
Move to first record	The first record in the target table is displayed.
Move to last record	The last record in the target table is displayed.
Run script	The target script is run.
Clear	Selectively clears target fields and ink controls on a form.

Related tasks:

"Events, actions, and targets" on page 70

Related reference information:

"Event descriptions" on page 133

Scripting support

Scripting support varies by target platform. Click a platform below for information about scripting support for that platform.

Mobile Application Builder scripting support::

"Scripting support for Java code platforms"
"Scripting support for Palm C code platforms" on page 137

Related tasks:

"Scripting in Mobile Application Builder" on page 74

Scripting support for Java code platforms

At this time, Mobile Application Builder supports only event script functions that return a void and has no capability to use any return values from a user-created function. However, you can use global scripts to write functions which return a value and later use them in event scripts.

To do more elaborate scripting, you need to manually edit the java code that is generated for the project and then run the batch file. As with most code generation tools, Mobile Application Builder does not understand or use any manual changes that have been made to the java code. Important: Any manual changes to the java code will be lost if you build the application again from within Mobile Application Builder.

Related tasks:

"Scripting in Mobile Application Builder" on page 74

Scripting support for Palm C code platforms

At this time, Mobile Application Builder supports only event script functions that return a void and has no capability to use any return values from a user-created function. However, you can use global scripts to write functions which return a value and later use them in event scripts.

To do more elaborate scripting, you need to manually edit the C code that is generated for the project and then run the make file. As with most code generation tools, Mobile Application Builder does not understand or use any manual changes that have been made to the C code. Important: Any manual changes to the C code will be lost if you build the application again from within Mobile Application Builder.

Related tasks:

"Scripting in Mobile Application Builder" on page 74

Printing support

Mobile Application Builder provides limited printing support for Palm OS devices. You can access printing capabilities by setting a default print solution for each project in **Preferences**. Support for printing varies by target platform.

Printing will be implemented on a form level with the following characteristics:

- The contents of all labels, fields, and list controls will be printed. Other controls will not print.
- Labels and fields or two lists can be printed next to each other. To do this, set the Y coordinates (the value of the property Top) for both controls to the same value. A tolerance of 2 pixels is allowed.
- Controls will be printed based on alignment on the form as follows:
  - From left to right
  - From top to bottom
- Exact printing behavior (i.e. wrapping versus truncating for long character strings) is determined by the printer you are using.

Supported print technologies

PalmPrint

PalmPrint is a Palm OS printing solution developed by Stevens Creek Software that allows you to print from your applications via infrared from all PalmOS 3.0/4.0 devices including the Palm III, V, VII, m100, and m500 as well as the Symbol SPT 1500 and 1700, IBM WorkPad, Handspring

Visor, and Sony Clie. The PalmPrint solution must be obtained from Stevens Creek Software before Mobile Application Builder print capability can be accessed using this solution.

Related tasks:

"Adding print capabilities to an application" on page 85

Related reference information:

"Printing API"

Printing API

Use the following print API calls to customize printing behavior.

Support for printing varies by target platform.

Table 47. Printing API calls, overview

Printing Solution	APIs
MAB_PRINT_VIA_PALMPRINT	MAB_Print_SetAlertIDForMessages
	MAB_Print_InitPrinter
	MAB_Print_Line
	MAB_Print_Buffer
	MAB_Print_ClosePrinter

Return Codes

Table 48. Return codes

Status	Description
MAB_PRINT_SUCCESS_CODE	The printing operation was successful.
MAB_PRINT_ERROR_UNKNOWN_CODE	The printing operation failed.

MAB_Print_SetAlertIDForMessages

Purpose: Use this function to set the Alert ID's for error messages. If this value is not set it will default to a MAB system alert called DBInformationAlert. To see error messages in custom alerts, set this before the call to any of the above API's.

Function Prototype:

```
void MAB_Print_SetAlertIDForMessages(UInt16 alertID)
```

Function Arguments:

Data type	Argument	Description
Word	alertID	The ID of the custom alert.

Returns: None

Usage

```
MAB_Print_SetAlertIDForMessages (Alert1);
```

MAB_Print_InitPrinter

Purpose: Use this function to initialize the printer for line-by-line printing. Pass the print library to be used for printing. This function is not needed for printing the buffer or printing the form.

Function Prototype:

```
MAB_PRINT_STATUS MAB_Print_InitPrinter(MAB_PrintLibrary lib)
```

Function Arguments:

Data type	Argument	Description
MAB_PrintLibrary	Lib	A valid printing library. Supported printing libraries are mentioned above.

Returns: MAB_PRINT_STATUS

Usage

```
MAB_Print_InitPrinter(MAB_PRINT_VIA_PALMPRINT);
```

MAB_Print_Line

Purpose: Use this function to print a line. The printer should be initialized before calling this API, using MAB_Print_InitPrinter. Calls to MAB_Print_Line should be followed by MAB_Print_ClosePrinter.

Function Prototype:

```
MAB_PRINT_STATUS MAB_Print_Line(char *buffer)
```

Function Arguments:

Data type	Argument	Description
Char *	Buffer	Data

Returns: MAB_PRINT_STATUS

Usage

```
MAB_Print_Line("Hello");
```

MAB_Print_Buffer

Purpose: Use this function to print a buffer. The buffer may contain multiple lines. You do not need to initialize or close the printer.

Function Prototype:

```
MAB_PRINT_STATUS MAB_Print_Buffer(MAB_PrintLibrary lib, char *buffer)
```

Function Arguments:

Data type	Argument	Description
MAB_PrintLibrary	Lib	A supported library name.
Char *	Buffer	Data

Returns: MAB_PRINT_STATUS

Usage

```
MAB_Print_Buffer(MAB_PRINT_VIA_PALMPRINT, "Hello");
```

MAB_Print_ClosePrinter

Purpose: Use this function to close the printer.

Function Prototype:

```
MAB_PRINT_STATUS MAB_Print_ClosePrinter()
```

Function Arguments: None.

Returns: MAB_PRINT_STATUS

Usage

```
MAB_Print_ClosePrinter();
```

Related tasks:

"Adding print capabilities to an application" on page 85

Related reference information:

"Printing support" on page 137

Barcode Scanning API

Use the following barcode scanning API calls to customize barcode scanning behavior. APIs are only supported for Palm barcode scanning applications, and are not supported for WinCE.

Support for barcode scanning varies by target platform.

Table 49. Barcode scanning API calls, overview

Barcode Scanning APIs
MAB_ScanSetAlertIDForMessages
MAB_ScanSetAimDuration
MAB_ScanSetTriggeringMode
MAB_ScanGetLastErrorCode
MAB_ScanGetLastErrorMessage
MAB_ScanDisableBarCodeType
MAB_ScanEnableBarCodeType
MAB_InitializeScanner
MAB_DoSoftScan
MAB_DecodeScan
MAB_ScanHandleNoDecode
MAB_DeInitScanner
MAB_ScanErrorHandler

Return Codes

Table 50. Return codes

Status	Description
MAB_SCAN_STATUS_OK	Success. No error returned.
MAB_SCAN_UNKNOWN_ERROR	An internal error occurred.
MAB_SCAN_BARCODE_NOT_SUPPORTED	This barcode type is either not supported or not enabled for this application.
MAB_SCAN_COMMUNICATIONS_ERROR	Could not communicate with the hardware.
MAB_SCAN_BAD_PARAM	One or more parameters used to initialize the scanner were out of range.
MAB_SCAN_BATCH_ERROR	Too many parameters used for initialization.
MAB_SCAN_NODECODE	The scanner could not decode the barcode successfully.
MAB_SCAN_SOFTSCAN_FAILED	The soft scan could not be initiated.
MAB_SCAN_CURRENT_FIELD_NOT_SCANAWARE	The current field is not scan-aware.

MAB_ScanSetAlertIDForMessages

Purpose: Use this function to set the Alert ID's for any error messages. If this value is not set it will default to a MAB system alert called DBErrorAlert. To see error messages in custom alerts, set this before the call to MAB_InitializeScanner.

Function Prototype:

```
void MAB_ScanSetAlertIDForMessages(Word alertID)
```

Function Arguments:

Data type	Argument	Description
`Word`	`alertID`	The ID of the custom alert.

Returns: None

Usage

```
MAB_ScanSetAlertIDForMessages(Alert1);
```

MAB_ScanSetAimDuration

Purpose: Use this function to set the length of time that the laser will remain on while scanning a barcode. The laser will remain on for the specified interval of time if a successful decode does not occur before that. The default value for this parameter is 20.

Function Prototype:

```
Boolean MAB_ScanSetAimDuration(UInt16 duration)
```

Function Arguments:

Data type	Argument	Description
`UInt16`	`Duration`	The length of time that the laser should remain on while scanning. The valid range for this parameter is between 5 and 99. A value of 20 here would imply that the laser should remain on for 2.0 seconds.

Returns: Boolean - true if the value passed was within range, otherwise false.

Usage

```
MAB_ScanSetAimDuration(20);
```

MAB_ScanSetTriggeringMode

Purpose: Use this function to set the triggering mode for the scanner. The value here determines whether the form will be enabled for a hard scan or a soft scan. A default value HOST is assumed, which will enable both types of scanning.

Function Prototype:

```
Boolean MAB_ScanSetTriggeringMode(UInt16 mode)
```

Function Arguments:

Data type	Argument	Description
`UInt16`	`Mode`	HOST: This should be set when the user needs to do both a soft scan and a hard scan in the application. LEVEL: This mode should be set when the user only needs hard scan in the application. The laser will be turned off in this case either on a successful decode or when the hardware trigger buttons are released.

Returns: None.

Usage

```
MAB_ScanSetTriggeringMode(HOST);
```

MAB_ScanGetLastErrorCode

Purpose: Use this function to obtain the error code resulting from the last MAB Scanner API call.

Function Prototype:

```
MAB_SCAN_STATUS MAB_ScanGetLastErrorCode()
```

Function Arguments: None.

Returns: `MAB_SCAN_STATUS`

Usage

```
int errorcode = MAB_ScanGetLastErrorCode();
```

MAB_ScanGetLastErrorMessage

Purpose: Use this function to obtain the error message corresponding to the error code resulting from the last MAB Scanner API call.

Function Prototype:

```
CharPtr MAB_ScanGetLastErrorMessage()
```

Function Arguments: None.

Returns: `CharPtr` - the message.

Usage

```
FrmCustomAlert(alertid,MAB_ScanGetLastErrorMessage(),"","");
```

MAB_ScanDisableBarCodeType

Purpose: Use this function to prevent any barcode types from being scanned by the application user. By default all of the barcode types are enabled in the application.

Function Prototype:

```
void MAB_ScanDisableBarCodeType(Word barCodeType);
```

Function Arguments:

Data type	Argument	Description
`Word`	`Barcode type`	Any of the barcode types supported by MAB.

Returns: None.

Usage

```
MAB_ScanDisableBarCodeType(barUPCA);
```

MAB_ScanEnableBarCodeType

Purpose: Use this function to enable any barcode type that is required in the application. Since by default all barcode types are enabled in the application, this function is useful when a barcode type has been disabled using `MAB_ScanDisableBarCodeType` somewhere earlier in the application.

Function Prototype:

```
void MAB_ScanEnableBarCodeType(Word barCodeType);
```

Function Arguments:

Data type	Argument	Description
`Word`	`Barcode type`	Any of the barcode types supported by MAB.

Returns: None.

Usage

```
MAB_ScanEnableBarCodeType(barUPCA);
```

MAB_InitializeScanner

Purpose: Use this function to initialize the scanner hardware. After this function is called the scanner starts drawing power. The Mobile Application Builder action ″Initialize Scanner ″ also uses this function.

Function Prototype:

```
MAB_SCAN_STATUS MAB_InitializeScanner()
```

Function Arguments: None.

Returns: MAB_SCAN_STATUS

Usage

```
if(MAB_InitializeScanner() == MAB_SCAN_STATUS_OK){
                // Successful initialization.
                }
                else{
                //display error - unsuccessful initialization
                }
```

MAB_DoSoftScan

Purpose: Use this function to initiate a soft scan. This function can be called from anywhere, as a response to a button or a menu item. This function is called as a result of the Mobile Application Builder action ″Do Soft Scan.″

Function Prototype:

```
MAB_SCAN_STATUS MAB_DoSoftScan();
```

Function Arguments: None.

Returns: MAB_SCAN_STATUS

Usage

```
if(MAB_DoSoftScan() == MAB_SCAN_STATUS_OK){
                //successful initiation
                }
                else{
                //could not initiate a soft scan.
                }
```

MAB_DecodeScan

Purpose: Use this function to decode a soft scan or a hard scan that has just been initiated. The function should be used only in those scripts that are handlers for the event scanDecodeEvent. This function is used by the Mobile Application Builder action ″Decode Scan To.″

Function Prototype:

```
CharPtr MAB_DecodeScan (void);
```

Function Arguments: None.

Returns: `CharPtr` - Returns the decoded barcode, if the decode was successful. Returns NULL if the barcode could not be decoded. Use `MAB_SCAN_STATUS` to get the error code for the last operation.

Usage

```
...
                szText = MAB_DecodeScan();
                if(szText){
                //successful decode
                }
                else{
                //no decode
                }
```

MAB_DeInitScanner

Purpose: Use this function to de-initialize the scanner hardware. This function is called as a result of the Mobile Application Builder action ″DeInitScanner.″

Function Prototype:

```
MAB_SCAN_STATUS MAB_DeInitScanner(void);
```

Function Arguments: None.

Returns: `MAB_SCAN_STATUS`

Usage

```
...
                szText = MAB_DeInitScanner ();
                if(szText){
                //deinit successful
                }
                else{
                //deinit failed.
                }
```

MAB_ScanErrorHandler

Purpose: Use this function to display an error message for the status received as a result of the last performed action. This function makes use of the AlertID set using the function `MAB_ScanSetAlertIDForMessages`.

Function Prototype:

```
Boolean MAB_ScanErrorHandler(UInt16 status);
```

Function Arguments:

Data type	Argument	Description
`UInt16`	`status`	Any of the aforementioned status.

Returns: Boolean - Always returns true.

Usage

```
MAB_ScanErrorHandler(MAB_SCAN_NODECODE);
```

Related tasks:

"Adding barcode scanning capabilities to an application" on page 88

Related reference information:

"Supported 1-dimensional barcode types for Palm applications"

Supported 1-dimensional barcode types for Palm applications

Table 51. Supported 1-dimensional barcode types for Palm applications

barCODE39, barUPCA, barUPCE, barUPCE1, barEAN13, barEAN8, barD25, barI2OF5, barCODABAR, barCODE128, barCODE93, barTRIOPTIC39, barUCC_EAN128, barMSI_PLESSEY, barUPCE1, barBOOKLAND_EAN, barISBT128, barCOUPON, barPDF417

Related tasks:

"Adding barcode scanning capabilities to an application" on page 88

Related reference information:

"Barcode Scanning API" on page 141
"Supported barcode types for WinCE applications"

Supported barcode types for WinCE applications

Table 52. Supported 1-dimensional barcode types for WinCE applications

EAN8 EAN13 CODE39 UPCA UPCE0 UPCE1 MSI CODE93 CODE128

Related tasks:

"Adding barcode scanning capabilities to an application" on page 88

Related reference information:

"Supported 1-dimensional barcode types for Palm applications" on page 147

Troubleshooting

If you are experiencing trouble building or testing your application, review the list of possible solutions below.

Troubleshooting

Palm development environment setup errors
Some common errors that may occur when setting up the Palm development environment.

Table 53. Palm development environment setup errors. Description

Symptom	Possible Solution
Application will not build.	Path not set property. Ensure path contains: • <cygwin install path>\cygwin_b20\hi586_cygwin32\bin • <prc-tools install path>\bin
Make fails with error like ″:cpp - file or directory not found″	MAKE_MODE not set to ′UNIX′. Environment variable MAKE_MODE=UNIX must be set so Make will execute commands with sh.exe instead of command.com.
Make fails with error like ″:m68k-palmos-gcc - file or directory not found″	Mounts not set up properly. Compiler cannot find palmdev tools. Run bash shell and type mount to see mounts. Ensure all required mounts are there.

Table 53. Palm development environment setup errors (continued). Description

Symptom	Possible Solution
Compiler error - cannot find PalmOS.h	SDK is not installed, specified, or not setup properly. • Ensure SDK is installed in <palmdev location>/sdk-##, where ## is SDK version number, i.e./sdk-4.0. • Specify the version of the SDK you want the compiler to use with the -palmos## flag, where ## is the SDK version number. • Create a symbolic link /sdk in the <palmdev location> directory which points to the SDK you want to use by default.
Compile error indicating compiler cannot find libraries or tools.	GCC_EXEC_PREFIX still set from old GNU toolset. Unset the GCC_EXEC_PREFIX environment variables.
DB2 Everyplace application builds without error, but crashes in emulator or on Palm device.	Linking old DB2eGNU.o which is compiled by 0.5.0 toolset. Need DB2eGNU.o compiled with new PRC-Tools.

Palm build errors

A Mobile Application Builder error message appears stating that the build did not complete successfully. View the message log.

Table 54. Palm build errors. Description

Symptom	Possible Solution
In the message log, file PalmOS.h was not found.	The SDK is either not installed properly or is the wrong level. The SDK level should be 4.0.
″MAB.rcp″ followed by an error condition appears in the message log	• You may be using a back level resource compiler. To determine the resource compiler version: 1. Open a DOS or Windows NT Command Prompt. 2. Type `pilrc` 3. It should say `version 2.4`. • A step may have been skipped when installing GNU PRC-Tools. Refer to the GNU PRC-Tools installation information in Setup and Configuration. • If you are unable to resolve the problem, save your Mobile Application Builder file (*.mab) and message log for errors that you cannot resolve.
″undefined″ error in a function appears in the message log	• An element may have been renamed and not updated in a script. The old element name is still being referenced in a script. • Set or Get text functions must be added to script using the Paste Part window.
″error : FORM, MENU, ALERT, VERSION, STRING, CATEGORIES, APPLICATIONICONNAME, APPLICATION, BITMAP, SMALLICON, ICON, TRAP, FONT or TRANSLATION expected ... make: *** [bin.stamp] Error 1″ appears in the message log	PilRC V2.8 has not been installed properly.

Table 54. Palm build errors (continued). Description

Symptom	Possible Solution
″MAB.rcp(27): error: Bitmap not monochrome″ appears in the message log	Palm pre-version 4.0 only supports black and white or monochrome images. Make sure the image you are using has been saved in this format.
In the message log, bitmap file not found. Displays path to file without slashes.	Store bitmaps in your application’s local project directory. Do not enter a path to the file, only enter the file name.
″Unable to find pabappic.bmp″ appears in the message log	This unlikely error may occur if an application that was created pre-version 8.1 is migrated to version 8.1. References to `pabappic.bmp` in the existing application should be changed to `MAB_Icon.bmp`. The Mobile Application Builder migration utility is designed to correct this error.
″request for member ’Column1’ in something not a structure or union″ appears in the message log	This error may occur if a you are accessing the internal data structure of Mobile Application Builder. For Mobile Application Builder v8.1 all of the buffer variable definitions have been changed from static variable of table structure type to pointer of type table structure. To fix this problem, change the way data members of structure are accessed, i.e., change ″.″ to reference ″->″.
General build error	If you are unable to resolve the problem, save your Mobile Application Builder file (*.mab) and message log for errors that you cannot resolve.

Palm application errors

Table 55. Palm application errors

Symptom	Possible Solution
When multiline data is entered on a Windows machine and then synced to Palm and displayed in a Palm multiline field, a junk character ’[]’ is displayed on the screen of the device.	This problem exists due to different approaches used by Windows and non-Windows environments for new line characters. Windows uses two characters, Carriage Return (Cr) and Line Feed (Lf), while non-Windows platforms use only Lf for this purpose. Since Palm OS does not recognize the Cr character, it displays the ’[]’ character. **Fix:** If you are seeing this character in your application, you can set a flag to eliminate this character during code generation. Select Global Definitions in the Project pane, then click […] next to Preprocessor Statements in the Properties and Events pane to open a window where you can type preprocessor statements. In this window, type `#define TRIMCR true`. This is a case sensitive statement so type it as is. **Limitations:** This is only a one way process, meaning that only incoming data is checked for the ’/r’ character. Data saved in a table will use the Palm new line character, i.e. ’/n’.

Symbian Crystal build errors

A Mobile Application Builder error message appears stating that the build did not complete successfully. View the message log.

Table 56. Symbian Crystal build errors

Symptom	Possible Solution
Application does not build.	If you have another version of Perl installed, the application build process may fail if the other Perl version appears before the Symbian SDK's Perl in the PATH environment variable. If this happens you will have to set the PATH appropriately. Go to **Start -> Control Panel-> System -> Advanced -> Environment Variables -> System Variables**. Select PATH and click **Edit**. Verify that `C:\Perl\bin` appears before any other Perl version.
General build error	If you are unable to resolve the problem, save your Mobile Application Builder file (*.mab) and message log for errors that you cannot resolve.

Symbian UIQ build errors

A Mobile Application Builder error message appears stating that the build did not complete successfully. View the message log.

Table 57. Symbian Crystal build errors. Description

Symptom	Possible Solution
Error in the message log states: ″Internal error - cpp.exe has returned the error code 33″	Move the `aiftool.rh` file into the `Symbian\UIQ_70\epoc32\include` directory. You can extract it from com.symbian.api.GT-shared_0_0_70.sdkpkg IIRC.

Unable to test Palm application

Table 58. Palm test errors

Symptom	Possible Solution
Palm Emulator does not open from Mobile Application Builder interface	A step may have been skipped when installing the Palm OS Emulator. Refer to emulator installation information in Setup and Configuration.
Cannot get two different applications to work at once	The Palm application IDs for the application may be the same. Refer to "Application IDs" on page 42 for more information.
No data in the tables when running the application	The tables used during development may not actually be on the device. See *DB2 Everyplace Sync Server Administration Guide* for more information on loading tables onto mobile devices.
Able to open application, start to use it, and an error message appears (SQLSTATE=####)	See *Application Development Guide*, installed with the DB2 Everyplace engine, for SQL error code information.
Palm emulator or device not functioning as expected	Refer to the documentation that comes with your the device for more information.

Other general errors

Table 59. General errors

Symptom	Possible Solution
Table changes not available in project	If a table definition file was changed after it was first imported into the project, the changes will not appear. Delete the changed table definition file from your project and import it again. See Working with data sources for more information.
In New project window or Rename project window, **OK** button not available.	Make sure you have not violated any naming restrictions for the project name, project directory, or application name. See "Naming conventions for projects and applications" on page 34 for more information.

Related tasks:

"MAB version 8.1.4 setup and configuration" on page 12
"Building a Mobile Application Builder application" on page 91
"Testing a Mobile Application Builder application" on page 96

Links to other sources of information

Below are links to other sources of information that may be helpful to review.

- DB2 Everyplace information
  - DB2 Everyplace Web Site
  - DB2 Everyplace forum
  - DB2 Everyplace library
  - DB2 Everyplace support
- Palm OS information
  - Palm OS documentation
  - Palm OS Creator ID database
- Support Software
  - Palm OS Emulator (POSE)
  - Palm OS SDK 4.0 for Windows
  - PalmPrint FAQ's (Steven's Creek Software)
- Barcode Scanning
  - Symbol Developer Zone
  - Symbol Development SDK download site
- Java tools/information
  - IBM developerWords Java technology zone
  - java.sun.com
  - Insignia
- Symbian information
  - Symbian UID information
  - Forum Nokia

– Sony Ericsson Mobility World

Chapter 3. Examples

This section provides examples for Mobile Application Builder.

DB2 Everyplace sample applications

DB2 Everyplace includes sample applications for each target platform, as described in the following tables.

Table 60. DB2 Everyplace sample applications by target platform

Target platform	Type	Sample applications
Palm	Client	• DB2eCLP • VNurse • NurseInit • DB2eAppl • DB2eJavaCLP • Sync GUI
Palm	MAB	• PersonList • VNApp • VNPlus
Symbian 6	Client	• DB2eCLP • VNurse • NurseInit • DB2eAppl • DB2eJavaCLP • Sync GUI • Common
Symbian 6	MAB	• PersonList • VNApp
Symbian 7	Client	• DB2eCLP • NurseInit • Sync GUI • PersonList
Symbian 7	MAB	• PersonList • VNApp
WinCE	Client	• DB2eCLP • VNurse • VNurseInit • DB2eAppl • DB2eJavaCLP • Sync GUI
WinCE	MAB	• PersonList • VNApp

Table 60. DB2 Everyplace sample applications by target platform (continued)

Target platform	Type	Sample applications
Win32	Client	- DB2eCLP - DB2eAppl - DB2eJavaCLP
MIDP	Client	- DB2eCLP - Dump.java
EmbeddedLinux and Neutrino (including Sharp Zaurus)	Client	- Command Line CLP - NurseInit - Sync GUI - PersonList
EmbeddedLinux and Neutrino (including Sharp Zaurus)	MAB	- PersonList - VNApp

Table 61. DB2 Everyplace sample applications by version and location

Application	Description	Version	Directory
DB2eCLP	Utility file	EE/DE/SDK	`Client\platform\database\lang\DB2eCLP`
VNurse	Up and running sample, combination of NurseInit and Nurse	EE/DE/SDK	`Client\platform\database\lang\Samples\VNurse`
Sync GUI - `testicync.exe` - `isyncui.prc` - `isyncui.exe` - `isync.sis` - `goISync`	Up and running synchronization sample	EE/SDK	`Client\platform\sync\lang`
ClientAPIsample - ISyncSample - GoISyncConsole	Synchronization Java samples	SDK	`Clients\clientapisample\Java_API`
NurseInit	Creates and populates tables	SDK	`SDK\Samples\platform\language\NurseInit`
Common - `DB2eEng.cpp` - `DB2eEng.h`	A code sample for Symbian 6	SDK	`SDK\Samples\Symbian6\language\Common`
JDBCSample - `DB2eAppl` - `DB2eJavaCLP`	JDBC sample application	SDK	`Clients\platform\database\JDBC\lang`
MIDP `Dump.java`	MIDP sample application com.ibm.mobileserves.demo	SDK	`Clients\MIDP\lang\Samples`
PersonList	Sample MAB application	SDK	`SDK\ApplicationBuilder\Projects\Samples\platform`
VNApp	Sample MAB application	SDK	`SDK\ApplicationBuilder\Projects\Samples\platform\`
VNPlus	Sample MAB application	SDK	`SDK\ApplicationBuilder\Projects\Samples\platform\`

Example of a CREATE TABLE statement

Below is an example of the CREATE TABLE statement. This is the CREATE TABLE statement used in `VNPerson.ddl`.

Table 62. CREATE TABLE statement example

```
CREATE TABLE VNPerson
(ID Char(9) PRIMARY KEY,
Name Varchar(40),
Address Varchar(50),
City Varchar(25),
HomePhone Varchar(20),
WorkPhone Varchar(20),
MobilePhone Varchar(20))
```

Related tasks:

“Tables and data sources” on page 43

Defining database queries

Application Builder projects can have multiple tables in each application. These tables can be joined to display information from up to two tables at once. Because of this it is important to be familiar with each table’s structure and content.

The Visiting Nurses sample application best illustrates several examples of table joining functions and database queries possible in Mobile Application Builder.

Database query tasks/examples::

“Opening the Visiting Nurses sample application”
“Guidelines for defining database queries in your application” on page 158
“Example query — Select all records” on page 158
“Example query — Select a specific record” on page 159
“Example query — Join two tables” on page 160
“Example query — Select a record with matching column values” on page 160

Related tasks:

“Tables and data sources” on page 43

Opening the Visiting Nurses sample application

The Visiting Nurses sample application best illustrates several examples of table joining functions and database queries possible in Mobile Application Builder.

To open the Visiting Nurses application::

1. Click **File —> Open Project**. The Open Project window opens.

2. Navigate to the Application Builder installation directory and select the `VNApp.pab` file in the `...\Projects\Samples\<target OS>\VNApp` directory.
3. Click **Open**.

Related tasks:

"Tables and data sources" on page 43

Guidelines for defining database queries in your application

Database queries are generated for your application from the properties you have specified for forms, fields, and lists. The Data source and Data field properties define the database tables and columns used in queries, and the Link source and Link field properties, if specified, define how to parameterize and join tables in a SQL SELECT statement.

A recommended guideline is to keep your forms simple by only associating one query with a form. Defining multiple join queries on the same form will not produce the desired results. This situation can occur if you specify Link source and Link field properties for both a list and a field on your form.

Related tasks:

"Tables and data sources" on page 43

Generating queries with multiple parameters

Forms and list controls use the Selection Criteria property. You can use this property to generate a SELECT query that contains multiple parameters in the WHERE clause of the query. The property setting allows users to include any number of columns in the WHERE clause with a parameter marker, and use a host variable to bind the value of the parameter at run time. Typically, the value of the host variable will be set using the Update Host Variable property setting before it is used for binding a parameter

Related tasks:

"Using host variables" on page 82
"Setting the Selection criteria for a list control or choice list control" on page 85

Example query — Select all records

This example query illustrates how to selects all records from a table.

This query is used in the PersonList sample, which is demonstrated in the Quick Tour: Opening, building, and testing a Palm application. The query retrieves each of the records in the VNPerson table. By associating a database table and columns with the form fields, the example query below is generated.

Example SQL:

```
SELECT ID, Name, Address, City, HomePhone, WorkPhone, MobilePhone FROM
VNPerson
```

Table 63. PersonList fields with corresponding Data source and Data field properties

Field Label	Data source	Data field
ID	VNPerson	ID
Name	VNPerson	Name
Address	VNPerson	Address
City	VNPerson	City
Home Phone	VNPerson	HomePhone
Work Phone	VNPerson	WorkPhone
Mobile Phone	VNPerson	MobilePhone

Related tasks:

"Tables and data sources" on page 43
"Modifying the SQL for a form event" on page 81

Example query — Select a specific record

This example query illustrates how to select a specific record from a table where a column value equals a parameter value.

This query is used in the Visiting Nurse sample on the Patient Information form. This query selects the record from the VNPerson table where ID equals the current value of ID. In the Visiting Nurse application, the current value of ID is equal to the ID of the selected patient on the Schedule form.

By associating a database table and columns with the form fields, the example query below is generated. Additionally, by specifying values for the Form properties, Data source and Data field, the query includes a search condition which restricts the query to the VNPerson record for the current ID.

Example SQL:

```
SELECT ID, Name, Address, City, HomePhone, WorkPhone, MobilePhone FROM
VNPerson WHERE ID=?
```

Table 64. Patient Information form and fields with corresponding Data source and Data field properties

Form/Field Label	Data source	Data field
Patient Information Form	VNPerson	ID
Name	VNPerson	Name
Address	VNPerson	Address
City	VNPerson	City
Home Phone	VNPerson	HomePhone
Work Phone	VNPerson	WorkPhone
Mobile Phone	VNPerson	MobilePhone

Related tasks:

"Tables and data sources" on page 43
"Modifying the SQL for a form event" on page 81

Example query — Join two tables

This example query illustrates how to join two tables.

This query is used on the Schedule form in the Visiting Nurse sample on the first form, Schedule. This query joins records in the VNSchedule and VNPerson tables where the PatientID in the VNSchedule table equals the ID in the VNPerson table. This search condition is specified by defining the Link source and Link field values for the VNSchedule.PatientID column.

Note: To properly join these two tables, we need to specify the VNSchedule.PatientID and VNPerson.ID columns because they contain the unique identifiers for the patient. However, we do not want to display this ID in the list, so we set the Usable attribute to UNUSABLE.

By default, queries retrieve all columns defined for a table, not just those columns defined in your application.

Example SQL:

```
SELECT VNSchedule.PatientID, VNSchedule.Time_C, VNPerson.ID,VNPerson.Name,
VNPerson.Address, VNPerson.City, VNPerson.HomePhone, VNPerson.WorkPhone,
VNPerson.MobilePhone FROM VNSchedule, VNPerson WHERE
VNSchedule.PatientID=VNPerson.ID
```

Table 65. Schedule form list control Data source, Data field, Link source, and Link field properties

List1Form1	Data source	Data field	Link source	Link field
Column 1	VNSchedule	Time_C		
Column 2	VNPerson	Name		
Column 3	VNSchedule	PatientID	VNPerson	ID

Related tasks:

"Tables and data sources" on page 43
"Modifying the SQL for a form event" on page 81

Example query — Select a record with matching column values

This example query illustrates how to select a record from a table where a column value in one table equals a column value in another table.

This query is used in the Visiting Nurse sample on the Contact Information form. This query selects the record from the VNPerson table where ID equals the current value of VNContact.ContactID. In the Visiting Nurse application, the current value of ContactID is equal to the ContactID of the selected Contact on the Emergency Contacts form.

This query is identical to the previous one, but is executed with a different parameter value. In the previous query the VNPerson.ID parameter equals the current value. In this query the VNPerson.ID parameter equals the value of the current VNContact.ContactID. This parameter specification is specified by the Data source, Data field, Link source, and Link field properties of the form.

Example SQL:

```
SELECT ID, Name, Address, City, HomePhone, WorkPhone, MobilePhone FROM
VNPerson WHERE ID=?
```

Table 66. Contact Information form and fields with corresponding Data source, Data field, Link source, and Link field properties

Form/Field Label	Data source	Data field	Link source	Link field
Contact Information Form	VNPerson	ID		
Contact Information Form			VNContact	ContactID
Name	VNPerson	Name		
Address	VNPerson	Address		
City	VNPerson	City		
Home Phone	VNPerson	HomePhone		
Work Phone	VNPerson	WorkPhone		
Mobile Phone	VNPerson	MobilePhone		

Related tasks:

"Tables and data sources" on page 43
"Modifying the SQL for a form event" on page 81

Using DB2 CLI function in scripts

The code in this section assumes the following table definition:

```
CREATE TABLE blobs (id INT, data BLOB(8192))
```

It may be desired to insert and retrieve table column values in scripts which you attach to forms or controls. Below are two routines which show how this can be done for BLOB data types using DB2 CLI function. Once a pointer to BLOB is retrieved, the Palm API WinDrawBitmap can be used to display the BLOB value.

```
// ************************************************
//  Function to insert data into a BLOB data type
// ************************************************
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;

   // Prepare the insert.
   if (rc = SQLPrepare(stmt, SQL_insert, SQL_NTS))
      return myerror("SQLPrepare (insert)"), rc;

   // Bind the blob id.
   if (rc = SQLBindParameter(stmt, 1,
      SQL_PARAM_INPUT, SQL_C_SHORT, SQL_INTEGER, 0, 0,
      &id, 0, NULL))
         return myerror("SQLBindParameter (insert/1)"), rc;

   // Bind the blob size.
   if (rc = SQLBindParameter(stmt, 2,
      SQL_PARAM_INPUT, SQL_C_SHORT, SQL_INTEGER, 0, 0,
      &size, 0, NULL))
         return myerror("SQLBindParameter (insert/2)"), rc;

   // Bind the blob data.
   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;

   // Now perform the insert.
   if (rc = SQLExecute(stmt))
      return myerror("SQLExecute (insert)"), rc;

   return SQL_SUCCESS;
}

// ************************************************
//  Function to get pointer to BLOB
// ************************************************
char *DB2e_get_blob(Word id)
{
   // SQL paramaters 1
   static SQLCHAR SQL_select[] = "select size,
 blob from blobs where id = ?";

   Word       size;
   char       *blob;
   SQLINTEGER SQL_size;

   // Prepare the select.
   if (SQLPrepare(stmt, SQL_select, SQL_NTS))
      return myerror("SQLPrepare (select)"), (char *)NULL;
   // Bind the blob id.
   if (SQLBindParameter(stmt, 1,
      SQL_PARAM_INPUT, SQL_C_SHORT, SQL_INTEGER, 0, 0,
      &id, 0, NULL))
           return myerror("SQLBindParameter (select/1)"),
   (char *)NULL;

   // Now perform the select.
   if (SQLExecute(stmt))
      return myerror("SQLExecute (select)"), (char *)NULL;

   // Try to fetch the blob.
   switch (SQLFetch(stmt)) {
     case SQL_SUCCESS:
          break;
     case SQL_NO_DATA_FOUND:
          return (char *)NULL;
     default:
          return myerror("SQLFetch"), (char *)NULL;
     }

   // Read the size column.
   if (SQLGetData(stmt, 1,
      SQL_C_SHORT,
        &size, 0,
        NULL))
           return myerror("SQLGetData (size)"), (char *)NULL;

   if (!(blob = MemPtrNew(size)))
      return (char *)NULL;

   // Read the blob column.
   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;
}
```

Related concepts:

"Scripting support" on page 136

Related tasks:

"Scripting in Mobile Application Builder" on page 74

Releasing memory for variables

If you allocate memory for a variable and then you do not release it, you can get memory leaks using the Palm Emulator v3.5 or higher. Some MAB-generated functions allocate memory and return a pointer to the allocated memory. If you are using these functions in scripts, you must explicity release the allocated memory to avoid memory leaks.

The example below shows a method for releasing allocated memory.

```
CharPtr commentTxt;
commentTxt = PAB_getFieldText(Field8Form7);
.
.
.
.
// release memory allocated for commentTxt
if (commentTxt)
 MemPtrFree(commentTxt) ;
```

Related concepts:

"Scripting support" on page 136

Related tasks:

"Scripting in Mobile Application Builder" on page 74

Index

Notices

Index entries

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country/region or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country/region where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product, and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information that has been exchanged, should contact:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems, and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs, in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (*your company name*) (*year*). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _*enter the year or years*_. All rights reserved.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both, and have been used in at least one of the documents in the DB2 UDB documentation library.

ACF/VTAM
AISPO
AIX
AIXwindows
AnyNet
APPN
AS/400
BookManager
C Set++
C/370
CICS
Database 2
DataHub
DataJoiner
DataPropagator
DataRefresher
DB2
DB2 Connect
DB2 Extenders
DB2 OLAP Server
DB2 Universal Database
Distributed Relational Database Architecture
DRDA
eServer
Extended Services
FFST
First Failure Support Technology
IBM
IMS
IMS/ESA
iSeries
LAN Distance
MVS
MVS/ESA
MVS/XA
Net.Data
NetView
OS/390
OS/400
PowerPC
pSeries
QBIC
QMF
RACF
RISC System/6000
RS/6000
S/370
SP
SQL/400
SQL/DS
System/370
System/390
SystemView
Tivoli
VisualAge
VM/ESA
VSE/ESA
VTAM
WebExplorer
WebSphere
WIN-OS/2
z/OS
zSeries

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the documents in the DB2 UDB documentation library:

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.