# SYBASE®

An **SAP** Company

Sybase Unwired WorkSpace - Device
Application Development

# Sybase Unwired Platform 1.5.5

# Contents

# Product Task Flow

Use Sybase® Unwired Platform to develop mobile applications, and to manage the production environment. Understanding the end-to-end product task flow enables you to use Unwired Platform strategically in your enterprise.

**Developers** Use Sybase Unwired WorkSpace to develop mobile applications. The developer's license includes everything necessary to develop and test your creations—access to sample or external data sources, access to the Eclipse development environment, API classes, and Unwired Server. The basic steps for creating a mobile application include:

1. Create a connection profile to a structured or unstructured data source.
2. Create a connection profile to Unwired Server.
3. Create device applications or mobile workflows:
   a. Create mobile business objects.

   Use Unwired WorkSpace to create a project container, then create one or more mobile business objects (MBOs). Mobile business objects contain the business logic, operations (create, update, delete, and other), attributes, and relationships for the mobile application. For example, an MBO may include the business logic for creating, editing, and deleting customer records. You can create an MBO by dragging and dropping an object from the data source, or using the creation wizard and then bind the MBO to a data source. Alternatively, you can create an MBO and defer binding to a data source, or create a local business object. Generate replication-based or message-based code, and deploy to Unwired Server.
   b. Create a device application, which is the user interface that runs on a mobile device.
      - Generate client object API code in Eclipse, then develop the device application in a native IDE.
      - You can use the Device Application Designer to create a prototype BlackBerry® or Windows Mobile device application using the graphical user interface, drag-and-drop, and stock screens.
      - Use Mobile Workflow Forms Editor to develop a message-based workflow package.
      - Identify synchronization keys, set up personalization, and implement error handling.
4. Deploy the mobile application from the development environment to Unwired Server.
5. Deploy the device application to an emulator or mobile device and test.

**System Administrators** Use the Sybase Control Center administrative console, a Web-based user interface to configure and deploy mobile applications and workflow packages from Unwired Server to the production environment, and to manage the production environment. Multiple users can use the administrative console. Steps for deploying the mobile application in a production environment include:

1. Package the mobile application for deployment.

   Create a mobile application package on the Unwired Server. Configure MBO server-side parameters, set up personalization, and perform role mapping for the production environment.

2. Deploy the mobile application package.

   Once configured, deploy the mobile application package on Unwired Server. Once deployed, users can access the mobile application from mobile devices. Mobile applications can be pushed to the device or scheduled for deployment. Unwired Server manages synchronization between the data source and the mobile device.

   Optionally use Afaria® with Unwired Platform to provision mobile applications, and manage devices and users.

**Mobile Device Users** Use mobile devices (including smartphones, laptops, handheld devices, and notebooks) to access mobile applications.

From the mobile device, log in to the mobile application; navigate the user interface; synchronize data and applications through Unwired Server to the data source; and create, update, and delete data records and transactions.

**See also**
- *API Documentation* on page 333

# About Unwired Platform

Unwired Platform provides an integrated solution for mobilizing your enterprise.

Unwired Platform functions include:

- Data sources – connect to your data sources, such as databases, packaged applications, and service data sources.
- Data center – manage data connections between your data sources and the mobile applications that are deployed to devices.
- Network – manage messaging between data sources and mobile devices.
- Administration console – manage security, access, authorization, and deployment of device applications to mobile devices from Sybase Control Center.
- (Optional) Afaria – provision users, device clients, and mobile applications.
- Unified development tool – develop mobile applications from the Eclipse development environment. Develop device applications using a native IDE, such as Microsoft Forms Designer from the Visual Studio development environment, or the client object APIs.

  You can also develop mobile workflows using Mobile Workflow Forms Editor from the Eclipse development environment.

**Figure 1: Unwired Platform functions**
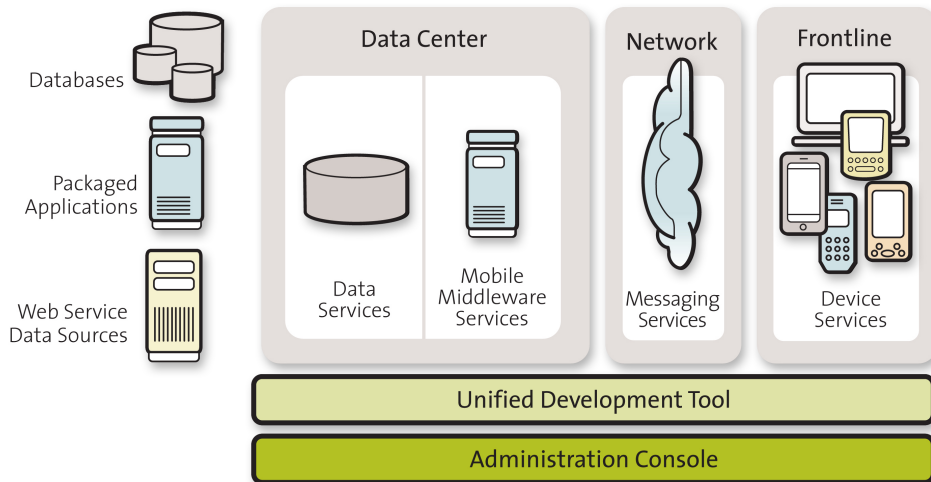


**See also**
- *Tutorials* on page 7
- *Samples* on page 7

## Solution Architecture for Mobilization

Unwired WorkSpace provides an integrated solution for mobilizing your enterprise.

Unwired WorkSpace provides the tools you need to take your enterprise data and applications into the field to mobile workers quickly and without programming. Create dynamic, secure mobile applications; and deploy, manage, and monitor them from a Web-based administrative console.

Mobile workers, equipped with a variety of mobile devices, laptops, and notebooks, have access to current data from any place, any time. Your sales force has access to order processing; delivery personnel have access to inventory and customer data; technicians have access to equipment data; and executives have access to financial and sales information. Transactions are handled remotely and securely in real time.

Unwired WorkSpace provides the development environment that lets you develop mobile applications and device applications using your existing enterprise data. Unwired Server manages the secure handling of data connections and transactions. An administration console enables you to manage security, access, authorization, and deployment of mobile applications from a Web browser. Mobile device users download the current data and applications, and upload transactions through online or offline connections.

**See also**

- *Unwired Platform Components* on page 4
- *Component Descriptions* on page 4
- *Unwired Platform Features* on page 6
- *Unwired Platform Components* on page 4
- *Unwired Platform Features* on page 6

## Unwired Platform Components

The Unwired Platform configuration includes several major components.
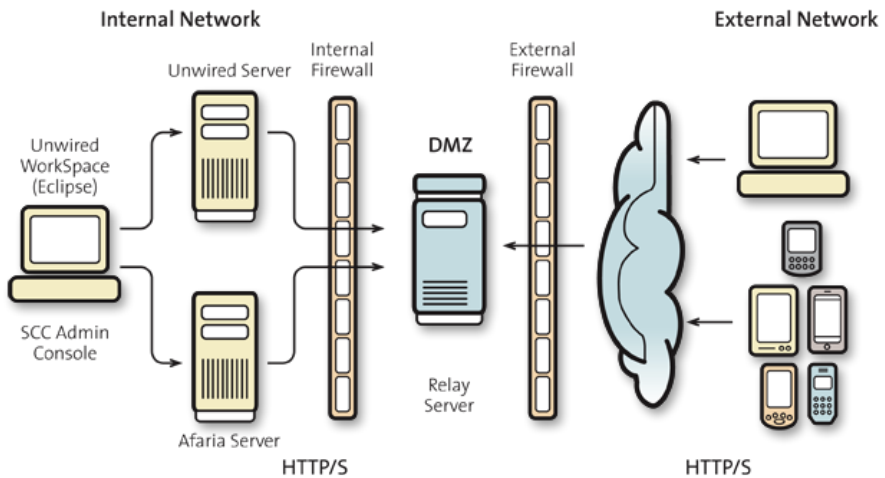


**See also**

- *Solution Architecture for Mobilization* on page 3
- *Component Descriptions* on page 4
- *Unwired Platform Features* on page 6
- *Solution Architecture for Mobilization* on page 3
- *Unwired Platform Features* on page 6

## Component Descriptions

Primary Unwired Platform component descriptions.

| Component | Description |
|---|---|
| Unwired Server | A runtime server that brokers connections and processing between your existing legacy or enterprise information system (EIS) data source and mobile device applications. |
| | Unwired Server enables you to create and retrieve data from your existing data source. Mobile applications, based on mobile business objects (MBOs), represent the data from your legacy or EIS system. Unwired Server manages access to data using your existing security policies, and provides end-to-end security from your data source to the mobile device. |
| | Unwired Server manages data synchronization between mobile devices and the back-end system for online and offline modes. Multiple synchronization strategies and policies are available to ensure the most efficient processing. A consolidated database, also called the cache, is used to manage transactions. |
| | You can deploy multiple Unwired Servers in a server farm, and can manage multiple Unwired Servers in a hosted domain. |
| System Management <br><br> • Sybase Control Center (SCC) console <br> • Command line utilities | Sybase Control Center (SCC) is a Web-based console used to configure and manage Unwired Server and its components. Because it is a Web console, multiple administrators can access different administrable components from a single access point. |
| | Sybase Control Center provides Unwired Server an integrated set of management and configuration capabilities for servers, server logs, package deployment, subscriptions, roles, and users. Once an Unwired Server and its features are configured for your mobile ecosystem, this consolidated management interface allows a bidirectional flow of information from within the data center boundaries to extend critical information to the mobile wireless workforce. |
| | Command line utilities enable you to automate management tasks for certain management functions such as deploying packages. |
| Relay Server | A relay server forwards client requests to a cluster of servers —Unwired Server or Afaria. A relay server is implemented as a pair of Web extensions that run in a Web server. Relay servers run in the DMZ. |

| Component | Description |
|---|---|
| Afaria Server | (Optional) A runtime server used with Unwired Server to perform user, application, and device provisioning in your production environment. |
| Application Development Environment | Unwired WorkSpace – a plug-in to your Eclipse development environment that provides the tools you need to create mobile applications. Experienced Eclipse developers will already be familiar with the consistent use of Windows, Explorers, Views, Wizards, and Preferences; developers new to Eclipse can use Sybase Unwired WorkSpace online help, cheat sheets, samples, and tutorials to quickly become familiar.<br><br>Unwired WorkSpace can be used to extend enterprise data sources (such as Adaptive Server® Enterprise, DB2, SQL Server, and Oracle) and applications (such as SAP® and Remedy) to mobile applications. They can be installed on the same Windows machine, or on different machines. |

**See also**
- *Solution Architecture for Mobilization* on page 3
- *Unwired Platform Components* on page 4
- *Unwired Platform Features* on page 6

## Unwired Platform Features

Unwired Platform features enable you to manage both development and production environments.

- Connect to your data sources, such as databases, packaged applications, and service data sources.
- Create mobile business objects (MBOs) to create a connection to your back-end data source. Define operations, attributes, relationships, and synchronization capabilities as part of the MBO.
- Develop mobile applications from the Sybase Unwired WorkSpace development environment using custom coding and the API.
- Create prototype BlackBerry and Windows Mobile device applications using the Device Application Designer.
- Package and deploy mobile business objects from the development environment to Unwired Server. Unwired Server manages messaging, data services, and synchronization between your data sources and your mobile devices.
- Manage Unwired Server from the Sybase Control Center, a Web browser base interface. From this administrative console, you can manage security, access, and authorization.

**See also**
*   *Solution Architecture for Mobilization* on page 3
*   *Unwired Platform Components* on page 4
*   *Component Descriptions* on page 4
*   *Solution Architecture for Mobilization* on page 3
*   *Unwired Platform Components* on page 4

# Tutorials

Tutorials demonstrate how to use Sybase Unwired Platform tools to mobilize your enterprise.

*   Getting started tutorials provide step-by-step instructions to get started quickly. They are available on Product Documentation.
*   Advanced tutorials demonstrate advanced Sybase Unwired Platform features. They are also available on Product Documentation.

**See also**
*   *About Unwired Platform* on page 2
*   *Samples* on page 7

# Samples

Sample applications are fully developed, working applications that demonstrate the features and capabilities of Sybase Unwired Platform.

Check the Sybase Web site regularly for updates. Navigate to the Sybase Web site, then select *Products > Sybase Unwired Platform > Use tab*.

**See also**
*   *About Unwired Platform* on page 2
*   *Tutorials* on page 7

# Documentation Road Map for Unwired Platform

Learn more about Sybase® Unwired Platform documentation.

**Table 1. Unwired Platform documentation**

| Document | Description |
|---|---|
| *Sybase Unwired Platform Installation Guide* | Describes how to install or upgrade Sybase Unwired Platform. Check the *Sybase Unwired Platform Release Bulletin* for additional information and corrections.<br><br>Audience: IT installation team, training team, system administrators involved in planning, and any user installing the system.<br><br>Use: during the planning and installation phase. |
| *Sybase Unwired Platform Release Bulletin* | Provides information about known issues, and updates. The document is updated periodically.<br><br>Audience: IT installation team, training team, system administrators involved in planning, and any user who needs up-to-date information.<br><br>Use: during the planning and installation phase, and throughout the product life cycle. |
| *New Features* | Describes new or updated features.<br><br>Audience: all users.<br><br>Use: any time to learn what is available. |
| *Fundamentals* | Describes basic mobility concepts and how Sybase Unwired Platform enables you design mobility solutions.<br><br>Audience: all users.<br><br>Use: during the planning and installation phase, or any time for reference. |

| Document | Description |
|---|---|
| *System Administration* | Describes how to plan, configure, manage, and monitor Sybase Unwired Platform. Use with the *Sybase Control Center for Sybase Unwired Platform* online documentation. |
| | Audience: installation team, test team, system administrators responsible for managing and monitoring Sybase Unwired Platform, and for provisioning device clients. |
| | Use: during the installation phase, implementation phase, and for ongoing operation, maintenance, and administration of Sybase Unwired Platform. |
| *Sybase Control Center for Sybase Unwired Platform* | Describes how to use the Sybase Control Center administration console to configure, manage and monitor Sybase Unwired Platform. The online documentation is available when you launch the console (**Start > Sybase > Sybase Control Center**, and select the question mark symbol in the top right quadrant of the screen). |
| | Audience: system administrators responsible for managing and monitoring Sybase Unwired Platform, and system administrators responsible for provisioning device clients. |
| | Use: for ongoing operation, administration, and maintenance of the system. |
| *Troubleshooting* | Provides information for troubleshooting, solving, or reporting problems. |
| | Audience: IT staff responsible for keeping Sybase Unwired Platform running, developers, and system administrators. |
| | Use: during installation and implementation, development and deployment, and ongoing maintenance. |

| Document | Description |
|---|---|
| Getting started tutorials | Tutorials for trying out basic development functionality.<br><br>Audience: new developers, or any interested user.<br><br>Use: after installation.<br><br>• Learn mobile business object (MBO) basics, and create a mobile device application:<br>　• *Tutorial: Mobile Business Object Development*<br>　• *Tutorial: BlackBerry Application Development using Device Application Designer*<br>　• *Tutorial: Windows Mobile Device Application Development using Device Application Designer*<br>• Create native mobile device applications:<br>　• *Tutorial: BlackBerry Application Development using Custom Development*<br>　• *Tutorial: iPhone Application Development using Custom Development*<br>　• *Tutorial: Windows Mobile Application Development using Custom Development*<br>• Create a mobile workflow package:<br>　• *Tutorial: Mobile Workflow Package Development* |
| *Sybase Unwired WorkSpace – Mobile Business Object Development* | Online help for developing MBOs.<br><br>Audience: new and experienced developers.<br><br>Use: after system installation. |
| *Sybase Unwired WorkSpace – Device Application Development* | Online help for developing device applications.<br><br>Audience: new and experienced developers.<br><br>Use: after system installation. |

| Document | Description |
|---|---|
| Developer references for device application customization | Information for client-side custom coding using the Client Object API.<br><br>Audience: experienced developers.<br><br>Use: to custom code client-side applications.<br><br>• *Developer Reference for BlackBerry*<br>• *Developer Reference for iOS*<br>• *Developer Reference for Mobile Workflow Packages*<br>• *Developer Reference for Windows and Windows Mobile* |
| Developer reference for Unwired Server side customization – *Reference: Custom Development for Unwired Server* | Information for custom coding using the Server API.<br><br>Audience: experienced developers.<br><br>Use: to customize and automate server-side implementations for device applications, and administration, such as data handling.<br><br>Dependencies: Use with *Fundamentals* and *Sybase Unwired WorkSpace – Mobile Business Object Development*. |
| Developer reference for system administration customization – *Reference: Administration APIs* | Information for custom coding using administration APIs.<br><br>Audience: experienced developers.<br><br>Use: to customize and automate administration at a coding level.<br><br>Dependencies: Use with *Fundamentals* and *System Administration*. |

Product Task Flow

# Configure - Eclipse Development Environment

Configure your Eclipse Edition development environment by connecting to Unwired Server and your data source, setting preferences, and setting up the tools you need.

## Preferences

Use preferences to configure the appearance of perspectives and views and to customize its tools.

Some preference settings apply across all development components, while others apply only to a specific component. To enable the sharing of editor preferences across components, see the Eclipse *WorkBench User Guide* on the online bookshelf.

### See also
- *Windows Mobile Client Application .DLL Dependencies* on page 44
- *Downloading the BlackBerry JDE and MDS Simulator* on page 47
- *Installing the BlackBerry Java Plug-in for Eclipse* on page 48
- *Configuring Unwired Server to Use HTTPS for Replication Sync* on page 49
- *Migrating Device Applications* on page 52

### Setting Help Display Preferences

Define how you want to display help topics from an Eclipse-based product to appear.

1. Select **Window | Preferences** from the main menu bar.
2. In the left pane, select **Help**.

   The Help options appear in the right pane.
3. Specify how to display help topics.
4. Click **OK**.

### Setting Mobile Development Preferences

Use the Preferences dialog to set Mobile Application Diagram and mobile business object preferences.

1. Either:
   - Open the Preferences dialog from the menu, by selecting **Window > Preferences**. This is the recommended method, since it shows all preference options. Or

- Open the Preferences dialog from the Mobile Application Diagram, by right-clicking in the Mobile Application Diagram and selecting **Preferences**.

2. In the left pane of the Preferences dialog, expand **Sybase, Inc > Mobile Development**.

3. In the Preferences dialog, configure preferences for:
   - Developer Profile
   - Device Application Designer
   - Logging
   - Miscellaneous
   - Mobile Application Diagram
   - Mobile Business Object
   - Mobile Workflow Forms Editor

4. Change preferences for the desired category, then click **Apply** to apply changes, or click **Restore Default** to use the default preference settings.

5. Click **OK**.

## Setting Device Application Designer Preferences

Use the Preferences dialog to set the appearance and device preferences for the Device Application Designer.

1. Right-click on any page of the Device Application Designer (except the Introduction page), and select **Preferences**.

2. In the Preferences dialog, configure the preferences for:
   - Appearance
   - Devices
   - Errors/Warnings
   - Font Mappings

3. Click **Apply**.

4. Click **OK**.

### Device Application Designer Preferences

Use the Preferences page to set Device Application Designer preferences for automatic screen creation, question dialogs, and font mappings.

*Automatic creation*

Use the automatic creation preferences to specify which screens to create automatically when you drag and drop a mobile business object onto the Flow Design page.

| Preference | Description |
|---|---|
| Show icons on the list item table of contents | If selected an image in the list item is created. |
| | Whether this option is selected or not, if the source screen of a non-alert type of connection is a Table of Contents screen, the connections will always be created. Menu items will, however, not be created. Instead, a list item with a connection action that points to the target screen is created. |
| Add a navigation bar to new screens | When a new screen is added, a navigation bar will also be added. |
| | Note the following: |
| | • This does not automatically add a navigation bar to a tab panel. |
| | • When using a user-created template, the navigation bar is not automatically added since the template is user-defined. |
| | • If you start editing a Device Application Designer document from the Screen Design, a new screen is added automatically, and a navigation bar will not be added. |
| | • If the navigation bar is in the first screen of application, the **Back** button is not available. |
| Add a toolbar when adding a grid or cell table | If selected, when adding either a grid or cell table to the Screen Design, an empty toolbar will also be added. |
| | **Note:** An empty toolbar causes a problem check if nothing is ever added. |
| Add a toolbar when adding a list details | If selected, when adding a list details control to the Screen Design, an empty toolbar will also be added. |
| | **Note:** An empty toolbar will cause a problem check if nothing is ever added. When you start adding operations like create, update, or delete, the problem check will add them into the toolbar if it exists. |
| **New Connection** | |

| Preference | Description |
|---|---|
| Create a menu item with a connection action for new connections | Create a menu item in the source screen for a new connection with a name that matches the target screen name. If a menu item with the target screen name already exists, a common name is used instead. |
| | **Note:** This does not apply to stock screens, screens with grid or cell tables, or a table of contents screen as the source screen. |
| Create a toolbar item with a connection action for new connections | Create a toolbar item on the Screen Design with an action assigned for the new connection, with a name that matches the target screen. If a toolbar item with the target screen name already exists, a common name is used instead. The toolbar item is created only if a toolbar exists on the screen design. The toolbar icon is the target screen icon. |
| | **Note:** This does not apply to stock screens, table of contents screens, tab-based screens, or screens with grid or cell tables. |
| **Screen Creation** – specify the screens that are created automatically when you add a mobile business object to the Flow Design. | |
| Operations | • Operations Screens<br>  • Pending Operations Screen – create a screen that displays mobile business object operations in tabular form.<br>  • Delete Operations Screen – create a screen that displays delete operations of mobile business objects in tabular form.<br>  • Logs Screen – create a screen that displays synchronization activity. |
| Relationships | • Relationship Screens – create all relationship screens for a mobile business object table screen.<br>  • Relationship Automatic Screens – create all relationship automatic screens for the relationship screen. This recursively creates screens for each relationship screen. |

| Preference | Description |
|---|---|
| General | • Search Screen – create a search screen for the mobile business object table screen.<br>• List Details Screen – create a list details screen for the mobile business object table screen.<br>• Synchronization Screen – create a synchronization screen. A synchronization screen lists all of the mobile business objects used and allows you to selectively synchronize synchronization groups. |

*Do not ask these questions*

Select the dialogs you do not want to display while designing the device application. For example, when deleting multiple screens or connections from the Flow Design, the confirmation dialog appears for each screen or connection you delete. If you do not want the dialog to display multiple times, select the options in the Flow Design section.

| Preference | Description |
|---|---|
| Flow Design | Select the dialogs you do not want to display while designing in the Flow Design. |
| Screen Design | Select the dialogs you do not want to display while designing in the Screen Design. |
| General | Select the dialogs you do not want to display. |

Unselect the boxes next to the dialogs you want to appear. You can also select these preferences during design time from the dialog box itself when you select **Do Not Show This Dialog Again** and click **Yes**.

*Automatic Screen Creation*

Automatically create screens that are populated with data from the data source by dragging and dropping a mobile business object onto the Flow Design.

*Setting the preferences for automatic screen creation*

Set the preferences for automatic screen creation in the Device Application Designer Preferences. These preferences allow you to decide which screens to create when you drop a mobile business object onto the Flow Design canvas.

The Relationship Automatic Screens option means that all sub screens are created after the initial relationship screen is created.

By default, Delete Operation Screen is unselected, which means the Device Application Designer:

- Creates the main screen with a table for the mobile business object.
- Adds a Delete menu item to the main screen.
- Adds an alert action of the question type, which goes back to the main screen to the menu item. Adding this action means the user is prompted to confirm before deleting.
- Adds a save context action to the menu item.
- Adds an operation action that performs the delete operation to the menu item.

If Delete Operation Screen is selected, a Delete screen is created.

*Behavior of automatic screen creation*
When using the auto creation feature, keep these points in mind:

If you drop more than one mobile business object onto the Flow Design canvas, only the main screens are created. If a screen has multiple related screens that can be created, use the assist

option  at the top of the screen to create the screens.

When using MBOs with multiple relationships, if different behavior is expected for the different types of relationships, you should manually create different screens to accommodate the different types of relationships instead of using the same screen for all relationships.

*Editing Device Simulator Settings*
Use the Device Preferences to edit the simulator settings for devices.

1. Right-click on any page of the Device Application Designer (except Introduction), and select **Preferences**.
2. Expand **Devices** and select the device for which to edit the settings.
3. From the Device Configurations list, select the device for which to edit the settings and click **Edit**.
4. In the Edit Device dialog, the Simulator Location shows the default location for the device. To change the location, click **Browse**.
5. The Simulator File shows the batch file to run for the selected simulator. This batch file is used by the generation wizard. Click **Browse** to open the file dialog to update both the simulator location and simulator file text boxes. Updating the batch file updates the path automatically.
6. Click **OK**.

**Device Application Designer Appearance Preferences**

Set the appearance preferences for the Device Application Designer pages and palette groupings.

*Device Application Designer page preferences*

Set the preferences for the appearance of the Device Application Designer pages and Palette groupings.

| Section | Description |
|---|---|
| Device Application Designer pages | Select the pages you want to view in the Device Application Designer editor during the development process.<br><br>• Show the Introduction page – by default this is selected. Unselect to hide the Introduction page.<br>• Show the Source page – select to show the Source page in the editor. |
| Flow Design page | Use these options to configure the Grid, Geometry options, and so on in the Flow Design page:<br><br>• Grid – select to display the grid in the Flow Design page.<br>• Snap to geometry – select to show the lines of geometry for aligning objects in the Flow Design page.<br>• Show the connection type – unselect to remove the name from the connection link line; you may want to do this to clear clutter from flows with a lot of connections. The connections remain color-coded, and the tooltip on the line describes the connection type.<br>• Route connection directly – this style is used by default and shows the connection line directly. If you unselect this option, a "Manhattan" style of routing is used, which has limitations if there are many connections in the flow design. |

| Section | Description |
|---------|-------------|
| Palette grouping | If selected, the palette group appears as a drop-down. Use this option to specify how objects in the Flow Design and Screen Design Palettes appear: <br><br> • Flow Design <br>     • Screens <br>     • Stock Screens <br>     • Settings Screens <br>     • Alerts <br>     • Connections <br> • Screen Design <br>     • Controls <br>     • Containers <br>     • Cells <br>     • Menu <br>     • Actions |

*Appearance Preferences for Controls*
Set the preferences for the appearance of the Device Application Designer controls.

| Control | Preferences |
|---------|-------------|
| Label | • Wrap to multiple lines – select to truncate and wrap text that is too long to fit on one line to multiple lines.<br>• Style – select the label style from the drop-down menu.<br>• Span<br>  • Horizontal – select the number of columns the widget will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display. Select **All** to set the horizontal span across the display.<br>  • Vertical – enter the number of rows the widget will occupy.<br>• Alignment<br>  • Horizontal<br>    • Inheritance – the control inherits the properties that are set for the container in which it is contained. If it is not in a container, it inherits the properties set in the Display.<br>    • Left – left-justified alignment.<br>    • Center – centered alignment.<br>    • Right – right-justified alignment.<br>  • Vertical<br>    • Top – positions the control at the top of the span.<br>    • Center– positions the control in the center of the span.<br>    • Bottom – positions the control at the bottom of the span. |

| Control | Preferences |
|---------|-------------|
| Edit Box | • Show border – select to show the edit box border on the device.<br>• Empty string means null – when this option is selected a null value is passed to the server if the user leaves the input field empty. Some backend systems require this behavior.<br><br>**Note:** This option is enabled only for the STRING data type.<br><br>• Style – select the edit box style.<br>• Span<br>  • Horizontal – select the number of columns the edit box will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display. Select **All** to set the horizontal span across the display.<br>  • Vertical – enter the number of rows the edit box will occupy. |
| Choice | • Empty string means null – when this option is selected a null value is passed to the server if the user leaves the input field empty. Some backend systems require this behavior.<br><br>**Note:** This option is enabled only for the STRING data type.<br><br>• Style – select the style for the choice widget.<br>• Span<br>  • Horizontal – select the number of columns the choice widget will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display. Select **All** to set the horizontal span across the display.<br>  • Vertical – enter the number of rows the choice widget will occupy. |

| Control | Preferences |
|---------|-------------|
| Hyperlink | • Text equals value – select so that, by default, the text entered in the Hyperlink text edit box is treated as the hyperlink value.<br>• Focus style – select the style for the hyperlink when it is the object of focus.<br>• Unfocus style – select the style for the hyperlink when it is not the object of focus.<br>• Span<br>  • Horizontal – select the number of columns the hyperlink will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display. Select **All** to set the horizontal span across the display.<br>  • Vertical – enter the number of rows the hyperlink will occupy.<br>• Alignment<br>  Horizontal<br>  • Inheritance – the control inherits whatever properties are set for the container in which it is contained. If it is not in a container, it inherits the properties set in the Display.<br>  • Left – left-justified alignment.<br>  • Center – centered alignment.<br>  • Right – right-justified alignment.<br>  Vertical<br>  • Top – positions the control at the top of the span.<br>  • Center– positions the control in the center of the span.<br>  • Bottom – positions the control at the bottom of the span. |

| Control | Preferences |
|---|---|
| Checkbox/Switch | <ul><li>Style – select the style for the checkbox or switch.</li><li>Span<ul><li>Horizontal – select the number of columns the checkbox or radio button will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display. Select **All** to set the horizontal span across the display.</li><li>Vertical – enter the number of rows the checkbox or radio button will occupy.</li></ul></li><li>Alignment<br>Horizontal<ul><li>Inheritance – the control inherits the properties that are set for the container in which it is contained. If it is not in a container, it inherits the properties set in the Display.</li><li>Left – left-justified alignment.</li><li>Center – centered alignment.</li><li>Right – right-justified alignment.</li></ul>Vertical<ul><li>Top – positions the control at the top of the span.</li><li>Center – positions the control in the center of the span.</li><li>Bottom – positions the control at the bottom of the span.</li></ul></li></ul> |

| Control | Preferences |
|---|---|
| Radio Button/Segment | • Text equals value – select so that, by default, the text entered in the Radio Button/Segment text edit box is treated as the Radio Button/Segment value.<br>• Span<br>  • Horizontal – select the number of columns the checkbox or radio button will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display. Select **All** to set the horizontal span across the display.<br>  • Vertical – enter the number of rows the checkbox or radio button will occupy.<br>• Alignment<br>Horizontal<br>  • Inheritance – the control inherits the properties that are set for the container in which it is contained. If it is not in a container, it inherits the properties set in the Display.<br>  • Left – left-justified alignment.<br>  • Center – centered alignment.<br>  • Right – right-justified alignment.<br>Vertical<br>  • Top – positions the control at the top of the span.<br>  • Center – positions the control in the center of the span.<br>  • Bottom – positions the control at the bottom of the span. |
| List Item | • Focus style – select the style for the list item when it is the object in focus.<br>• Unfocus style – select the style for the list item when it is not the object of focus. |

| Control | Preferences |
|---------|-------------|
| Button | • Focus style – select the style for the button when it is in focus.<br>• Unfocus style – select the style for the button when it is not in focus.<br>• Span<br>    • Horizontal – select the number of columns the button will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display. Select **All** to set the horizontal span across the display.<br>    • Vertical – enter the number of rows the button will occupy.<br>• Alignment<br>Horizontal<br>    • Inheritance – the control inherits whatever properties are set for the container in which it is contained. If it is not in a container, it inherits the properties set in the Display.<br>    • Left – left-justified alignment.<br>    • Center – centered alignment.<br>    • Right – right-justified alignment.<br>Vertical<br>    • Top – positions the control at the top of the span.<br>    • Center– positions the control in the center of the span.<br>    • Bottom – positions the control at the bottom of the span. |

| Control | Preferences |
|---------|-------------|
| Image<br><br>Spacer | • Show spacing occupied – select if you want the image or spacer to completely fill the region of the display it occupies.<br>• Span<br>  • Horizontal – select the number of columns the image or spacer will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display. Select **All** to set the horizontal span across the display.<br>  • Vertical – enter the number or rows the image or spacer will occupy. |
| Separator | • Separator thickness – select the thickness of the separator line, in pixels.<br>• Style – select the style for the separator.<br>• Horizontal span – select the number of columns the separator will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display. Select **All** to set the horizontal span across the display. |

| Control | Preferences |
|---------|-------------|
| Grid Table | Select styles for: <br><br> • Header style <br> • Odd row style <br> • Even row style <br> • Pending style <br> • Border style <br> • Focus style <br> • Support sorting – select for the ability to sort on columns in the table and enter a name for the sort menu label. <br> • Support paging – select to enable paging on the record data in the column. <br>    • Start paging record number – enter the number of records the table should contain before it starts paging. The minimum number for paging records is 25, the maximum is 10000. <br>    • Paging record size – enter the number of records to store in the table page. The minimum number for paging records is 25, the maximum is 10000. <br>    • Number of pages to cache – enter the number of table pages to cache in memory. The minimum number for paging records is 2, the maximum is 10. |

| Control | Preferences |
|---------|-------------|
| Cell Table | • Focus style – select the style for the cell when it is in focus.<br>• Unfocus style – select the style for the cell when it is not in focus.<br>• Number of columns – select the number of columns the cell table will have.<br>• Assign percentages – the width of each column in the cell table, in percentages. The defaults are 33% for the first column, 34% for the second column, and 33% for the third column.<br>• Support sorting – select for the ability to sort on columns in the table and enter a name for the sort menu label.<br>• Support paging – select to enable paging on the record data in the column.<br>   • Start paging record number – enter the number of records the table should contain before it starts paging. The minimum number for paging records is 25, the maximum is 10000.<br>   • Paging record size – enter the number of records to store in the table page. The minimum number for paging records is 25, the maximum is 10000.<br>   • Number of pages to cache – enter the number of table pages to cache in memory. The minimum number for paging records is 2, the maximum is 10. |

| Control | Preferences |
|---------|-------------|
| Cell Label | • Wrap to multiple lines – select to truncate and wrap text that is too long to fit on one line to multiple lines. <br> • Focus style – select the style for the cell label when it is in focus. <br> • Unfocus style – select the style for the cell label when it is not in focus. <br> • Span <br>  • Horizontal – select the number of columns the cell label will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display. Select **All** to set the horizontal span across the display. <br>  • Vertical – enter the number of rows the cell label will occupy. <br> • Alignment <br>  • Horizontal: <br>   • Inheritance – the control inherits whatever properties are set for the container in which it is contained. If it is not in a container, it inherits the properties set in the Display. <br>   • Left – left-justified alignment. <br>   • Center – centered alignment. <br>   • Right – right-justified alignment. <br>  • Vertical: <br>   • Top – positions the control at the top of the span. <br>   • Center– positions the control in the center of the span. <br>   • Bottom – positions the control at the bottom of the span. |

| Control | Preferences |
|---------|-------------|
| Cell Image | • Scale proportionally – large images will be scaled to fit. Selecting this option preserves the width and height ratio.<br>• Span<br>  • Horizontal – select the number of columns the cell image will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display. Select **All** to set the horizontal span across the display.<br>  • Vertical – enter the number of rows the cell image will occupy.<br>• Alignment<br>  • Horizontal:<br>    • Inheritance – the control inherits whatever properties are set for the container in which it is contained. If it is not in a container, it inherits the properties set in the Display.<br>    • Left – left-justified alignment.<br>    • Center – centered alignment.<br>    • Right – right-justified alignment.<br>  • Vertical:<br>    • Top – positions the control at the top of the span.<br>    • Center– positions the control in the center of the span.<br>    • Bottom – positions the control at the bottom of the span. |

| Control | Preferences |
|---------|-------------|
| List Detail | Select the styles for:<br><br>• Odd row style<br>• Even row style<br>• Pending style<br>• Border style<br>• Focus style<br>• Assign percentages – the width of each column in the list detail, in percentages. The defaults are 50% for the first column and 50% for the second column<br>• Horizontal alignment – choose the horizontal alignment for the columns:<br>   • Inheritance – the control inherits whatever properties are set for the container in which it is contained. If it is not in a container, it inherits the properties set in the Display.<br>   • Left – left-justified alignment.<br>   • Center – centered alignment.<br>   • Right – right-justified alignment.<br>• Use hot keys – select to allow users to customize hot keys for Previous and Next operations. You can select hot keys from characters a through z. The defaults are p and n, for Previous and Next, respectively.<br>• Show previous and next options in menu – display Previous and Next operations in the device menu. By default this option is selected. These menu options display dynamically at runtime on the device.<br>• Previous menu label – the label of the Previous menu. The menu label appears only when the Previous menu is shown dynamically at runtime on the device.<br>• Next menu label – the label of the Next menu. The menu label appears only when the Next menu is shown dynamically at runtime on the device |

| Control | Preferences |
|---|---|
| Navigation Bar | • Show back button – select if you want the back button to display on the navigation bar. If the screen is the first screen, this button is not available to device users.<br>• Style – select the navigation bar style.<br>• Border style – select the navigation bar border style. |

*Appearance Preferences for Containers*

Set the preferences for the appearance of the Device Application Designer containers.

| Container | Preferences |
|---|---|
| Display | • Number of columns – the number of columns in the display.<br>• Assign percentages – the width of each column in the display.<br>• Horizontal alignment – select the horizontal alignment of the display:<br>   • Left – left-justified alignment.<br>   • Center – centered alignment.<br>   • Right – right-justified alignment.<br>• Show pattern – adds a pattern to the background color.<br>• Style – select the style for the display. |

| Container | Preferences |
|---|---|
| Region | • Number of columns – the number of columns in a region. You can have 1 – 3 columns. The default is 2 columns.<br>• Assign percentages – the width of each column in the region, in percentages. The defaults are:<br>  • 1 column – 100%.<br>  • 2 columns – 35% for the first column, 65% for the second column.<br>  • 3 columns – 33% for the first column, 34% for the second column, and 33% for the third column.<br>• Show border – the foreground color of the style is used for the border of the region.<br>• Style – select the style to apply to the region.<br>• Span<br>  • Horizontal – select the number of columns the region will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display. Select **All** to set the horizontal span across the display.<br>  • Vertical – enter the number of rows the region will occupy. |

| Container | Preferences |
|---|---|
| Radio Group\Segment Group | <ul><li>Number of columns – the number of columns in a radio group. You can have 1 – 3 columns. The default is 1 column.</li><li>Assign percentages – the width of each column in the region or radio group, in percentages. The defaults are:<ul><li>1 column – 100%.</li><li>2 columns – 35% for the first column, 65% for the second column.</li><li>3 columns – 33% for the first column, 34% for the second column, and 33% for the third column.</li></ul></li><li>Read only – makes the radio buttons in the group read only. This option applies to all the radio buttons in the group, so either all buttons are read only or none are read only.</li><li>Empty button value means NULL – if selected, an empty string value on the selected control means the value shoule be NULL when assigned to an action.</li><li>Selected control style – select the style for the selected radio button.</li><li>Unselected control style – select the style for the unselected radio button.</li><li>Span<ul><li>Horizontal – select the number of columns the radio group will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display. Select **All** to set the horizontal span across the display.</li><li>Vertical – enter the number of rows the radio group will occupy.</li></ul></li></ul> |

| Container | Preferences |
|---|---|
| List Group | Span<br><br>• Horizontal – enter the number of columns the list group will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display.<br>• Vertical – enter the number of rows the list group will occupy. |
| Tab Folder | • Active tab style – select the style for the active tab.<br>• Inactive tab style – select the style for the inactive tab.<br>• Tab border style – select the style for the tab border. |
| Tab panel | • Number of columns – select the number of columns the tab panel will have.<br>• Assign percentages – the width of each tab panel, in percentages. The defaults are:<br> • 1 column – 100%.<br> • 2 columns – 35% for the first column, 65% for the second column.<br> • 3 columns – 33% for the first column, 34% for the second column, and 33% for the third column.<br>• Horizontal alignment – select the horizontal alignment for the items in the tab panel:<br> • Left – left-justified alignment.<br> • Center – centered alignment.<br> • Right – right-justified alignment.<br>• Show pattern – adds a pattern to the background color.<br>• Style – select the style to apply to the tab panel. |

| Container | Preferences |
|---|---|
| Toolbar | • Style – select the style to apply to the toolbar.<br>• Border style – select the border style to apply to the toolbar.<br>• Horizontal alignment – select the horizontal alignment for the items in the toolbar:<br>   • Left – left-justified alignment.<br>   • Center – centered alignment.<br>   • Right – right-justified alignment. |

### Device Application Designer Device Preferences

Edit the device simulator settings.

| Preference | Description |
|---|---|
| Default Configurations | Default Platform Device – select the default device to use when creating a new device application. |
| Advanced Configuration | By default this is selected so that system fonts and sizes are mapped to the default device font and size. If this is not selected, no device mapping is performed for system fonts and sizes. |

*BlackBerry*

| Preference | Description |
|---|---|
| Generate custom subclasses for all | When checked, all tabs, user interface definitions, and options definitions have a subclass generated. |
| Device Configurations for External BlackBerry JDE Installation | The tree view displays and allows you to edit device simulator locations and simulator batch files. Click **Edit** to change the path to the simulator and its batch files. |
| BlackBerry MDS port | The BlackBerry MDS port is used to detect if MDS server is running so that the "Start MDS automatically" option on BlackBerry application generation wizard works. |

*Windows Mobile*

The device preferences for Windows Mobile contain two sections:

- Device – create new or edit existing device configurations. You cannot edit device configurations that are pre-defined by the system.
- Platform – create new or edit existing device platform configurations. You cannot edit device platform configurations that are pre-defined by the system.

### Editing the BlackBerry Simulator Location
Edit the location of the BlackBerry Simulator and simulator batch files.

1. In Device Application Designer **Devices** preferences, select **BlackBerry**.
2. Select the device for which to edit the location, and click **Browse** to find the Simulator Location and Simulator File locations.
3. Click **OK**.

### Creating or Editing Windows Mobile Device Properties
Create new device properties or edit existing user-defined device properties for Windows Mobile devices.

You cannot edit device properties that are pre-defined by the system.

1. In Device Application Designer Devices preferences, select **Windows Mobile > Device**.
2. Click **Create** or select a device from the list and click **Edit** and enter or select the properties for the device.
3. Click **OK**.

### Creating or Editing Windows Mobile Device Platform Properties
Create a new device platform properties configuration or edit existing user-defined platform properties for Windows Mobile devices.

You cannot edit device platform properties that are pre-defined by the system.

1. In Device Application Designer Devices preferences, select **Windows Mobile > Platform**.
2. Click **Create** or select a device platform from the list and click **Edit** and enter or select the properties for:
   - Platform family name
   - Platform native name
   - Platform ID
   - Platform OS version
3. Click **OK**.

**Device Application Designer Error and Warning Preferences**

The Errors and Warnings preferences allow you to set the problem level for errors reported in the Problems view when you verify the device application.

You can edit the severity of errors and warnings on the Flow Design, Screen Design Controls, Screen Design Actions/Validation, General [Device Application Designer], BlackBerry, and Windows Mobile to these levels:

- Error – an error appears in the Problems view. Errors prevent the generation of the device application.
- Warning – a warning is not as critical as an error, and the problem that caused the warning appears in the Problems view.
- Ignore – when set to "Ignore," this condition does not appear in the Problems view.

*Editing Error and Warning Preferences*

Use the Preferences dialog to edit the severity of the warnings and errors that appear in the Problems view.

1. Right-click on any page of the Device Application Designer (except the Introduction page), and select **Preferences**.
2. In the Preferences dialog, navigate to **Sybase, Inc > Mobile Development > Device Application Designer > Errors/Warnings**.
3. From the Problem list, select the error or warning to edit, then choose the severity level:
   - Error – an error appears in the Problems view. Errors prevent the generation of the device application.
   - Warning – a warning is not as critical as an error, and the problem that caused the warning appears in the Problems view.
   - Ignore – when severity level is set to "ignore," the applicable condition does not appear in the Problems view.
4. Click **Apply**.
5. Click **OK**.

**Device Application Designer Font Mappings Preferences**

Set the preferences for mapping the system font and size to the device font and size.

Use the Font Mappings page to create, edit, and remove system fonts and sizes.

| Section | Description |
|---|---|
| Default Configurations | Default Sample Text – used in the Style References dialog preview. |

| Section | Description |
|---------|-------------|
| Font Configurations | This section displays all the fonts. Expand the tree to show the settings for each font. Click **Expand All** to display all the fonts and their settings. |

*Configuring Font Mappings*

Use the Font Mappings preferences dialog to map fonts that are selected for the device to an appropriate Windows font for display purposes.

1. Right-click on any page of the Device Application Designer (except the Introduction page), and select **Preferences**.
2. Select **Sybase, Inc. > Mobile Development > Device Application Designer > Font Mappings**.
3. In the Font Mappings dialog, select from:

| Option | Description |
|--------|-------------|
| **New** | Create a new font mapping. |
| **Edit** | Edit the selected font mapping. |
| **Remove** | Remove the selected font. You cannot remove pre-defined fonts. |
| **Expand All** | Expand the font tree and see all the attributes of the font. |
| **Collapse All** | Collapse the font tree. |

4. Click **Apply**.
5. Click **OK**.

*Creating a New Font Mapping*

Create new font mappings using the Font Mappings preferences.

1. Right-click in any page of the Device Application Designer (except the Introduction page), and select **Preferences**.
2. Select **Sybase, Inc > Mobile Development > Device Application Designer > Font Mappings**.
3. In the Font Mappings dialog, select a font and click **New**.
4. In the Create Font Mapping dialog:

| Option | Description |
|--------|-------------|
| Font Name | The name of the font found on the device. |
| Mapping Font | Select the font for the current system from the drop-down list. |

| Option | Description |
|---|---|
| Sample Text | Displays a preview of the font. |
| Use Default Sample Text | Select to use the default sample text specified in Font Mappings preference page. |
| Font Sizes | Shows the supported font sizes. |

**5.** Click **OK**.

*Editing Font Mappings*

Use the Edit Font Mapping dialog to edit the name and size of fonts.

**1.** Right-click on any page of the Device Application Designer (except the Introduction page), and select **Preferences**.

**2.** Select **Sybase, Inc > Mobile Development > Device Application Designer > Font Mappings**.

**3.** In the Font Mappings dialog, select a font and click **Edit**.

**Note:** SYSTEM fonts cannot be edited.

**4.** In the Edit Font Mapping dialog, enter your changes:

| Option | Description |
|---|---|
| **Font Name** | You cannot change the font name of he name of predefined system fonts. |
| **Mapping Font** | Select the font to edit from the drop-down list. |
| **Sample Text** | Displays a preview of the font. |
| **Use Default Sample Text** | Select to use the default sample text. |
| **Font Sizes** | Shows the supported font sizes. |

**5.** Click **OK**.

*Creating a New Font Size*

Use the Create Font Mapping dialog to create new font sizes, or edit existing font sizes.

**1.** Right-click in any page of the Device Application Designer (except the Introduction page), and select **Preferences**.

**2.** Select **Sybase Inc > Mobile Development > Device Application Designer > Font Mappings**.

**3.** In the Font Mappings dialog, click **New**.

**4.** In the Create Font Mapping dialog, click **New**.

**5.** In the Create Font Size dialog:

| Option | Description |
|---|---|
| **Size** | Enter the font size. |
| **Set as Default Font Size** | Select to use this as the default font size. |

**6.** Click **OK**.

*Editing Font Size*
Use the Edit Font Mapping dialog to edit the font size.

**1.** Right-click in any page of the Device Application Designer (except the Introduction page), and select **Preferences**.
**2.** Select **Sybase Inc > Mobile Development > Device Application Designer > Font Mappings**.
**3.** In the Font Mappings dialog, select the font for which to edit the size and click **Edit**.
**4.** In the Edit Font Mapping dialog, from Font Sizes, select the font for which to edit the size and click **Edit**.
**5.** In the Edit Font Size dialog, in Size, enter the size of the font and click **OK**.
**6.** In the Edit Font Mapping dialog, click **OK**.

*Removing a Font Size*
Use the Edit Font Mapping dialog to remove font sizes.

**1.** Right-click in any page of the Device Application Designer (except the Introduction page), and select **Preferences**.
**2.** Select **Sybase Inc > Mobile Development > Device Application Designer > Font Mappings**.
**3.** In the Font Mappings dialog, select the font for which to remove the size and click **Edit**.
**4.** In the Edit Font Mapping dialog, from Font Sizes, select the font for which to remove the size and click **Remove**.
**5.** In the confirmation dialog, click **Yes**.

*Setting the Default Size for a Font*
Use the Edit Font Mapping dialog to set the default size for the selected font.

**1.** Right-click on any page of the Device Application Designer (except the Introduction page), and select **Preferences**.
**2.** Select **Sybase Inc > Mobile Development > Device Application Designer > Font Mappings**.

3.  In the Font Mappings dialog, select the font for which to set the default size and click **Edit**.
4.  In the Edit Font Mapping dialog, from Font Sizes, select the font for which to set the default size and click **Set Default**.
5.  Click **OK**.

## Setting Mobile Workflow Forms Editor Preferences

Use the Preferences dialog to set the appearance and device preferences for the Mobile Workflow Forms Editor.

1.  Select **Window > Preferences**.
2.  In the Preferences dialog, select **Sybase, Inc. > Mobile Development > Mobile Workflow Forms Editor**.
3.  Select **Flow Design Page** to set the preferences for the Flow Design page.
4.  Select **Screen Design Page** to set the preferences for the Screen Design page.
5.  Click **Apply**.
6.  Click **OK**.

### Mobile Workflow Forms Editor Preferences

Use the Preferences dialog to set the preferences for the Flow Design and Screen Design pages.

Preference settings determine the default behavior and style for the Flow Design and Screen Design pages and their objects. You can change the style and behavior of objects individually in the Properties view for a selected object. You can set these preferences for the Flow Design and Screen Design pages:

| Preference | Description |
| --- | --- |
| Appearance | Set the color and font preferences for the designer. |

| Preference | Description |
|---|---|
| Connections | When a new connection is created, it follows the default style defined in the preferences settings. You can change the style of the connection line individually in the Flow Design Properties view.<br><br>Select the default line style for new connections:<br><br>• Oblique (default) – the connection lines are not aligned with the horizontal or vertical axis and you can create break points anywhere on the connection line. If you drag the connection line, a new breakpoint is created at the drag point and two lines are created.<br>• Rectilinear – the connection line always remains either horizontal or vertical. |
| Printing | Set the general printing preferences. |
| Rulers and Grid | Set the preferences for the ruler and grids:<br><br>• Ruler options:<br>  • Show rulers for new diagram<br>  • Ruler units<br>• Grid options:<br>  • Show grid for new diagrams<br>  • Snap to grid for new diagrams – select if you want objects in the diagram to stick to grid lines when they are being moved.<br>  • Snap to shapes for new diagrams<br>  • Grid spacing (in inches) |

## Windows Mobile Client Application .DLL Dependencies

To build device clients on Windows Mobile, some DLL files, which are provided in the Unwired Platform installation, are required in certain situations, such as when using a secure port for synchronization.

If required, copy the following .dll files to the location used for referencing them in the Visual Studio application source project.

**Note:** Some of these DLLs are located in multiple locations in the Sybase Unwired Platform installation directories. This table shows only one location.

| Platform | Location | Files | Notes |
|---|---|---|---|
| Windows Mobile Professional 6.0, 6.1, and 6.5 | `%Sybase%\Un-wiredPlatform\Servers\Un-wiredServer\ClientAPI\SQLite\debug` | • `System.Da-ta.SQLite.dll`<br>• `SQLite.In-terop.dll` | Required for messaging-based applications. |
| | `%Sybase%\Un-wiredPlatform\Servers\Un-wiredServer\ClientAPI\dotnet\ce\<.Net Com-pact Frame-work version>\` | `sup-cli-ent.dll` | |
| | • `%Sybase%\Unwired-Platform\Servers\Unwired-Server\Cli-entAPI\Ul-tralite\Win32`<br>• `%Sybase%\Unwired-Platform\Servers\Unwired-Server\Cli-entAPI\Ul-tralite\CE\Arm.50` | • `ulnet11.dll`<br>• `mlcrsa11.dll` | The `mlcrsa11.dll` file is needed only if you are using a secure port (HTTPS) for synchronization. |

| Platform | Location | Files | Notes |
|---|---|---|---|
| Windows XP, Vista, Windows 7 | `%Sybase%\Un-` `wiredPlatform` `\Servers\Un-` `wiredServer` `\ClientAPI` `\Ultralite\CE` `\Assembly\V2` | `iAnywhere.Da-` `ta.Ultra-` `Lite.dll` | |
| | `%Sybase%\Un-` `wiredPlatform` `\Servers\Un-` `wiredServer` `\ClientAPI` `\Ultralite\CE` `\Assembly` `\V2\<Lang>` | `iAnywhere.Da-` `ta.Ultra-` `Lite.resour-` `ces.dll` | Copy from the respective locale-specific folders. |
| | `%Sybase%\Un-` `wiredPlatform` `\Servers\Un-` `wiredServer` `\ClientAPI` `\dotnet` `\win32\` | `sup-cli-` `ent.dll` | |
| | `%Sybase%\Un-` `wiredPlatform` `\Servers\Un-` `wiredServer` `\ClientAPI` `\Ultralite` `\win32` | • `ulnet11.dll` • `mlcrsa11.dl` `l` | The `mlcrsa11.dll` file is required only if you are using a secure port (HTTPS) for synchronization. |

| Platform | Location | Files | Notes |
|---|---|---|---|
| | `%Sybase%\Un-`<br>`wiredPlatform`<br>`\Servers\Un-`<br>`wiredServer`<br>`\ClientAPI`<br>`\Ultralite`<br>`\win32\Assem-`<br>`bly\V2` | `iAnywhere.Da-`<br>`ta.Ultra-`<br>`Lite.dll` | |
| | `%Sybase%\Un-`<br>`wiredPlatform`<br>`\Servers\Un-`<br>`wiredServer`<br>`\ClientAPI`<br>`\Ultralite`<br>`\win32\Assem-`<br>`bly\V2\<lan-`<br>`guage>` | `iAnywhere.Da-`<br>`ta.Ultra-`<br>`Lite.resour-`<br>`ces.dll` | Copy from the respec-tive locale-specific folders. |

**See also**
- *Preferences* on page 13
- *Downloading the BlackBerry JDE and MDS Simulator* on page 47
- *Installing the BlackBerry Java Plug-in for Eclipse* on page 48
- *Configuring Unwired Server to Use HTTPS for Replication Sync* on page 49
- *Migrating Device Applications* on page 52

# Downloading the BlackBerry JDE and MDS Simulator

To generate and distribute BlackBerry device applications built with the Unwired WorkSpace Device Application Designer, download the MDS simulator and the BlackBerry JDE and its prerequisites from the BlackBerry Web site.

**Prerequisites**
You must have a BlackBerry developer account to download the BlackBerry JDE. You may be required to register if you do not already have an account. Before you download the JDE, ensure the 32-bit JDK has already been installed, even for 64-bit operating systems; otherwise, MDS will not start.

**Task**

**Note:** The BlackBerry JDE is a standalone development environment. The BlackBerry Java Plug-in for Eclipse v1.1 is recommended.

1. Go to the BlackBerry Web site at *http://na.blackberry.com/eng/developers/javaappdev/ javadevenv.jsp* to download and install the BlackBerry JDE.
2. Go to *http://na.blackberry.com/eng/developers/browserdev/devtoolsdownloads.jsp* to download and install the MDS simulator.

**See also**
- *Preferences* on page 13
- *Windows Mobile Client Application .DLL Dependencies* on page 44
- *Installing the BlackBerry Java Plug-in for Eclipse* on page 48
- *Configuring Unwired Server to Use HTTPS for Replication Sync* on page 49
- *Migrating Device Applications* on page 52

# Installing the BlackBerry Java Plug-in for Eclipse

The Device Application Designer supports the BlackBerry Java Plug-in for Eclipse, which allows you to generate the device application code using the Device Application Designer code generation wizard, then debug the generated code.

**Prerequisites**
You must have a BlackBerry developer account to download the BlackBerry Java Plug-in for Eclipse. You may be required to register if you do not already have an account.

**Task**

**Note:** To ensure that you are using the supported version of the BlackBerry Java Plug-in for Eclipse, see the topic *Sybase Unwired Platform 1.5.5 > New Features > Supported Hardware and Software*.

1. Shut down Unwired WorkSpace.
2. Go to *http://us.blackberry.com/developers/javaappdev/* and download the BlackBerry Java Plug-in for Eclipse (full installer) to a temporary folder.
3. Double-click the setup application file.
4. On the Introduction page, click **Next**.
5. Accept or decline the terms of the license agreement and click **Next**.
6. Choose `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Eclipse` as the installation directory and click **Next**.

7. Review the information on the Pre-installation Summary screen and click **Install**.

8. Click **Done**.

**See also**

- *Preferences* on page 13
- *Windows Mobile Client Application .DLL Dependencies* on page 44
- *Downloading the BlackBerry JDE and MDS Simulator* on page 47
- *Configuring Unwired Server to Use HTTPS for Replication Sync* on page 49
- *Migrating Device Applications* on page 52

# Configuring Unwired Server to Use HTTPS for Replication Sync

Enable SSL encryption by configuring the replication-based synchronization HTTPS port.

1. In the left navigation pane of Sybase Control Center for Unwired Platform, expand the **Servers** node and click the server name.

2. Click **Server Configuration**.

3. In the right administration pane, on the **Replication** tab, click **Synchronization Listener**.

4. Select Secure synchronization port as the protocol used for synchronization and configure the certificate properties. In the optional properties section, specify the myserver_identity.crt certificate file using the fully qualified path to the file, along with the password you entered during certificate creation.

   See *Sybase Control Center > Configure > Configuring Unwired Platform > Unwired Server Configuration > Configuring Unwired Server Properties > Replication > Configuring Replication-Based Synchronization Properties.*.

**See also**

- *Preferences* on page 13
- *Windows Mobile Client Application .DLL Dependencies* on page 44
- *Downloading the BlackBerry JDE and MDS Simulator* on page 47
- *Installing the BlackBerry Java Plug-in for Eclipse* on page 48
- *Migrating Device Applications* on page 52

## Generating a Certificate Request

Configure the Web server to use a certificate authority (CA) certificate that you create or another certificate signed by a CA.

You can use either the server's administration tool or the **createcert** utility to generate the certificate request.

1. To create a signed certificate request on the Web server:
   a. Open your server's administration tool. For example, on IIS, open IIS Manager.
   b. Use the administration tool to create a new certificate request, and save the text file.
2. To create a certificate request with **createcert**:
   a. At a command prompt, change to `<UnwiredPlatform_InstallDir>`
      `\UnwiredPlatform\Servers\SQLAnywhere11\BIN32`.
   b. Run:

      **createcert -r**

   c. Follow the prompts to complete the required information. For example:

      ```
      <UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers
      \SQLAnywhere11\BIN32>createcert -r
      SQL Anywhere X.509 Certificate Generator Version 11.0.1.2405
      Enter RSA key length (512-16384): 1024
      Generating key pair...
      Country Code: US
      State/Province: CA
      Locality: Dublin
      Organization: MyCompany, Inc
      Organizational Unit: Engineering
      Common Name: MyCompany
      Enter file path to save request: certreq.txt
      Enter file path to save private key: myclient_private.crt
      Enter password to protect private key: sybase
      ```

      See *System Administration > System Reference > Command Line Utilities > Certificate and Key Management Utilities > Create Certificate (createcert) Utility*.

   Use the generated CA certificate file to sign the server certificate, or do so using the root CA certificate, as described in *System Administration guide > Security Administration > Security Layers > Transport Security Setup > Encrypting Relay Server Connections > Signing the Certificate*.

## Connecting to Unwired Server Using HTTPS from BlackBerry Device Applications

Connect to Unwired Server from BlackBerry device applications using HTTPS.

These are the high-level steps for connecting to Unwired Server from BlackBerry clients. For detailed information, see *System Administration for Sybase Unwired Platform > Security Administration > Security Layers > Transport Security Setup > Encrypting Replication-Based Synchronization Connections*.

1. Follow the procedure in *System Administration for Sybase Unwired Platform > Security Administration > Security Layers > Transport Security Setup > Encrypting Relay Server Connections > Generating a Certificate Request* to generate three certificates—public, private, and identity keys.

2. Copy the three certificate files to: `<UnwiredPlatform_InstallDir>\Servers \SQLAnywhere11\BIN32`.

3. Follow the procedures in *System Administration for Sybase Unwired Platform > Security Administration > Security Layers > Transport Security Setup > Encrypting Replication-Based Synchronization Connections*.

4. Deploy the mobile business object to Unwired Server.

5. Copy the public certificate file to the device folder, for example, `C:\Program Files \Research In Motion\BlackBerry JDE <os-version>\simulator`.

6. Generate the device application code and start the device application.

7. Access the Profile settings screen and in **Stream Params**, enter the certificate information and the stream type, for example:

   ```
   tls_type=rsa;trusted_certificates=C:/Program Files/
   Research In Motion/BlackBerry JDE <os-version>/simulator/
   rsa_public_cert.crt;certificate_company=Sybase;certificate
   _unit=ITS;certificate_name=UEP,MobiLinkStreamType = HTTPS
   ```

8. Check **Secure**.

   The port number for HTTPS is selected automatically if Unwired Server is running HTTPS.

## Connecting to Unwired Server Using Windows Mobile Device Clients

Connect to Unwired Server from Windows Mobile clients.

These are the high-level steps for connecting to Unwired Server from Windows Mobile clients. For detailed information, see *System Administration for Sybase Unwired Platform > Security Administration > Security Layers > Transport Security Setup > Encrypting Replication-Based Synchronization Connections*.

**Note:** This procedure applies only to mobile applications that use replication-based synchronization.

1. Follow the procedure in *System Administration for Sybase Unwired Platform > Security Administration > Security Layers > Transport Security Setup > Encrypting Relay Server Connections > Generating a Certificate Request* to generate three certificates—public, private, and identity keys.

2. Copy the three certificate files to: `<UnwiredPlatform_InstallDir>\Servers \SQLAnywhere11\BIN32`.

3. Follow the procedures in *System Administration for Sybase Unwired Platform > Security Administration > Security Layers > Transport Security Setup > Encrypting Replication-Based Synchronization Connections*.

4. Deploy the mobile business object to Unwired Server.

5. Copy the public certificate file to the `My Documents` folder.

   **Note:** You must put the certificate file in My Documents because when you browse for the file, only the `My Documents` directory is shown on the device.

6. Generate the device application code and start the device application.

7. Access the Profile settings screen and in **Stream Params**, enter the certificate information and the stream type, for example:

   `\Program Files\rsa_public_cert.crt`

8. Check **Secure**.

   The port number for HTTPS is selected automatically if Unwired Server is running HTTPS.

# Migrating Device Applications

Device applications created in Unwired Platform 1.0 or 1.2 are automatically updated for version 1.5.5.

Migration of device application files from Sybase Unwired Platform 1.0 and 1.2 to 1.5.5 is automatic when you:

1. Open the version 1.0 or 1.2 device application file.

   **Note:** Device applications created in Unwired Platform versions earlier than 1.2 are migrated to version 1.2 and then migrated to version 1.5.5.

2. Save the device application file.

3. Re-deploy the device application file.

When opening device application files from earlier Unwired Platform versions, the device application is updated or changed in the following ways:

| Feature | Change |
|---|---|
| User templates | Version control will be added to user template screens. When a create or delete is performed on a template screen, if the template screen's version is not current, the template screen migration will be performed on each of the user template screens. |
| | Template screens that are not migrated are written to a `template_screen_name.xml.bad` file. |

| Feature | Change |
|---------|--------|
| Styles | Device Application Designer after 1.5.2 uses style sets instead of styles, so the style definition of versions earlier than 1.5.2 are no longer valid. When an earlier version of the device application file is opened, the XML structure of the style is silently modified. |
| Tab folder and tab panel | Tab folders can no longer have any sibling controls on the display. When a device application file is opened, if there are controls outside of tab folders, they are removed.<br><br>Actions that reference removed controls are reported by problem check.<br><br>**Tab panel**<br><br>When device application files that have controls within a tab panel are opened, the controls are silently moved to the panel display. |
| Headers and footers | Headers and footers are no longer used and are silently removed. |
| Stock screen properties | Unused and unsupported stock screen properties are removed when the device application file is opened. Platform-specific properties are moved to the RIM section. |
| Stock screens | The About, Push Settings, Screen Saver, and Profile stock screens are now settings screens. When a device application file that contains these screens is opened, those stock screens are removed, as well as connections to and from those screens. |
| Radio button | Style definitions are now defined once for all Radio Buttons within a Radio Group. The style that was assigned to the Radio Button with the lowest ID is now used for the Radio Group.<br><br>A **text_equals_value** attribute has been added to the radio button element definition. During migration, it is checked for each radio button and if the name and the value are identical, the **text_equals_value** attribute is set to true and the value attribute is removed.<br><br>If the value is uniquely defined or defined as NULL, the **text_equals_value** attribute is set to false. |
| Menu items | When a Device Application Designer containing device context menus and Unwired Platform context menus is opened, the create, update, delete, details, and relationship context menu items are silently removed. |
| Hyperlink | During migration the hyperlink attribute **activation** will be removed if it was set, because the logical type is unknown. Verify the device application file and use the content assist feature to define the logical types. |

**See also**
- *Preferences* on page 13
- *Windows Mobile Client Application .DLL Dependencies* on page 44

- *Downloading the BlackBerry JDE and MDS Simulator* on page 47
- *Installing the BlackBerry Java Plug-in for Eclipse* on page 48
- *Configuring Unwired Server to Use HTTPS for Replication Sync* on page 49

# Develop a Device Application

Develop custom device applications from your data source for one or more device platforms.

Once you have developed your mobile business objects (MBOs), you have these options for creating custom device applications:

- In the Mobile Application Diagram, invoke the Generate Code wizard to generate code in Java (BlackBerry devices), C# (Windows Mobile devices), or Objective C (iOS devices), which can be used to call the mobile business object operations. This code can then be imported into an integrated development environment (IDE) of your choice to create the device application. You can also customize the generated object API code in the applicable development environment, where you can then test by deploying to an emulator or device. For example, if you are using Visual Studio to develop custom applications for Windows Mobile devices, generate the C# code from the data source in Eclipse, then customize the client object API code for the device application in Visual Studio.

  **Note:** You can use the Device Application Designer in Eclipse to build a prototype device application; then generate and compile the code for BlackBerry or Windows Mobile device platforms, and test by deploying to an emulator or device.

- Use the Mobile Workflow Forms Editor in Eclipse to generate the code for a mobile workflow package.

See *Sybase Unwired WorkSpace – Device Application Development* for information about the options described above.

## Using Object API to Develop a Device Application

One option for creating custom device applications is to generate object API code on which to build your device application.

Unwired Platform provides the Code Generation wizard for generating object API code. Code generation creates the business logic, attributes, and operations for your Mobile Business Object. You can generate code for these platforms:

- RIM BlackBerry
- Windows Mobile
- Windows
- iOS

See the guidelines for generating code for each platform type.

See the *Sybase Unwired Platform Developer Reference* for each platform for information about the client object API.

## Generating Object API Code

Generate object API code containing mobile business object (MBO) references, which allows you to use APIs to develop device applications for various mobile devices.

### Prerequisites

Before generating device client code, develop the MBOs that will be referenced in the device applications you are developing.

### Task

**1.** Launch the **Code Generation** wizard.

| From | Action |
|---|---|
| **The Mobile Application Diagram** | Right-click within the Mobile Application Diagram and select **Generate Code**. |
| **WorkSpace Navigator** | Right-click the Mobile Application project folder that contains the mobile objects for which you are generating API code, and select **Generate Code**. |

**2.** (Optional) Enter the information for these options:

**Note:** This page of the code generation wizard is seen only if you are using the Advanced developer profile.

| Option | Description |
|---|---|
| Select code genera-tion configuration | Select either an existing configuration that contains code generation set-tings, or generate device client code without using a configuration:<br>• Continue without a configuration – select this option to generate device code without using a configuration.<br>• Select an existing configuration – select this option to either select an existing configuration from which you generate device client code, or create a new configuration. Selecting this option enables:<br>  • Select code generation configuration – lists any existing configu-rations, from which you can select and use for this session. You can also delete any and all existing saved configurations.<br>  • Create new configuration – enter the **Name** of the new configura-tion and click **Create** to save the configuration for future sessions. Select an existing configuration as a starting point for this session and click **Clone** to modify the configuration. |

**3.** Click **Next**.

**4.** In Select Mobile Objects, select all the MBOs in the mobile application project or select MBOs under a specific synchronization group, whose references, metadata, and dependencies (referenced MBOs) are included in the generated device code.

Dependent MBOs are automatically added (or removed) from the Dependencies section depending on your selections.

Unwired Server automatically computes the page size after you choose the MBOs based on total attribute size. If an MBO's accumulated attribute size is larger than the page size setting, a warning displays.

**Note:** Code generation fails if the server-side (run-time) enterprise information system (EIS) data sources referenced by the MBOs in the project are not running and available to connect to when you generate object API code.

**5.** Click **Next**.

**6.** Enter the information for these configuration options:

| Option | Description |
|---|---|
| Language | Choose the language used for developing the client applications:<br>• Java<br>• C#<br>• Objective-C |
| Platform | Select the platform ( target device) from the drop-down list for which the device client code is intended. The platform is dependent on the language selected.<br>• Java<br>  • Java SE<br>  • Java ME for BlackBerry<br>• C#<br>  • NET Framework for Windows<br>  • NET Compact Framework 3.5 for Windows Mobile<br>• Objective C<br>  • iOS |
| Unwired Server | Specify a default Unwired Server connection profile to which the generated code connects at runtime. |
| Server domain | Choose the domain to which the generated code will connect. If you specified an Unwired Server to which you previously connected successfully, the first domain in the list is chosen by default. You can enter a different domain manually.<br><br>**Note:** This field is only enabled when an Unwired Server is selected. |

| Option | Description |
|---|---|
| Page size | Optionally, select the page size for the generated client code. If the page size is not set, the default page size is 4KB at runtime. The default is a proposed page size based on the selected MBO's attributes. |
| | The page size should be larger than the sum of all attribute lengths for any MBO that is included with all the MBOs selected, and must be valid for the database. If the page size is changed, but does not meet these guidelines, object queries that use string or binary attributes with a **WHERE** clause may fail. |
| | **Note:** This field is only enabled when an Unwired Server is selected. The page size option is not enabled for message-based applications. |
| Package, Namespace, or Name Prefix | • Package – enter a package name for Java.<br>• Namespace – enter a namespace for C#.<br>• Name Prefix – enter a name prefix for Objective C. |
| Destination | Specify the destination of the generated device client files. Enter (or **Browse**) to either a **Project path** (Mobile Application project) location or **File system path** location. Select **Clean up destination before code generation** to clean up the destination folder before generating the device client files. |
| | **Note:** If you select Java as the language, enter a project path, specify a mobile application project folder, and select Generated Code as the destination. JAR files are automatically added to the destination for the platform that supports compiling of the generated client code. |
| Replication-based | Select to use replication-based synchronization. |
| | **Note:** This option is not available for mobile applications that will run on iOS devices. |

| Option | Description |
|---|---|
| Message-based | Select to use message-based synchronization. |
| | **Note:** Use message based synchronization for iOS and Windows Mobile mobile applications. This option is not available for Java applications. |
| Backward compatible | Select if you want the generated code to be compatible with the SUP 1.2 release. |

**7.** Select **Generate metadata classes** to generate metadata for the attributes and operations of each generated client object.

**8.** Select **Generate metadata and object manager classes** to generate both the metadata for the attributes and operations of each generated client object and an object manager for the generated metadata.

   The object manager allows you to retrieve the metadata of packages, MBOs, attributes, operations, and parameters during runtime using the name instead of the object instance.

**9.** If you selected Java as the language, you can select **Generate JavaDoc** to include the Object API JavaDoc documentation in the output directory.

**10.** Click **Finish**.

### Replication-based Synchronization Versus Message-based Synchronization
When choosing between generating code for replication-based synchronization or message-based synchronization, keep the following in mind:

A mobile application can be either message-based or replication-based, but not both at the same time. A package of mobile business objects can be deployed as either message-based or replication-based.

*Replication-based synchronization*

- Supported on BlackBerry and Windows Mobile devices.
- Data flow follows an upload/download pattern.
- Data is synchronous (supports background synchronization).
- Uses the "poke-pull" model of push, where a notification is pushed to the device (poke), and the device fetches the content (pull).
- Best for mostly disconnected mode.

*Message-based synchronization*

- Supported on iOS and Windows Mobile devices.
- Uses an asynchronous messaging protocol.
- Provides reliable messaging between the device and the server.

- Provides fine-grained synchronization (synchronization is provided at the data level—each process communicates only with the process it depends on).
- Best for always available mode.

### Generating C# Object API Code
Generate object API code for applications that will run on Windows Mobile.

1. Launch the **Code Generation** wizard.

| From | Action |
|------|--------|
| **The Mobile Application Diagram** | Right-click within the Mobile Application Diagram and select **Generate Code**. |
| **WorkSpace Navigator** | Right-click the Mobile Application project folder that contains the mobile objects for which you are generating API code, and select **Generate Code**. |

2. (Optional) Enter the information for these options:

**Note:** This page of the code generation wizard is seen only if you are using the Advanced developer profile.

| Option | Description |
|--------|-------------|
| Select code generation configuration | Select either an existing configuration that contains code generation settings, or generate device client code without using a configuration:<br>• Continue without a configuration – select this option to generate device code without using a configuration.<br>• Select an existing configuration – select this option to either select an existing configuration from which you generate device client code, or create a new configuration. Selecting this option enables:<br>  • Select code generation configuration – lists any existing configurations, from which you can select and use for this session. You can also delete any and all existing saved configurations.<br>  • Create new configuration – enter the **Name** of the new configuration and click **Create** to save the configuration for future sessions. Select an existing configuration as a starting point for this session and click **Clone** to modify the configuration. |

3. Click **Next**.
4. In Select Mobile Objects, select all the MBOs in the mobile application project or select MBOs under a specific synchronization group, whose references, metadata, and dependencies (referenced MBOs) are included in the generated device code.

   Dependent MBOs are automatically added (or removed) from the Dependencies section depending on your selections.

**Note:** Code generation fails if the server-side (run-time) enterprise information system (EIS) data sources referenced by the MBOs in the project are not running and available to connect to when you generate object API code.

5. Click **Next**.
6. Enter the information for these configuration options:

| Option | Description |
|---|---|
| Language | Select **C#**. |
| Platform | Select the platform ( target device) from the drop-down list for which the device client code is intended.<br>• NET Framework for Windows<br>• NET Compact Framework 3.5 for Windows Mobile |
| Unwired Server | Specify a default Unwired Server connection profile to which the generated code connects at runtime. |
| Server domain | Choose the domain to which the generated code will connect. If you specified an Unwired Server to which you previously connected successfully, the first domain in the list is chosen by default. You can enter a different domain manually.<br><br>**Note:** This field is only enabled when an Unwired Server is selected. |
| Page size | Optionally, select the page size for the generated client code. If the page size is not set, the default page size is 16KB at runtime. The default is a proposed page size based on the selected MBO's attributes.<br><br>The page size should be larger than the sum of all attribute lengths for any MBO that is included with all the MBOs selected, and must be valid for the database. If the page size is changed, but does not meet these guidelines, object queries that use string or binary attributes with a **WHERE** clause may fail.<br><br>**Note:** This field is only enabled when an Unwired Server is selected. The page size option is not enabled for message-based applications. |

| Option | Description |
|--------|-------------|
| Namespace | Enter a namespace for C#. |
| | **Note:** The namespace name should follow naming conventions for C#. Do not use ".com" in the namespace. |
| Destination | Specify the destination of the generated device client files. Enter (or **Browse**) to either a **Project path** (Mobile Application project) location or **File system path** location. Select **Clean up destination before code generation** to clean up the destination folder before generating the device client files. |
| Replication-based | Select to use replication-based synchronization. |
| Message-based | Select to use message-based synchronization. |
| Backward compatible | Select so the generated code is compatible with the SUP 1.2 release. |

7.  Select **Generate metadata classes** to generate metadata for the attributes and operations of each generated client object.

8.  Select **Generate metadata and object manager classes** to generate both the metadata for the attributes and operations of each generated client object and an object manager for the generated metadata.

   The object manager allows you to retrieve the metadata of packages, MBOs, attributes, operations, and parameters during runtime using the name instead of the object instance.

9.  Click **Finish**.

### Generating Java Object API Code

Generate object API code for applications that will run on BlackBerry devices.

1.  Launch the **Code Generation** wizard.

| From | Action |
|------|--------|
| **The Mobile Application Diagram** | Right-click within the Mobile Application Diagram and select **Generate Code**. |
| **WorkSpace Navigator** | Right-click the Mobile Application project folder that contains the mobile objects for which you are generating API code, and select **Generate Code**. |

2.  (Optional) Enter the information for these options:

**Note:** This page of the code generation wizard is seen only if you are using the Advanced developer profile.

| Option | Description |
|---|---|
| Select code genera-tion configuration | Select either an existing configuration that contains code generation set-tings, or generate device client code without using a configuration: <br>• Continue without a configuration – select this option to generate device code without using a configuration. <br>• Select an existing configuration – select this option to either select an existing configuration from which you generate device client code, or create a new configuration. Selecting this option enables: <br>   • Select code generation configuration – lists any existing configu-rations, from which you can select and use for this session. You can also delete any and all existing saved configurations. <br>   • Create new configuration – enter the **Name** of the new configura-tion and click **Create** to save the configuration for future sessions. Select an existing configuration as a starting point for this session and click **Clone** to modify the configuration. |

3. Click **Next**.
4. In Select Mobile Objects, select all the MBOs in the mobile application project or select MBOs under a specific synchronization group, whose references, metadata, and dependencies (referenced MBOs) are included in the generated device code.

   Dependent MBOs are automatically added (or removed) from the Dependencies section depending on your selections.

   **Note:** Code generation fails if the server-side (run-time) enterprise information system (EIS) data sources referenced by the MBOs in the project are not running and available to connect to when you generate object API code.

5. Click **Next**.
6. Enter the information for these configuration options:

| Option | Description |
|---|---|
| Language | Select **Java**. |

| Option | Description |
|---|---|
| Platform | Select the platform ( target device) from the drop-down list for which the device client code is intended.<br>• Java SE<br>• Java ME for BlackBerry<br><br>**Note:** When generating code into a plain Java project with language 'Java' and platform 'Java Me for BlackBerry', compilation errors are generated because of code references to RIM API's. To avoid errors, generate code into a BlackBerry project. |
| Unwired Server | Specify a default Unwired Server connection profile to which the generated code connects at runtime. |
| Server domain | Choose the domain to which the generated code will connect. If you specified an Unwired Server to which you previously connected successfully, the first domain in the list is chosen by default. You can enter a different domain manually.<br><br>**Note:** This field is only enabled when an Unwired Server is selected. |
| Page size | Optionally, select the page size for the generated client code. If the page size is not set, the default page size is 4KB at runtime. The default is a proposed page size based on the selected MBO's attributes.<br><br>The page size should be larger than the sum of all attribute lengths for any MBO that is included with all the MBOs selected, and must be valid for the database. If the page size is changed, but does not meet these guidelines, object queries that use string or binary attributes with a **WHERE** clause may fail.<br><br>**Note:** This field is only enabled when an Unwired Server is selected. |
| Package | Enter a name for the Java package. |

| Option | Description |
|---|---|
| Destination | Specify the destination of the generated device client files. Enter (or **Browse**) to a **Project path** (Mobile Application project) location, and select Generated Code as the destination. JAR files are automatically added to the destination project.<br><br>Select **Clean up destination before code generation** to clean up the destination folder before generating the device client files. |
| Replication-based | Select to use replication-based synchronization for the application. |
| Message-based | This option is not supported for Java applications. |
| Backward compatible | Select so the generated code is compatible with the SUP 1.2 release. |

7. Select **Generate metadata classes** to generate metadata for the attributes and operations of each generated client object.

8. Select **Generate metadata and object manager classes** to generate both the metadata for the attributes and operations of each generated client object and an object manager for the generated metadata.

   The object manager allows you to retrieve the metadata of packages, MBOs, attributes, operations, and parameters during runtime using the name instead of the object instance.

9. (Optional) Select **Generate JavaDoc** to generate API documentation from the source code.

10. Click **Finish**.

### Generating Objective-C Object API Code
Generate Objective-C code for applications that will run on Apple devices.

1. Launch the **Code Generation** wizard.

| From | Action |
|---|---|
| **The Mobile Application Diagram** | Right-click within the Mobile Application Diagram and select **Generate Code**. |
| **WorkSpace Navigator** | Right-click the Mobile Application project folder that contains the mobile objects for which you are generating API code, and select **Generate Code**. |

2. (Optional) Enter the information for these options:

**Note:** This page of the code generation wizard is seen only if you are using the Advanced developer profile.

| Option | Description |
|---|---|
| Select code genera-tion configuration | Select either an existing configuration that contains code generation set-tings, or generate device client code without using a configuration:<br>• Continue without a configuration – select this option to generate device code without using a configuration.<br>• Select an existing configuration – select this option to either select an existing configuration from which you generate device client code, or create a new configuration. Selecting this option enables:<br>  • Select code generation configuration – lists any existing configu-rations, from which you can select and use for this session. You can also delete any and all existing saved configurations.<br>  • Create new configuration – enter the **Name** of the new configura-tion and click **Create** to save the configuration for future sessions.<br>    Select an existing configuration as a starting point for this session and click **Clone** to modify the configuration. |

3. Click **Next**.
4. In Select Mobile Objects, select all the MBOs in the mobile application project or select MBOs under a specific synchronization group, whose references, metadata, and dependencies (referenced MBOs) are included in the generated device code.

   Dependent MBOs are automatically added (or removed) from the Dependencies section depending on your selections.

   **Note:** Code generation fails if the server-side (run-time) enterprise information system (EIS) data sources referenced by the MBOs in the project are not running and available to connect to when you generate object API code.

5. Click **Next**.
6. Enter the information for these configuration options:

| Option | Enter |
|---|---|
| Language | Objective-C |
| Platform | iPhone |
| Unwired Server | Specify a default Unwired Server connection profile to which the generated code connects at runtime. |

| Option | Enter |
|--------|-------|
| Server domain | Choose the domain to which the generated code will connect. If you specified an Unwired Server to which you previously connected successfully, the first domain in the list is chosen by default. You can enter a different domain manually.<br><br>**Note:** This field is only enabled when an Unwired Server is selected. |
| Page size | Not enabled for Objective-C. |
| Name Prefix | Enter a name prefix for Objective C. |
| Destination | Specify the destination of the generated device client files. Enter (or **Browse**) to either a **Project path** (Mobile Application project) location or **File system path** location.<br><br>Select **Clean up destination before code generation** to clean up the destination folder before generating the device client files. |
| Replication-based | This option is not available for iOS. |
| Message-based | Selected by default. |
| Backward compatible | Not enabled for iOS. |

**7.** Select **Generate metadata classes** to generate metadata for the attributes and operations of each generated client object.

**8.** Select **Generate metadata and object manager classes** to generate both the metadata for the attributes and operations of each generated client object and an object manager for the generated metadata.

The object manager allows you to retrieve the metadata of packages, MBOs, attributes, operations, and parameters during runtime using the name instead of the object instance.

**9.** Click **Finish**.

### Generated Code Location and Contents

Generated object API code is stored in the project's Generated Code sub-folder by default, for example, `C:\Documents and Settings\administrator\workspace \<Unwired Platform project name>\Generated Code\src`. Language, platform, and whether or not you select the Generate metadata classes option determines the class files generated in this folder.

Assuming you generate code in the default location, you can access it from WorkSpace Navigator by expanding the Mobile Application project folder for which the code is generated, and expand the Generated Code folder.

The contents of the folder is determined by the options you selected from the Generate Code wizard, and include generated class (.java, .cs, .h, .m) files that contain:

- MBO - the business logic of your MBO.
- Synchronization parameters - any synchronization parameters for the MBOs.
- Init - Unwired Server connection information used by device applications to access MBOs.
- Personalization - personalization and personalization synchronization parameters used by the MBOs.
- Metadata - if you selected **Generate metadata classes**, the metadata classes which allow you to use code completion and compile-time checking to ensure that run-time references to the metadata are correct.

### Validating Generated Code

Validation rules are enforced when generating client code for C# and Java. Define prefix names in the Mobile Business Object Preferences page to correct validation errors.

Sybase Unwired WorkSpace validates and enforces identifier rules and checks for key word conflicts in generated Java and C# code. For example, by displaying error messages in the Properties view or in the wizard. Other than the known name conversion rules (converting '.' to '_', removing white space from names, and so on), there is no other language specific name conversion. For example, cust_id is not changed to custId.

You can specify the prefix string for mobile business object, attribute, parameter, or operation names from the Mobile Business Object Preferences page. This allows you to decide what prefix to use to correct any errors generated from the name validation.

1. Select **Window > Preferences**.
2. Expand **Sybase, Inc > Mobile Development**.
3. Select **Mobile Business Object**.
4. Add or modify the **Naming Prefix** settings as needed.

The defined prefixes are added to the names (object, attribute, operation, and parameter) whenever these are auto-generated. For example, when you drag-and-drop a data source onto the Mobile Application Diagram.

## Building an iOS Application Using the Object API

Generate code to build a device application that runs on iOS devices.

1. Create a Mobile Application Project.
2. Develop the mobile business objects to which the object API will refer.
3. Deploy the mobile application project to Unwired Server.

4. Use the Code Generation wizard to generate object API code using Objective-C as the language
5. Set up the iOS client application in XCode.

   See the *Developer Reference for iOS*.

## Building a BlackBerry Application Using the Object API

Generate code to build a device application that runs on RIM BlackBerry devices.

1. Create a Mobile Application Project.
2. Develop the mobile business objects to which your code will refer.
3. Deploy the mobile application project to Unwired Server.
4. Use the Code Generation wizard to generate object API code using Java as the language and Java ME for BlackBerry as the platform.
5. Create the BlackBerry JDE project that will contain the generated object API code and customized UI code.
6. From the `eclipse\plugins\com.sybase.uep.bob.rim\generate \blackberry` directory, copy the `UltraLiteJ.cod` and `UltraLite.alx` files to the Simulator directory.
7. Build the BlackBerry project.

   See the *Developer Reference for BlackBerry*.

## Developing Device Applications for Windows Mobile and Windows Platforms

Build a device application that runs on Windows Mobile and Windows platforms.

1. From the Visual Studio main menu select **File > New > Project**.
2. In the Visual C# project type, select **SybaseMobileApplication** and click **OK**.
3. In the next screen, select:
   - Target platform:
     - Windows Mobile 5.0 Professional
     - Windows Mobile 6.0, 6.1, and 6.5 Professional and Standard
     - Windows
   - Library version – .NET version 3.5
   - Language – the language used in the resource DLLs, to be included in the generated project.

   Different sets of DLLs are included in the project based on your selections. The project contains all assemblies and runtime support DLLs required to access the Object API.
4. (Optional) Select **Include SUP Windows Mobile Controls** to reference the dependent Sybase Windows Mobile Component Library Assemblies in the generated project.

This allows you to use Sybase Unwired platform UI controls.

5. Click **OK** to generate the Visual Studio Project with the dependent Sybase Unwired Platform .NET assemblies.

6. Build the Solution. From the Visual Studio main menu select **Build > Build Solution**.

    The DLLs are copied to the target directory and the directory structure is flattened.

    Once generated and built, you can write custom mobile applications based on your selections.

7. Develop the mobile business objects (MBOs) that implement the business logic.

    See these online help topics:
    * *Sybase Unwired Platform 1.5 > Sybase Unwired Workspace 1.5 – Eclipse Edition > Develop > Developing a Mobile Business Object*

8. Use the Code Generation wizard to generate the C# Object API code for the mobile business object.

9. Add the generated code to the new project you created from the template.

    For more information, see the *Rebuilding the Generated Solution in Visual Studio*.

### Generating Online Help for Visual Studio

You can generate online help for your mobile application project using the .xml file that is generated with the Object API code.

When you generate the C# Object API code for the mobile application project, an .xml file is generated for the mobile business objects. You can use the generated .xml file to build online help for the mobile application project in Visual Studio.

1. Use the Code Generation wizard to generate the C# Object API Code.

2. Build the generated solution in Visual Studio:

    a) In Visual Studio, select **File > Open > Project/Solution**.
    b) Browse to the solution file (.sln) you want to open and double-click the file.
    c) In Solution Explorer, right-click the solution and select **Rebuild Solution**.
    d) Select **File > Save**.
       The source `.xml` file for building online help is included with the generated project, which contains all assemblies and runtime support DLLs required to access the object API.

3. There are different tools available for building online help. This procedure uses the free tool called Sandcastle Help File Builder. You can download Sandcastle Help File Builder installer from *http://shfb.codeplex.com*.

4. Create a SandCastle Helpfile builder project (`.shfb`) in Sand Castle File Builder by specifying the assemblies and the generated xml file as input.

5. Use the `.shfb` project file in the daily build script to build the document. For example:

```
<Target Name="Documentation">
  <Exec Command="$(SandCastleHelpBuilderPath)
```

```
Infrastructure.Core.shfb" />
</Target>
```

### Windows Mobile Device Framework Assemblies

The Windows Mobile Device Framework consists of the following three assemblies, which provide device component integration, custom controls, and Device Application Designer actions:

**Table 2. Sybase Windows Device Framework Assemblies**

| Name | Function |
|------|----------|
| Sybase.UnwiredPlatform.Windows | Provides drawing improvement and device component integration. For example, integration with the device's phone, e-mail, or camera functionalities. |
| Sybase.UnwiredPlatform.Windows.Forms | Provides customized control, MVP pattern, action framework, and validation framework. |
| Sybase.UnwiredPlatform.Windows.StockScreens | Provides Device Application Designer specific features. For example, stock screens, prepared stock actions, and variables. |

### Rebuilding the Generated Solution in Visual Studio

After generating the C# Object API code for a Mobile Application project, you can modify the code in Visual Studio.

#### Prerequisites

Visual Studio must be installed.

#### Task

When you generate the Object API code for a Windows Mobile device using the Mobile Application project Code Generation wizard, the Visual Studio solution is saved to the folder you specified.

**Note:** If you generate 32-bit Windows code on a 64-bit machine, in the Visual Studio Configuration Manager for building and deploying the active solution, change the Active Solution Platform from "AnyCPU" to "x86" before deploying and running the application.

1. In Visual Studio, select **File > Open > Project/Solution**.
2. Browse to the solution file (.sln) you want to open and double-click the file.
3. In Solution Explorer, right-click the solution and select **Rebuild Solution**.
4. Select **File > Save**.
   You can now open the form for which you want to modify the code.

# Prototyping a Device Application Using the Designer

You can use the Device Application Designer to create prototype Windows Mobile and BlackBerry device applications.

## Prototyping a Device Application Using the Device Application Designer

A device application is a software application that runs on a mobile device. You can create prototype device applications for BlackBerry and Windows Mobile devices using the Device Application Designer.

1. *Accessing the Device Application Designer*

   Access the Device Application Designer.

2. *Designing the Flow of Device Applications*

   Use the Flow Design page to create the screen flow for the mobile application.

3. *Adding Screens from the Palette to the Flow Design*

   Use the Flow Design page to add and connect screens you design, as well as customized stock screens.

4. *Adding Connections to Screens*

   The Start icon is displayed on the Flow Design page, and its connection is attached to the first screen you add to the Flow Design. You can change the target to any screen in the Flow Design.

5. *Selecting a Device*

   Select a device to display an empty canvas in the Screen Design page, formatted for that device.

6. *Adding Controls to the Screen*

   Use the Screen Design Palette to add controls to your device application screen.

7. *Generating Code For a BlackBerry Device Application*

   Use the Generate Device Application wizard to generate the device application code.

### The Device Application Designer

The Device Application Designer consists of views and pages that allow you to model the interface of the device interface screens.

You can view the device application in two ways:

- Outline ⓘ – shows all the objects in the designer, for all pages except the Introduction or Source page. The outline includes context menus that match the context menu selection for the objects on the page. If you select a node on the tree outline, the corresponding node in the editor is highlighted, and vice versa. The outline view is related to the object model, so delete and command stack actions appear in this view. For example, if you delete an item, undo and redo commands appear; the same as in the editor.

- Thumbnail ⓘ – shows the Flow Design page as a thumbnail. Therefore, when you design a screen and attach actions to buttons on the Screen Design page, the flow changes, and you can see the change in the thumbnail view. The thumbnail also gives context for the screen you are working on with respect to the entire application design. This view includes the standard viewable region locator for scrolling the image.

The Device Application Designer has drag-and-drop functionality, which allows you to drag and drop mobile business objects of certain types and settings onto the Flow design canvas to create screens automatically. You can also drag and drop mobile business objects onto controls that accept actions.

The Device Application Designer contains these pages:

**Introduction**

Creating a new custom device application or double-clicking an existing one, opens the Introduction section of the Device Application Designer. This page is the starting point for designing your custom device application. You can access help and launch tutorials from this page.

**Flow Design**

Use this page to specify the manner in which screens link together. You can then link specific screens and actions to a mobile business object on a server. The Palette view shows available screens, alerts, and connections.

The Flow Design page displays an empty canvas with the Start icon, and the Palette to the right, which shows the available stock screens, settings screens, alerts, and connections. You can drag and drop items from the Palette to the canvas. Screens that you drag to the Flow Design canvas display as icons and show the connection lines between them. Dragging a new screen into this page starts a new screen. Double-clicking a screen opens the Screen Design page.

The Settings Screens box is for the pre-built, single-page screens that are included in the client settings. Drag and drop settings screens, such as Screen Saver or Push Settings, from the Palette to the Settings Screens box. There are no connections or alerts to or from settings screens.

Right-clicking anywhere in the Flow Design page displays a context menu.
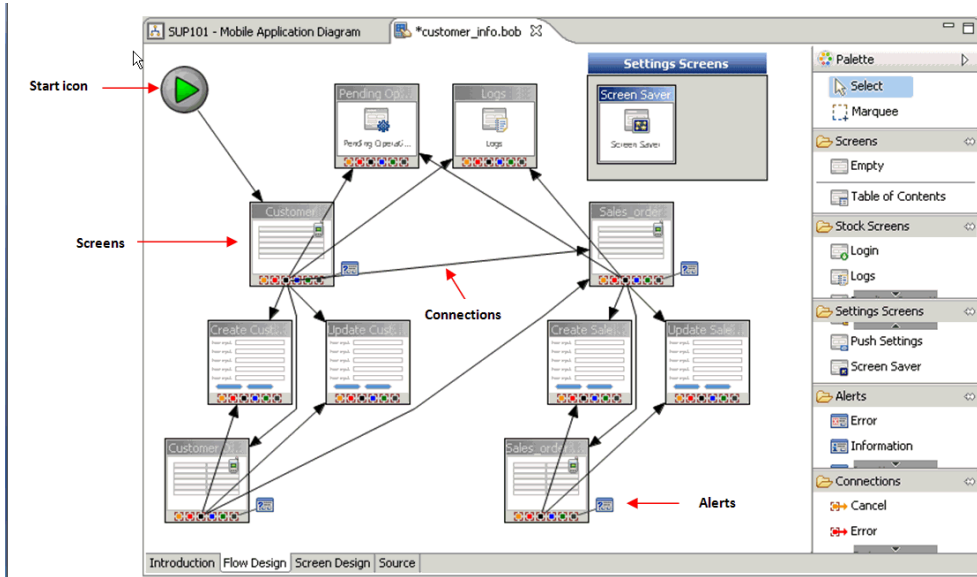
**Screen Design**

Use this page to design the screen of your custom device application. The Palette view shows the menu, controls, containers, navigation bar, toolbar, and actions you can use to design a new screen. Drag controls to design the user interface on the device screen.

The Device Application toolbar includes a list of existing screens.
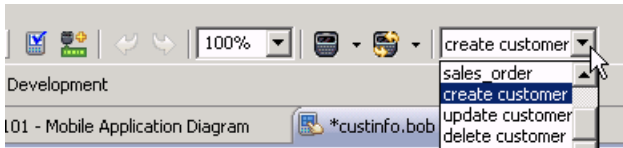


You can open a different screen by selecting the corresponding screen name from the list.

**Source**

This page appears only if you have configured it in the Device Application Designer Appearance preferences. This page shows the XML code of the custom device application. You can edit the XML code directly in the Source page, and changes are immediately reflected in the Screen Design and Flow Design pages.

### Accessing the Device Application Designer

Access the Device Application Designer.

1. In the Mobile Development perspective, select **File > New > Device Application Designer**.

Alternatively, in WorkSpace Navigator, right-click any Mobile Application Project folder and select **Create Device Application Designer**. This option makes automatic entries in some of the fields in the Device Application Designer wizard, such as the Parent folder.

**2.** In the New Device Application Designer wizard, enter the following information, then click **Finish**:

| Option | Description |
|---|---|
| **Parent folder** | Enter or select the parent folder under which you want to save the device application. |
| **File name** | Accept the default, or enter a new name for the device application. The extension for the file name is .bob. |
| **Advanced** | Select **Link to file in file system** to link the device application to a file.<br><br>Click **Browse** to locate the file to which you want to link.<br><br>Click **Variables** to select a path variable, or create a new one. |
| **Connection profile** | Accept the default connection profile, or select an existing profile from the drop-down list. To create a new connection profile, click **Create Profile**. |
| **Platform device** | Select the platform and device for which you are creating the device application. This selection determines which features are available. |

The Device Application Designer appears.

**3.** Click **Flow Design** to start designing a device application.

## Selecting a View
Choose to view the device application in either Outline or Thumbnail view.

**1.** From the main menu select **Window > Show View > Outline**.
The Outline view appears.
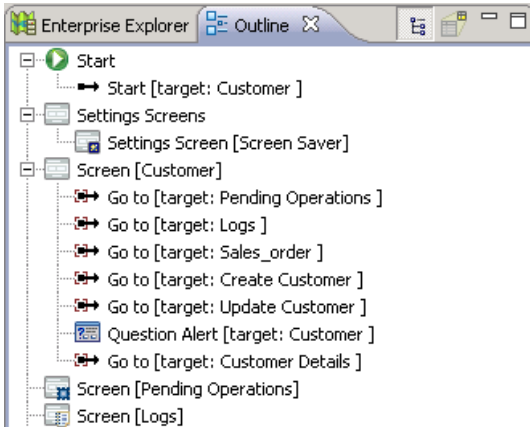
**2.** To switch to Thumbnail view, click the Thumbnail icon 📄.

### Device Application Designer Context Menus

You can right-click in any of the pages (except the Introduction page) of the Device
Application Designer to display a context menu and perform these actions.

| Menu Item | Description |
|-----------|-------------|
| Undo | Undoes the last action performed. |
| Redo | Re-does an action that you undid. |
| Cut | Cut a selected object. |
| Copy | You can copy and paste a single screen at a time from one flow design to another.<br><br>You can also copy and paste actions in the Screen Design page, on the same screen. This is useful when creating a complex action on a menu and you want the same thing on a button in the screen. |
| Paste | Paste an object into a new location that you cut or copied. |
| Delete | Delete a selected object. |
| Select All | Selects all the objects on the Device Application Designer display screen. |
| Preferences | Takes you to the Device Application Designer Preferences dialog, where you can configure the appearance of the Device Application Designer pages. |

| Menu Item | Description |
| --- | --- |
| Image References | Shows the available image references and also allows you to create new image references, and edit or delete existing image references. |
| Variable References | Shows current variable references and allows you to create new user variable references and edit and delete existing variable references. |
| Style References | Shows current style references and allows you to create new style references and edit and delete existing style references. |
| Unwired Platform Profile References | Shows the current Unwired Platform profiles and also allows you to create new profiles and edit or delete existing profiles. |
| Mobile Business Object References | Shows current mobile business object references and allows you to create new mobile business object references and edit or delete existing mobile business object references.<br><br>You can also configure initial synchronization parameters here. |
| Show Properties View | Takes you to the Properties page for the current screen and active control. |
| Verify | Verifies the controls and actions on the screen as well as the screens and connections in the Flow Design, and displays any problems found in the Problems view. |
| Clear Verify Problems | Clears the Problems view. |
| Generate Device Application | Select to generate the device application. |
| Screens | This menu item appears only on the Screen Design page. It allows you to switch to any of the other screens in the Flow Design. |

| Menu Item | Description |
|---|---|
| View | Choose from these views:<br><br>• Zoom In – select to zoom in on the objects in the canvas.<br>• Zoom Out – select to zoom out on the objects in the canvas.<br>• Grid – this menu item appears only in the Flow Design page of the Device Application Designer. If selected, a grid displays on the canvas and snaps objects to the grid when you move them.<br>• Snap to Geometry – this menu item appears only in the Flow Design page. If selected, lines of geometry for aligning objects display. |
| Remove Tab Panel | If there is only one tab panel on the screen, use this option to remove the tab panel and move contents of the tab panel into the screen display. |
| Align | If you select more than one screen in the Flow Design page, Alignment options appear on the context menu, which include:<br><br>• Alight Left<br>• Align Center<br>• Align Right<br>• Align Top<br>• Align Middle<br>• Align Bottom |
| Remove Unused Alerts and Connections | This menu item appears only if you are on the Flow Design page. |
| Update Table Of Contents | This menu item appears only if you right-click on a Table of Contents screen in the Flow Design page. Select to create "Go To" connections to any screen that currently does not exist in the table of contents. This creates a list item for each connection created by the table of contents. |

| Menu Item | Description |
|---|---|
| Create Screen Template | This menu item appears only when you have a screen selected in the Flow Design page. This option adds a new palette screen entry after the separator for the stock screens, with the name selected screen. The template contains all the same content as the screen for which it was created. When you drag the screen template on to the Flow Design, it creates a new screen with all of the defined items of the original screen. |
| Flow Design | This option appears in the context menu on the Flow Design page when you do not have any screens selected. The options include:<br><br>• Add Mobile Business Object Screen<br>• Add Operation Screen |
| Cell Table | This option appears in the context menu when you select the cell table control in the Screen Design page. It allows you to select a mobile business object with which to associate the cell table. |
| Grid Table | This option appears in the context menu when you select the grid table control in the Screen Design page. It allows you to select a mobile business object with which to associate the grid table. |
| List Detail | This option appears in the context menu when you select the list detail control in the Screen Design page. It allows you to select a mobile business object with which to associate the list detail. |

**Customizing the Device Application Designer Palette**
Use the Customize Palette dialog to hide items you do not use on the Device Application Designer Palette, and to delete user-created screen templates.

1. Right-click anywhere on the Palette and select **Customize**.
2. In the Customize Palette dialog, select the item to hide from the left pane, then select the **Hide** checkbox from the right pane.
3. Click **Apply**.
4. Click **OK**.

### Defining a Variable for a Path Definition

Use the Advanced options in the New Designer Application Designer wizard to specify file path locations.

1. In the New Device Application Designer wizard, click **Advanced**.
2. Select **Link to File in the File System** and click **Browse** to locate the file or click **Variables** to define a new path variable.
3. (Optional) In New Variable, enter the information for the new variable, and click **OK**:

| Option | Description |
|---|---|
| **Name** | Enter a name for the new variable. |
| **Location** | Enter the location of the variable, or click **File** to select a file. Click **Folder** to select a folder. |

*Path Variables*

Path variables specify locations on the file system.

You can specify the location of linked resources relative to these path variables, so you can avoid references to fixed locations on your file system. Using a path variable allows you to share projects containing linked resources with other users, without requiring them to have the exact same directory structure on their file system as you have on your file system.

## Designing the Flow of Device Applications

Use the Flow Design page to create the screen flow for the mobile application.

1. In the Device Application Designer, click **Flow Design**.
2. In the Flow Design page, add device screens to the canvas by either of these methods:
   - Drag and drop a mobile business object from a mobile application project onto the Flow Design canvas to create screens automatically.
   - Drag and drop screens from the Flow Design palette.

   The Start icon on the Flow Design page is automatically attached to the first screen you drag and drop onto the Flow Design page. You can change the target.
3. Add additional screens, as needed.
4. Add connections between the screens.
5. Click **Screen Design** to add controls and actions to the screens.

**Flow Design**

When designing the screen flow for your mobile application, remember the special considerations for designing a user interface for a mobile device.

The goal when designing the screen flow for a mobile device is to allow users to navigate screens as intuitively as possible. The Flow Design page allows you to visualize the screen flow for the device application.

Mobile users expect to access information quickly. For example, an inventory management system can provide a large quantity of information, but users need only a small portion of that information at one time. Limit the information to only the amount that is necessary for the application to be useful and for the user to be able to complete their task efficiently.

- Keep in mind the limitations of mobile devices while designing the screen flow:
    - Memory limitations
    - Battery life
    - Small screen size
    - Only one screen at a time appears to the user

**Flow Design Page Properties**

Describes the properties for the Flow Design page.

This table shows the properties that are displayed for the different objects on the Flow design canvas when they are selected.

| Object | Properties/Description |
|---|---|
| Starting Point | The starting point properties dictate where the Start icon is placed in the graphical editor pane.<br><br>• X – this is the X-coordinate of the Start icon and must be a positive integer.<br>• Y – this is the Y-coordinate of the Start icon and must be a positive integer. |

| Object | Properties/Description |
|---|---|
| Screen | This displays the properties of the selected screen on the Flow Design page. The properties include:<br><br>• Name – title of the selected screen. This cannot be empty.<br>• X – X-coordinate of the screen icon in the graphical editor pane.<br>• Y – Y-coordinate of the screen icon in the graphical editor pane.<br>• Width – the width of the screen in integers.<br>• Height – the height of the screen in pixels.<br>• Type – shows the type of screen, for example, user, table of contents, login, and so on.<br><br>**Note:** The X, Y, Width, and Height values must be positive integers. |
| Connection | • Target screen – the screen where the connection ends. You can select any screen from the drop-down list.<br>• Control – select the control that is using the connection on the Screen Design page.<br>Click Go To to open the design page and set the screen design to the source of this connection.<br>• Close source screen on connect – select so that the source screen closes after connecting to the target screen. |

| Object | Properties/Description |
|---|---|
| Alert | • Target screen – the screen where the alert ends. You can select any screen from the drop-down list.<br>If the target and the source are the same screen, the alert pops up and returns to the same screen.<br>If the target is a different screen from the source, the alert pops up and goes to another screen.<br>• Message – enter the message you want to display with the alert.<br>• Control – select the actual control using this connection on the Screen Design page.<br>• Go to – click to open the design page, set the screen design to the source of this connection, and select the control specified in the Control drop-down list.<br>• OK button label – enter the text for the button label for information, error, and question alerts. The default value is OK.<br>• Cancel button label – enter the text for the Cancel button label for question alerts. The default value is Cancel.<br>• Close source screen on connect – select so that the source screen closes after connecting to the target screen. |

These properties are shown in the Flow Design when none of the objects on the canvas are selected.

*Device*

| Property | Description |
|---|---|
| Platform device | Select the platform from the first drop-down list and the device from the second drop-down list. The Platform determines the features available in the Device Application Designer, and the Device determines the Screen Design look and feel. |

| Property | Description |
|---|---|
| Device ribbon name | The name of the mobile application ribbon that appears on the device home screen. |
| Device ribbon image | The icon you select to appear on the device home screen. Clicking on the icon starts the application with which it is associated. |
| Device database page size | When the device database reads or writes data it works with fixed blocks of data called pages. When the page size is increased, the size of the block of data that is read or written in one operation is increased, which reduces the need to access flash memory. The default page size is 4k, which is appropriate in most cases. **Note:** If there is data that is more than 4k in a single row on the device, you must increase the page size to accommodate the data. The UltraLite database supports a page size of up to 32k. For this setting to work correctly, you must choose a database page size that is supported by both the underlying database and the device itself. Otherwise, you may get file I/O failed exception and the application exits. |
| Device database cache size | When the device database reads or writes data it works with fixed blocks of data called pages. When the cache size is increased, the number of pages that are cached for operations is increased, which reduces the need to access flash memory. The default cache size is 128k. **Note:** For this setting to work correctly, you must choose a database cache size that is supported by both the underlying database and the device itself. Otherwise, you may get file I/O failed exception and the application exits. |
| Show millisecond on time selection | If this option is selected, the DATETIME and TIME data types use the millisecond version on the device and in the Device Application Designer. |

The **platform-specific** tab displays only the properties that are platform-specific. The property values displayed in the Platform-specific tab can be read-only or configurable.

*Synchronization*

| Property | Description |
|---|---|
| Synchronize all synchronization groups on first connection | Synchronize all synchronization groups with their mobile business objects in the device application to the device when starting the client application for the first time. |
| Synchronize all synchronization groups on start-up if older than | Synchronize all the synchronization groups in the device application if the synchronization groups have not been synchronized for the defined time period.<br><br>Input the time period number in the editbox, and select the time period type (minute, hour or day) from the drop-down list. |
| Synchronize synchronization groups if pending operations are greater than or equal to | Synchronize the synchronization groups if the number of pending operations equals or exceeds the number selected.<br><br>From the drop-down list, select the maximum number of pending operations before synchronization of the synchronization groups should occur. |
| Synchronize synchronization group immediately on delete operations | When a user performs a delete, immediately synchronize the synchronization group. |
| Synchronize progress messages | The message that appears with the progress bar during synchronization. Select the state type from the drop-down list, and type the message in the editbox. |

*Localization*
The table displays the default locale and other available locales. This table describes the actions you can perform by clicking any of the buttons on the right.

| Button | Description |
|---|---|
| New | Create a new locale. |
| Update | Update the current locale. |
| Load | Load the selected locale. |
| Validate | Validate the selected locale. |

| Button | Description |
|--------|-------------|
| Edit | Edit the selected locale. |
| Remove | Remove the selected locale. |

### Flow Design Palette Options

Use the Flow Design Palette options to configure the screens, alerts, and connections in your device application.

The items that appear on the Flow Design Palette depend on the capabilities of the selected device. Only items supported by the selected device platform display in the Palette. For example, if you selected Windows Mobile as the device platform, the BlackBerry Custom Code screen does not appear in the Flow Design Palette.

*Screens*

| Option | Description |
|--------|-------------|
| Empty | Creates an empty screen canvas that you can use to design device applications, using the Screen Design page. |
| Table of Contents | Automatically creates "Go To" connections to screens on the Flow Design window when you add it to the Flow Design. It then creates screen design widgets, which allow the user to click something to go to the other screens. |

*Stock Screens*

You can customize the controls on these provided default (stock) screens.

| Screen | Description |
|--------|-------------|
| Login | A stock screen used to request user name and password information from the user. |
| Logs | Shows synchronization activity and client application activities, for example, the server log and client logs. |
| Pending Operations | Lists all pending operations that are not yet executed against a mobile business object. |
| Personalization | Lists all the personalization keys in all packages of MBO references for the current application. You can assign values to these keys. |
| Search | Defines search criteria for tables. |

| Screen | Description |
|---|---|
| Synchronize | Selects a synchronization group to synchronize. The referenced mobile business objects for each available synchronization group display under the synchronization group. When a synchronization group is synchronized, all of the mobile business objects referenced by the synchronization group are synchronized. |
| BlackBerry Custom Code | Creates a custom-coded BlackBerry screen. |
| Windows Mobile Custom Code | Creates a custom-coded Windows Mobile screen. |

*Settings Screens*
Only settings screens with settings supported by the selected profile are shown in the palette. Settings screens are not connected to other screens in the Flow Design.

| Screen | Description |
|---|---|
| About | Design your own "About screen," that can contain information like version, copyright, author and so on. |
| Logging | Specifies the default log level, for example, ERROR, WARN, INFO, and so on, that the client can use for the client side log level definition. In addition, it specifies the label for the default log level. |
| Profile | Lists and defines connection profiles to the data sources.<br><br>**Note:** Not for use with message based applications. |
| Push Settings | Configures push HTTP notifications.<br><br>**Note:** Not for use with message based applications. |
| Screen Saver | Displays a screen saver. The screen saver requires a login screen since it locks out with a password, based on a user's inactivity.<br><br>**Note:** Not supported on Windows Mobile. |

*Alerts*

| Option | Description |
|--------|-------------|
| Error | Defines this alert to pop up an error alert dialog, in which the user can click only **OK**. |
| Information | Defines this to pop up a dialog box that contains an informational message. |
| Question | Displays a biderectional alert with user response support. If the user cancels on the question, they are always returned to the original screen. If a user clicks **OK**, they go to a defined target screen. |

*Connections*

| Option | Description |
|--------|-------------|
| Cancel | Defines this connection for a screen that the user automatically goes to when he or she clicks **Cancel**. |
| Error | Defines this connection for a screen that the user automatically goes to when an error occurs. |
| Go To | Defines this connection for a screen that the user automatically goes to when he or she clicks **OK** or **Next**. |
| Pending | Creates a connection for pending synchronizations. |
| Success | Creates a connection to a screen that goes to another screen upon the success of an operation. |

### Using Mobile Business Objects As a Data Source

Use mobile business objects (MBO) from a mobile application project as the data source for the device application.

When using mobile business objects as a data source for the device application, keep in mind:

- During design time, you can use mobile business objects that have not been deployed to Unwired Server, but before deploying the application for testing on a device or simulator, you must deploy the mobile business objects referenced in the device application, to the Unwired Server.
- All the mobile business objects in the deployed package must be used in the device application. If a package contains mobile business objects you do not want to use, you can

unselect them in the Contents page of the Deploy Mobile Application Project wizard before deploying the package.

- To parse empty BINARY (0x) to the synchronization parameter from the client side and get the records with empty BINARY, you can use one of these methods:
  - In the mobile business object, set the default value of the synchronization parameter to empty BINARY. Then when you configure the initial synchronization parameter in the Device Application Designer, the server default (literal empty) is used, so records with empty BINARY can be synchronized.
  - Define a BINARY Personalization Key with an empty default value, and configure the initial synchronization parameter in the Device Application Designer to use the Personalization Key as the input value.
- In a case where there are multiple parents for the same child MBO, and one parent MBO has a relationship that is one-to-one, and another parent MBO has a relationship that is one-to-many, duplicate screens for the child detail and its create, update, delete operations are generated. This does not happen if both parents have a one-to-many relationship defined.
- On Windows Mobile devices, you cannot use mobile business objects from two different mobile application projects in one Device Application Designer.
- Mobile business objects that use message-based synchronization are not supported on BlackBerry devices.

### *Using Drag and Drop to Add Mobile Business Objects to the Flow Design*
Drag and drop mobile business objects onto the Flow Design canvas to create screens that are populated with data from the data source.

1. In the Device Application Designer, open the **Flow Design**.
2. In WorkSpace Navigator, locate the mobile business object for which to create a screen, select the mobile business object and drag and drop it onto the **Flow Design** page. Screens and connections are created automatically if the **Automatic Creation** preference in the **Preferences > Device Application Designer** dialog is selected, which, by default, it is.
3. You can manually create screens by clicking the mobile business object icon in the upper-right corner of the initial screen to see the options for creating additional screens. Double-click your selection to create the screen.
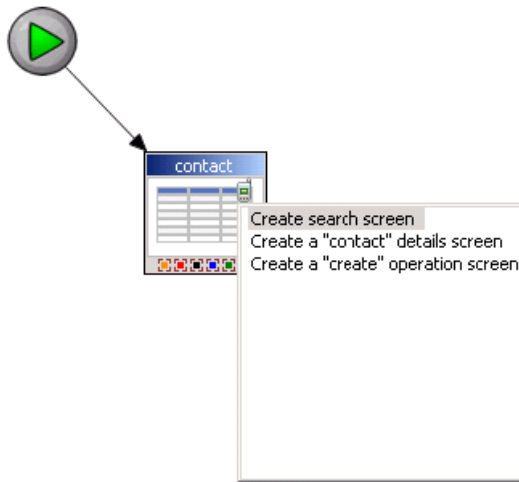
The screen is added to the Flow Design canvas.

### *Automatic Screen Creation*
Automatically create screens that are populated with data from the data source by dragging and dropping a mobile business object onto the Flow Design.

### *Setting the preferences for automatic screen creation*
Set the preferences for automatic screen creation in the Device Application Designer Preferences. These preferences allow you to decide which screens to create when you drop a mobile business object onto the Flow Design canvas.

The Relationship Automatic Screens option means that all sub screens are created after the initial relationship screen is created.

By default, Delete Operation Screen is unselected, which means the Device Application Designer:

- Creates the main screen with a table for the mobile business object.
- Adds a Delete menu item to the main screen.
- Adds an alert action of the question type, which goes back to the main screen to the menu item. Adding this action means the user is prompted to confirm before deleting.
- Adds a save context action to the menu item.
- Adds an operation action that performs the delete operation to the menu item.

If Delete Operation Screen is selected, a Delete screen is created.

### *Behavior of automatic screen creation*
When using the auto creation feature, keep these points in mind:

If you drop more than one mobile business object onto the Flow Design canvas, only the main screens are created. If a screen has multiple related screens that can be created, use the assist

option  at the top of the screen to create the screens.

When using MBOs with multiple relationships, if different behavior is expected for the different types of relationships, you should manually create different screens to accommodate the different types of relationships instead of using the same screen for all relationships.

### Structure Mobile Business Objects

You can use the Device Application Designer to create device applications that use structure mobile business objects as the data source.

When a mobile business object (MBO) is dropped from Workspace Navigator to the Flow Design page, the screens and their relationships are created automatically. The screens consist of the Cell listing screen, a Detail screen, and Operation forms screens, which include the create, update, and delete operations. Any existing relationship creates more screens by repeating the process on the relationship MBO.

You should not drag and drop a structure MBO from WorkSpace Navigator because it will be treated like an unstructured MBO, so any references from this MBO to other structures (attribute types with an array type of "structure []") are returned with a relationship, which means that these "sub-structures" will also create a full set of screens. You can drag and drop an unstructured MBO that references a structure MBO onto the Flow Design page. This will ensure that the screens will be created correctly.

### Flattened structure

MBO operations that reference a structure in the operations (without the type being an array "[]") are flattened as single parameters in the Device Application Designer.

### MBO array structure

If the MBO structure reference is an array of structures, a screen set is generated that allows you to see the list of the structures and handle the create, update, and delete operations, as well as the details of the structure.

An operation that references a structure array type which in turn references another structure array, is not supported in the Device Application Designer.

### Personalization keys

When a personalization key is assigned as a single value inside the structure, the key is mapped as a normal personalization key in the operation dialog for a flattened parameter. For flattened items, no parameters are generated in the operations, so a form without these parameters is generated and the operation dialog shows the personalization key assigned, but the user cannot change the configuration.

*Data Persistence on the Client Side When Using a Structured Mobile Business Object*

Use custom coding so that structured mobile business object (MBO) data persists on the BlackBerry device.

By default, data does not persist to the client side when you use an MBO of the structured type because the data resides in the RAM memory. This means that every time you exit the device application or restart the device, the data is lost. Data persistence is useful when you want to preserve the state of data, for example, if you have to exit the mobile application while inputting a long list of data, you would want the data to persist on the file system so that you can continue inputting the data when you restart the application or device.

You can achieve data persistence on the client side by creating a custom coded subclass that implements data Persistence.

The following example uses a local MBO (LocalMyObjectWithSimpletype) to persist the structured MBO data, but you can use a database, file, or something else. The workflow is similar no matter what method you choose to use.

The basic flow for creating data persistence for a structured MBO is:

1. Create the local MBO (LocalMyObjectWithSimpletype). The mobile application project used in this example contains the GetMyObject MBO with a single operation, **insertMyObjectWithSimpleType**. Its parameter is a structured type— **MyObjectWithSimpleType**. So on the device side, to call the operation, a list of MyObjectWithSimpleType type instance needs to be passed as the input parameter. The list is the data that needs to be persisted.
2. Create a new device application in the Device Application Designer and drag and drop the GetMyObject MBO onto the Flow Design page.
3. Use the Mobile Business Object References dialog to add the local MBO (LocalMyObjectWithSimpletype) as a Mobile Business Object Reference for the GetMyObject MBO.
4. In the Device Application Designer preferences for **Device > BlackBerry**, select the **Generate custom subclass for all** option. You must restart Sybase Unwired Workspace for this to take effect.
   Another option, since you are customizing only the BOBCUIDefinition, is go to the Device Application Designer properties view, select the **Device** tab, and set the "Generate a custom coding subclass" platform property to true. This way, a BOBCUIDefinitionCustom.java file is generated as a subclass for the custom coding.
5. Use the Generate Device Application wizard to generate the device application code.
6. Locate the generated custom subclass in the generated `custom` folder, and write the SPMBOModel subclass that provides the persist implementation, as shown in this example:

```
class CustomSPMBOModel extends SPMBOModel
{
    public CustomSPMBOModel(String id)
```

```
    {
        super(id);
    }

    /**
     * Override to provide persistent function, in this case,
     * we use client only mbo to persist the data.
     * User can also use their own persist mechanism.
     * @return <description>
     */
    protected Vector readPersistentData()
    {
        ObjectList persistData =
                    LocalMyObjectWithSimpletype.findAll();
        if(persistData!=null&&persistData.size()>0)
        {
            ObjectList data = new ObjectList();
            int size = persistData.size();
            for(int i = 0; i<size; i++)
            {
                LocalMyObjectWithSimpletype persistObj =
                (LocalMyObjectWithSimpletype)persistData.
                                            elementAt(i);
                MyObjectWithSimpletype obj =
                                new MyObjectWithSimpletype();
                data.addElement(obj);
                obj.setBooleanType(persistObj.getBooleanType());
                obj.setDecimalType(persistObj.getDecimalType());
                obj.setDoubleType(persistObj.getDoubleType());
                obj.setFloatType(persistObj.getFloatType());
                obj.setIntType(persistObj.getIntType());
                obj.setStringType(persistObj.getStringType());
            }
            return data;
        }
        return null;
    }

    /**
     * Persist the structure mbo data to a local MBO
     * @return <description>
     */
    protected void doPersist(Object mbo, String operationName)
    {
        if(operationName.equals(CREATE_OP))
        {
            MyObjectWithSimpletype rawObj =
                            (MyObjectWithSimpletype)mbo;
            LocalMyObjectWithSimpletype persistObj =
                            new LocalMyObjectWithSimpletype();
            persistObj.setBooleanType(rawObj.getBooleanType());
            persistObj.setDecimalType(rawObj.getDecimalType());
            persistObj.setDoubleType(rawObj.getDoubleType());
            persistObj.setFloatType(rawObj.getFloatType());
            persistObj.setIntType(rawObj.getIntType());
            persistObj.setStringType(rawObj.getStringType());
```

```
                persistObj.create();
        }
        else if(operationName.equals(DELETE_OP))
        {
            MyObjectWithSimpletype rawObj =
                            (MyObjectWithSimpletype)mbo;
            LocalMyObjectWithSimpletype persistObj =
                            findPersistObj(rawObj);
            if(persistObj!=null)
            {
              ((LocalMyObjectWithSimpletype)persistObj).delete();
            }
        }
        else if(operationName.equals(UPDATE_OP))
        {
            MyObjectWithSimpletype rawObj =
                            (MyObjectWithSimpletype)mbo;
            LocalMyObjectWithSimpletype persistObj =
                            findPersistObj(rawObj);
            if(persistObj!=null)
            {
              persistObj.setBooleanType(rawObj.getBooleanType());
              persistObj.setDecimalType(rawObj.getDecimalType());
                persistObj.setDoubleType(rawObj.getDoubleType());
                persistObj.setFloatType(rawObj.getFloatType());
                persistObj.setIntType(rawObj.getIntType());
                persistObj.setStringType(rawObj.getStringType());
                persistObj.update();
            }
        }
    }

    private LocalMyObjectWithSimpletype findPersistObj
    (MyObjectWithSimpletype rawObj)
    {
        Query q = new Query();
        CompositeTest ct = new CompositeTest();
        ct.setCompositionType(CompositeTest.AND);
        ct.add(
            new AttributeTest(
                "IntType",
                rawObj.getIntType().toString(),
                AttributeTest.EQUAL));
        ct.add(
            new AttributeTest(
                "BooleanType",
                rawObj.getBooleanType().toString(),
                AttributeTest.EQUAL));
        ct.add(
            new AttributeTest(
                "DecimalType",
                rawObj.getDecimalType().toString(),
                AttributeTest.EQUAL));
        ct.add(
            new AttributeTest(
              "DoubleType",
```

```
        rawObj.getDoubleType().toString(),
        AttributeTest.EQUAL));
  ct.add(
    new AttributeTest(
      "FloatType",
      rawObj.getFloatType().toString(),
      AttributeTest.EQUAL));
  ct.add(
    new AttributeTest(
      "StringType",
      rawObj.getStringType().toString(),
      AttributeTest.EQUAL));
  q.setTestCriteria(ct);

  ObjectList result =
        LocalMyObjectWithSimpletype.findWithQuery(q);
  if(result!=null&&result.size()>0)
  {
      return (LocalMyObjectWithSimpletype)result.
                                    elementAt(0);
  }
  return null;
  }
}
```

**7.** Register the subclass to the `MBOModelFactory` to replace the default SPMBOModel. You can do this by overriding the super **init()** function in the `BOBCUIDefinitionCustom` class, as shown in this example:

```
public class BOBCUIDefinitionCustom extends BOBCUIDefinition
{
    // customize BOBCUIDefinition

    public void init()
    {
        super.init();
        MBOModelFactory.getInstance().regMBOModel(
            MBO_PERSISTSAMPLE_MYOBJECTWITHSIMPLETYPE,
            new CustomSPMBOModel(
                MBO_PERSISTSAMPLE_MYOBJECTWITHSIMPLETYPE));
    }
}
```

The constant `MBO_PERSISTSAMPLE_MYOBJECTWITHSIMPLETYPE` is the MBO ID defined for MyObjectWithSimpleType, and you can find it in the super class `BOBCUIDefinition`.

**8.** Regenerate the device application and launch the application.

*Complex Parameter Support*

Relationships are generated when dragging and dropping a mobile business object (MBO) with an operation that has a complex parameter (the parameter points to another MBO).

The relationship to the other MBO is generated from the MBO table. If the target MBO has no attributes other than structured MBOs, no relationship is generated.

For example, in a case where you have three mobile business objects, where MBO A —>MBO B —>MBO C, when MBO A is dropped onto the Flow Design page, the Create/Read/Update/ Delete (CRUD) operation is automatically generated. However, since MBO B has attributes that point only to MBO C, and no other attributes, there is no table data for MBO B, therefore, no CRUD table and forms to generate. If you want references from MBO A to MBO C, move the reference of MBO C directly into MBO A.

When generating the form for the operation, if a table exists, a Configure button and a link to the table are created automatically. The Configure button takes you to the MBO table (with its forms) so you can edit the complex MBO.

**Note:** There is no object query support for complex parameters.

*Local Mobile Business Objects*

Use a local mobile business object (MBO), which is not bound to a data source and cannot be synchronized with the consolidated database (CDB), but performs business functions locally on the device application.

Keep in mind the behavior and limitations of local MBOs used in device application development:
- Local MBOs must be deployed to Unwired Server. However, they are not available to Unwired Server.
- Operation status does not show on the device for local MBOs.

## Adding Screens from the Palette to the Flow Design

Use the Flow Design page to add and connect screens you design, as well as customized stock screens.
1. In the Device Application Designer, click **Flow Design**.
2. In the Flow Design page, select a screen from the **Palette > Screens** menu and drag and drop it onto the Flow Design canvas.
   The Start icon on the Flow Design window is automatically attached to the first screen you drag and drop onto the Flow Design window. You can change the target screen.

*Screens*

The Screens folder on the Flow Design Palette menu shows the default empty screen and any other screens configured in the Device Application Designer plug-in Screens folder.

Sybase Unwired Platform provides some default, or stock, screens that have controls on them that you can customize. The default screens consist of some of the most commonly used

screens, such as a login screen, a screen saver screen, an about screen and so on, so that you do not have to build from scratch every time.

**Note:** The Search screen is for use with grid and cell tables. You must configure the Action property for the Search screen to specify the screen to which to navigate upon success.

You can save existing screen designs as templates so you do not have to make the same changes again and again. These saved templates are displayed in the Palette.

### Connections

Use connections to connect screens with actions in the Flow Design.

Connection actions describe the navigation direction the screen should go after the user performs an action, cancels an action, or uses a control. Alert actions use actions defined in the Flow Design.

**Note:** You can use only one connection of each type from a given source screen, except for the GoTo connection type. You can have as many GoTo connection types as you want to on a source screen, but there cannot be more than one GoTo connection to the same target screen.

### Adding Connections to Screens

The Start icon is displayed on the Flow Design page, and its connection is attached to the first screen you add to the Flow Design. You can change the target to any screen in the Flow Design.

1. In the Flow Design page, select a connection type from **Palette > Connections**.
2. In the Flow Design page, click on the source screen and drag to connect it to the target screen.

   **Note:** Connections do not work until they are used in an action attached to a menu, button, or other control. Right-click on the connection to create a menu and add an action with this connection.

   The connection line displays between the two screens.

### Updating the Table of Contents Screen

Update the Table of Contents screen to automatically create Go To connections for new screens.

When you use the Table of Contents screen, a list item is created for each screen in the flow design. The name of the screen is used as the name of the list item. You can set a preference in the Device Application Designer preferences for the Table of Contents to use the screen's icon as well as the screen name for the list item. Each of the list items has an action assigned of the type of connection that goes to the screen for that index.

1. In the Device Application Designer Flow Design page, right-click on the Table of Contents Screen.

2. Select Update Table of Contents.
   A Go To connection is created for any screen that does not currently have one. Connections are not created for these screens:
   • Login
   • Search
   • Relationship target screens
   • Detail screens that are details on a table screen
   • Any screen with mobile business object table context variables

*Alerts*

An alert is designed to notify you of a problem or to ask a question.

All alerts can be directional, meaning upon closing the alert, the user can go to a new screen, or be returned to the original screen.

You can have as many alerts on a screen as you want, however, you cannot have more than one of the same type of alert from the source to the target. For instance, you cannot have two error alerts from a source screen to a target screen, but you could have an error alert and a question alert from the source screen to the target screen.

Alerts are shown as connection arrows with the alert type in the middle of the connection line.

## Designing the Screen for Device Applications

Use the Screen Design page of the Device Application Designer to design the screen for your device application.

1. In the Flow Design page, select a screen from the Palette and drag and drop it onto the canvas.
   You can select from the stock screens, or select the Empty screen to design yourself.
2. Double-click the screen.
   The Screen Design page opens.
3. Select the device for which you want to design the screen.
4. Drag and drop items from the Screen Design Palette onto the screen.
5. (Optional) Assign actions to the controls.
6. (Optional) Create a mobile business object reference.
7. Select **File > Save**.

### Screen Design

When designing screens for mobile devices, keep the following in mind:

• For touch screen devices, verify that the application displays correctly in both landscape and portrait views.

- Develop the user interface so it is consistent with existing applications for the specific platform so you can leverage users' existing knowledge for how to navigate applications on that particular platform. For example, let users open context menus in applications by clicking the trackball or trackpad for BlackBerry applications.
- For BlackBerry and Windows Mobile devices, put all actions a user can perform in the full menu for the application.
- Design the user interface so users can access information quickly and display only the information that is immediately relevant to the task.
- Limit the number of controls you place on each screen.
- Organize controls or groups of controls by using line separators.
- Display the most important information at the top of the screen.
- Include a navigation bar that describes the contents of the screen.
- Think about localization requirements early in the design process as localization can affect the layout of screens.
- Prompt users with a dialog box to save any unsaved data before closing a screen.
- Allow users to easily backtrack and undo actions in case they make a mistake.

### Screen Design Page Properties

These are the property pages that display in the Properties for the Screen Design page when no objects are selected.

The default values for the properties shown are determined by the Device Application Designer preferences. Right-click in any page (except the Introduction page) of the Device Application Designer and select **Preferences**.

*Display*
The Display properties shown are for the actual display area inside the device image. The Display page has three tabs—Definition, Platform-specific, and Coding.

**Table 3. Definition**

| Property | Description |
|---|---|
| Number of columns | The number of columns in the display. Only 1-column, 2-column and 3-column are supported. The default is 2-column. |
| Assign percentages | Select to set the percentages of the spinner controls. There is a spinner control for each column. The numbers restrict the size to a minimum of 10 and a maximum of 90 on a 2-column display and 80 on a 3-column display. The percent values are always set to equal 100% so that the columns in the display are aligned by these percentages. |

| Property | Description |
| --- | --- |
| Horizontal alignment | Pick the alignment default for all objects in the screen display. This aligns the objects to the point in the column corresponding to the choice. |
| | **Note:** Some controls ignore alignment ; for example, edit box, list item, toolbar item, image, list detail, choice, spacer, separator, and table because they occupy the entire column regardless of the display layout. |
| | The other controls allow for customizing the alignment for the specific widget regardless of the display layout. |
| Show pattern | Select to show a pattern in the background. |
| Background image | Click **Browse** to choose an image to display in the background. |
| Scale image to fill display size | Select to scale the image to fill the display size. If this is not selected, the image fits the fixed aspect. |
| Style | Select the style of the display. Click **Search** to search available styles. |
| | Click **Edit** to edit the selected style. |
| | Click **Restore** to restore the default style. |

The **Platform-specific** tab displays only the properties that are platform-specific. The property values displayed in the Platform-specific tab can be read-only or configurable.

*Coding*

The Coding tab is displayed for controls and containers that support events. Events allow you to specify custom events on screens and screen controls during design time. For example, a table of e-mails can display records differently (maybe with an icon for unread mail) depending on the state of the e-mail.

| Property | Description |
| --- | --- |
| ID | The internal ID of the control. This is a read-only field for informational purposes. Not all controls have IDs. This is left blank for controls that do not have an ID. |

| Property | Description |
|----------|-------------|
| Base name | An optional name for the control, for example, SubmitButton. This name can be empty, but if specified, it must be unique within a screen. This name is not localizable. Only alphanumeric (a-z, A-Z, 0-9) and underscores (_) are allowed in the name. |
| Events | Lists the available events for the selected control. The list of events varies according to the type of control selected. Select the events to customize by selecting one or more of the checkboxes. |

### *Stock or Settings Screen*

This tab is displayed in the screen properties if the selected screen is either a stock or settings screen. The properties are displayed in a table. Select a property and click **Configure** to configure individual properties.

Select a property and click **Restore** to restore the selected property to its default settings.

Select a property and click **Restore All** to restore all properties to their default settings.

If you are viewing the Properties for a BlackBerry Custom Code stock screen, you see a **Create Screen Class** button. Click this button to create a Java class for the BlackBerry custom screen.

### *Device*

| Property | Description |
|----------|-------------|
| Platform device | Select the platform from the first drop-down list and the device from the second drop-down list. The Platform determines the features available in the Device Application Designer, and the Device determines the Screen Design look and feel. |
| Device ribbon name | The name of the mobile application ribbon that appears on the device home screen. |
| Device ribbon image | The icon you select to appear on the device home screen. Clicking on the icon starts the application with which it is associated. |

| Property | Description |
|---|---|
| Device database page size | When the device database reads or writes data it works with fixed blocks of data called pages. When the page size is increased, the size of the block of data that is read or written in one operation is increased, which reduces the need to access flash memory. The default page size is 4k, which is appropriate in most cases. |
| | **Note:** If there is data that is more than 4k in a single row on the device, you must increase the page size to accommodate the data. The UltraLite database supports a page size of up to 32k. For this setting to work correctly, you must choose a database page size that is supported by both the underlying database and the device itself. Otherwise, you may get file I/O failed exception and the application exits. |
| Device database cache size | When the device database reads or writes data it works with fixed blocks of data called pages. When the cache size is increased, the number of pages that are cached for operations is increased, which reduces the need to access flash memory. The default cache size is 128k. |
| | **Note:** For this setting to work correctly, you must choose a database cache size that is supported by both the underlying database and the device itself. Otherwise, you may get file I/O failed exception and the application exits. |
| Show millisecond on time selection | If this option is selected, the DATETIME and TIME data types use the millisecond version on the device and in the Device Application Designer. |

The **platform-specific** tab displays only the properties that are platform-specific. The property values displayed in the Platform-specific tab can be read-only or configurable.

*Synchronization*

| Property | Description |
|---|---|
| Synchronize all synchronization groups on first connection | Synchronize all synchronization groups with their mobile business objects in the device application to the device when starting the client application for the first time. |
| Synchronize all synchronization groups on start-up if older than | Synchronize all the synchronization groups in the device application if the synchronization groups have not been synchronized for the defined time period.<br><br>Input the time period number in the editbox, and select the time period type (minute, hour or day) from the drop-down list. |
| Synchronize synchronization groups if pending operations are greater than or equal to | Synchronize the synchronization groups if the number of pending operations equals or exceeds the number selected.<br><br>From the drop-down list, select the maximum number of pending operations before synchronization of the synchronization groups should occur. |
| Synchronize synchronization group immediately on delete operations | When a user performs a delete, immediately synchronize the synchronization group. |
| Synchronize progress messages | The message that appears with the progress bar during synchronization. Select the state type from the drop-down list, and type the message in the editbox. |

*Mobile Business Object*

The Mobile Business Object property page displays when the selected control, such as a grid table, cell table, or list detail can be bound to a mobile business object. Use this page to associate a mobile business object to the selected control and configure the attributes to be displayed in the control. For example, you can associate a Customer List MBO to a Table control, and choose to show the First name, Last name, and Phone number of the customers. You can also associate the same MBO to another list detail control and choose to show all the attributes of a customer.

| Property | Description |
|---|---|
| Mobile business object | The MBO associated with the control. |
| | Click **Search** to browse the available mobile business objects in the mobile application projects. |
| | Click **Clear** to remove the selection and update the Attributes list. |
| | If the control is not originally associated with an MBO, the first four attributes of the new MBO are selected by default. Otherwise, a comparison of the previously selected attributes is performed against the attributes of the new MBO. If any previously selected attribute exists in the new MBO, the selection is retained. |
| Object query | Select to apply the object query to the data shown in the control. The object query drop-down shows the queries available for the selected MBO. If there are parameters in the query, the Initial Parameter Values button is enabled. |
| | Click **Initial Parameter Values** to launch the Initial Object Query Parameters configuration dialog. |
| Attributes | This lists all the attributes in the MBO. Drag and drop items up and down to reorder the attributes list. Only selected attributes can be ordered. |
| | This property does not appear if the control the MBO is bound to is a cell table because cell tables can use an attribute more than once. |
| | Click **Edit Attributes** to edit the MBO attributes. |
| | Click **Select All** to select all the attributes and **Deselect All** to the clear all the selections. |
| | **Note:** Linking a control to a mobile business object without selecting any of its attributes results in an unusable control and is reported in the Problems view when you verify the device application. |

| Property | Description |
|---|---|
| Error message | Enter the error message that the user sees when there is an error loading the mobile business object. |

### Localization

The table displays the default locale and other available locales. This table describes the actions you can perform by clicking any of the buttons on the right.

| Button | Description |
|---|---|
| New | Create a new locale. |
| Update | Update the current locale. |
| Load | Load the selected locale. |
| Validate | Validate the selected locale. |
| Edit | Edit the selected locale. |
| Remove | Remove the selected locale. |

### Validation

The Validation tab displays only when viewing the Properties page for an Edit Box control.

| Button | Description |
|---|---|
| New | Create a new rule validation item for the edit box. Newly created validation items appear in the table. You can order the validation items in the list by dragging and dropping. |
| Edit | Edit the selected validation item. |
| Remove | Remove the selected validation item. |

## Screen Design Palette Options

Drag and drop items from the palette to design the screen for a device application.

The items shown on the Screen Design palette vary depending on the selected device platform.

### Controls

Drag and drop controls, or widgets, from the Palette to the screen design.

Controls are objects on the screen that describe what to do upon execution of an action, for example, what to do in response to a mouse click, double-click, menu selection, button selection, and so on. Supported controls are determined by the selected device. Not all controls

are supported on all devices, so the controls you see on the Screen Design Palette vary according to the selected device.

Some controls do not do anything unless you explicitly assign an action to them, for example, buttons, hyperlinks, and menus.

| Option | Description | Supported device platform |
|---|---|---|
| Navigation bar | Create a header on the screen. | • BlackBerry<br>   • BlackBerry Curve 8530 and 8900<br>   • BlackBerry Pearl 8220<br>   • BlackBerry Bold 9000<br>   • BlackBerry Storm 9530<br>   • BlackBerry Storm 2 9550<br>   • BlackBerry Tour 9630<br>   • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |
| Label | Display text. | • BlackBerry<br>   • BlackBerry Curve 8530 and 8900<br>   • BlackBerry Pearl 8220<br>   • BlackBerry Bold 9000<br>   • BlackBerry Storm 9530<br>   • BlackBerry Storm 2 9550<br>   • BlackBerry Tour 9630<br>   • BlackBerry Bold 2 9700 |
| Edit box | Enter a line of text and assign values to strings, dates, decimals, integers, and so on. | • BlackBerry<br>   • BlackBerry Curve 8530 and 8900<br>   • BlackBerry Pearl 8220<br>   • BlackBerry Bold 9000<br>   • BlackBerry Storm 9530<br>   • BlackBerry Storm 2 9550<br>   • BlackBerry Tour 9630<br>   • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |

| Option | Description | Supported device platform |
|--------|-------------|---------------------------|
| Choice | Display a collection of items as a drop-down list. | • BlackBerry<br>  • BlackBerry Curve 8530 and 8900<br>  • BlackBerry Pearl 8220<br>  • BlackBerry Bold 9000<br>  • BlackBerry Storm 9530<br>  • BlackBerry Storm 2 9550<br>  • BlackBerry Tour 9630<br>  • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |
| Hyperlink | Display a label that is underlined like a link and that can be attached to an action for screen transitions. | • BlackBerry<br>  • BlackBerry Curve 8530 and 8900<br>  • BlackBerry Pearl 8220<br>  • BlackBerry Bold 9000<br>  • BlackBerry Storm 9530<br>  • BlackBerry Storm 2 9550<br>  • BlackBerry Tour 9630<br>  • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |
| Checkbox | Assign or display boolean values. | • BlackBerry<br>  • BlackBerry Curve 8530 and 8900<br>  • BlackBerry Pearl 8220<br>  • BlackBerry Bold 9000<br>  • BlackBerry Storm 9530<br>  • BlackBerry Storm 2 9550<br>  • BlackBerry Tour 9630<br>  • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |

| Option | Description | Supported device platform |
|--------|-------------|---------------------------|
| Radio button | Assign predefined options. | • BlackBerry<br>   • BlackBerry Curve 8530 and 8900<br>   • BlackBerry Pearl 8220<br>   • BlackBerry Bold 9000<br>   • BlackBerry Storm 9530<br>   • BlackBerry Storm 2 9550<br>   • BlackBerry Tour 9630<br>   • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |
| List item | Similar to buttons, list items are grouped in the List Group container and can have styles and actions assigned to them. | • BlackBerry<br>   • BlackBerry Curve 8530 and 8900<br>   • BlackBerry Pearl 8220<br>   • BlackBerry Bold 9000<br>   • BlackBerry Storm 9530<br>   • BlackBerry Storm 2 9550<br>   • BlackBerry Tour 9630<br>   • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |
| Button | Trigger actions such as screen transitions. | • BlackBerry<br>   • BlackBerry Curve 8530 and 8900<br>   • BlackBerry Pearl 8220<br>   • BlackBerry Bold 9000<br>   • BlackBerry Storm 9530<br>   • BlackBerry Storm 2 9550<br>   • BlackBerry Tour 9630<br>   • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |

| Option | Description | Supported device platform |
|---|---|---|
| Image | You can place an image into the display as widgets (button, list item, and so on). There are two types of images—one type uses the image references to display the image, the other type uses table context variables that are of image types. You can also assign actions to images. | • BlackBerry<br> • BlackBerry Curve 8530 and 8900<br> • BlackBerry Pearl 8220<br> • BlackBerry Bold 9000<br> • BlackBerry Storm 9530<br> • BlackBerry Storm 2 9550<br> • BlackBerry Tour 9630<br> • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |
| Spacer | Create a defined space on the screen. This allows for vertical spanning as well as horizontal spanning. | • BlackBerry<br> • BlackBerry Curve 8530 and 8900<br> • BlackBerry Pearl 8220<br> • BlackBerry Bold 9000<br> • BlackBerry Storm 9530<br> • BlackBerry Storm 2 9550<br> • BlackBerry Tour 9630<br> • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |
| Separator | Visually isolate parts of the screen. | • BlackBerry<br> • BlackBerry Curve 8530 and 8900<br> • BlackBerry Pearl 8220<br> • BlackBerry Bold 9000<br> • BlackBerry Storm 9530<br> • BlackBerry Storm 2 9550<br> • BlackBerry Tour 9630<br> • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |

| Option | Description | Supported device platform |
|---|---|---|
| Grid table | Add to the screen display or tab panel container. Used to display multiple data columns. | • BlackBerry<br>   • BlackBerry Curve 8530 and 8900<br>   • BlackBerry Pearl 8220<br>   • BlackBerry Bold 9000<br>   • BlackBerry Storm 9530<br>   • BlackBerry Storm 2 9550<br>   • BlackBerry Tour 9630<br>   • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |
| List detail | Show property details for columns. | • BlackBerry<br>   • BlackBerry Curve 8530 and 8900<br>   • BlackBerry Pearl 8220<br>   • BlackBerry Bold 9000<br>   • BlackBerry Storm 9530<br>   • BlackBerry Storm 2 9550<br>   • BlackBerry Tour 9630<br>   • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |
| Toolbar item | Add toolbar items to the toolbar container. Actions can be assigned to toolbar items. An image must be assigned to the toolbar item control. If you do not assign an image to toolbar items, a warning message is shown when you verify or generate the device application code. | • BlackBerry Storm 9530<br>• BlackBerry Storm 2 9550<br>• Windows Mobile Professional |

_Cells_

Drag and drop the cell table onto the screen. Cell labels and cell images can be dropped onto the cell table.

Cell tables show data records in cells as entities. Each record has little or no relation to other records, for example, an e-mail, contact information, and so on.

| Option | Description |
|--------|-------------|
| Cell table | Add to the screen display or tab panel container. Each cell in the table can contain data from multiple columns. Styles are assigned to all cells so they look uniform, but cells can use multiple styles. Each record spans the full width of the device screen and contains one or more lines. |
| Cell label | A text item inside a cell. It can be a hard-coded literal value, a value from a variable, a dynamic value from a mobile business object attribute, or a mix of all of these. |
| Cell image | An image item in a cell. The image can be a status indicator, and it can come from a mobile business object attribute, a table context variable, a value mapping, or an image reference. Depending on the type of image assigned, the cell image shows up differently in the editor and in the outline view. |

*Containers*

Drag and drop containers from the Palette to the screen design. A container control is used to group child controls together.

| Option | Description | Supported device platform |
|--------|-------------|---------------------------|
| Region | Only Display or Tab controls can contain a region container, which is primarily used to group controls that do not have their own containers.<br><br>When the **Show border** attribute is unchecked, The Region control is represented on the Screen Design page by a rectangle with dotted lines. It is not visible on the device client; it is transparent and shows the background color of its parent (Display or Tab).<br><br>All widgets inside the region inherit the alignment property of the Display. Child widgets can override the Display alignment if you set the individual value for the child. | • BlackBerry<br>  • BlackBerry Curve 8530 and 8900<br>  • BlackBerry Pearl 8220<br>  • BlackBerry Bold 9000<br>  • BlackBerry Storm 9530<br>  • BlackBerry Storm 2 9550<br>  • BlackBerry Tour 9630<br>  • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |
| Radio group | A region in which to group radio buttons. The radio group cannot be inside a region container. The radio group contains only radio buttons, and radio buttons can be added only to radio groups.<br><br>The radio group appears as a control to be added to a persist or submit action. | • BlackBerry<br>  • BlackBerry Curve 8530 and 8900<br>  • BlackBerry Pearl 8220<br>  • BlackBerry Bold 9000<br>  • BlackBerry Storm 9530<br>  • BlackBerry Storm 2 9550<br>  • BlackBerry Tour 9630<br>  • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |

| Option | Description | Supported device platform |
|---|---|---|
| List group | A container for list items. | • BlackBerry<br>   • BlackBerry Curve 8530 and 8900<br>   • BlackBerry Pearl 8220<br>   • BlackBerry Bold 9000<br>   • BlackBerry Storm 9530<br>   • BlackBerry Storm 2 9550<br>   • BlackBerry Tour 9630<br>   • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |
| Tab folder | A container for tab panels. | • BlackBerry<br>   • BlackBerry Curve 8530 and 8900<br>   • BlackBerry Pearl 8220<br>   • BlackBerry Bold 9000<br>   • BlackBerry Storm 9530<br>   • BlackBerry Storm 2 9550<br>   • BlackBerry Tour 9630<br>   • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |
| Tab panel | Used to organize information in a tabular format. For example, you can break a long form into sections represented by tab panels. Table and List Detail controls that are related to each other can also be grouped with tab panels. | • BlackBerry<br>   • BlackBerry Curve 8530 and 8900<br>   • BlackBerry Pearl 8220<br>   • BlackBerry Bold 9000<br>   • BlackBerry Storm 9530<br>   • BlackBerry Storm 2 9550<br>   • BlackBerry Tour 9630<br>   • BlackBerry Bold 2 9700<br>• Windows Mobile Professional |
| Toolbar | The toolbar displays at the bottom of the device screen and occupies the full width of the display area. The toolbar contains toolbar items. | • BlackBerry Storm 9530<br>• BlackBerry Storm 2 9550<br>• Windows Mobile Professional |

### *Menu and Table Context*

Drag and drop menu and table context items onto the Table Context or Menu.

| Option | Description |
|---|---|
| Context item | Add to the Table Context section to assign an action to a grid table or cell table. The context item has a type attribute that defines the nature of the context item. For example, it can be a connection to an update operation screen, a connection to a delete operation, a connection to a details screen, a connection to a relationship screen, a connection to a create screen, or none of the above. |
| Menu item | Performs the assigned action, for example refresh or synchronize. |
| Menu separator | Drag and drop menu separators from the Palette onto the screen menu to visually organize the menu using horizontal lines. You can use multiple menu separators, which you can move and delete. Menu separators are not tied to actions. |

### *Actions*

Assign actions to controls, menu items, and context items that are meant to perform an action.

You can assign actions to hyperlinks, buttons, list items, image widgets, toolbar items, context items, and menu items. Some actions support only one instance of the action, while others support multiple instances of the action. For example, for actions assigned to a menu item, you can add only one "exit" action, while there can be any number of "operation" actions.

Controls that support actions accept the drag and drop of an action onto the control. Additionally, the Properties page for controls to which you can assign actions displays a subsection called **Actions**. An item that is assigned an action will have the action decorator attached to its display ▩. You can also drag a mobile business object, or operations, from the Workspace Navigator onto any control that supports an action, and it will create the Synchronize action for the mobile business object and Operation action for operations.

The action items you see on the Screen Design palette depends on the device platform you have selected. Not all actions are available for all platforms.

Choose from these actions:

| Action | Description | Platform |
|---|---|---|
| Alert | Indicates that a predefined event has occurred, or that an error has occurred.<br><br>**Note:** The "question" alert is unique in that the user has a choice. Any actions listed after the question are executed as a Yes/OK answer while No/Cancel stops all actions from executing. | • BlackBerry<br>• Windows Mobile Professional |
| Connection | Navigates to another screen.<br><br>**Note:** If there are multiple actions assigned to one control or menu, the connection action must be the last action performed. | • BlackBerry<br>• Windows Mobile Professional |
| Navigate back | The back action is a simplified connection action. It directs the user back to the parent screen. If a navigate back action is applied to the top screen, the action is ignored on most platforms. The navigate back action allows an update form to be used from both the table screen and the list details screen. This also allows relationships screens to be referenced from both the table and list details controls. | • BlackBerry<br>• Windows Mobile Professional |
| Exit | Directs the client application to quit.<br><br>**Note:** There can be only one Exit action on the action list, and it must be the last action on the list. | • BlackBerry<br>• Windows Mobile Professional |
| Logout | Logs the user out of the client.<br><br>**Note:** There can be only one Logout action on the action list, and it must be the last action on the list. | • BlackBerry<br>• Windows Mobile Professional |
| Operation | Takes the controls on the screen and submits them to a mobile application for insert, update, and delete actions. | • BlackBerry<br>• Windows Mobile Professional |
| Persist | Works with user variables created in the Variable section. Allows you to save values from the screen controls or literals to set values on variables created in the Variable section when the action is performed. | • BlackBerry<br>• Windows Mobile Professional |

| Action | Description | Platform |
|---|---|---|
| Refresh | Adds a refresh action to the list of actions, which refreshes the current screen based on updated data. You may want to refresh, for example, after you have synchronized a mobile business object. | • BlackBerry<br>• Windows Mobile Professional |
| Synchronize | Performs synchronization on the selected mobile application. The synchronization action allows you to filter data by setting the synchronization parameters for the selected mobile business object. | • BlackBerry<br>• Windows Mobile Professional |
| Object Query | Allows you to change the values of an object query dynamically by assigning controls, literals, variables, null, and so on to the object query parameters. This also allows you to change the object query assigned to the table or list detail. The action is supported on table and list detail controls. | • BlackBerry<br>• Windows Mobile Professional |
| Tab activation | Brings a specified tab to the forefront. There are three modes for the tab activation action:<br>• Specific<br>• Previous<br>• Next<br>The tab activation action can wrap, which means if the current tab is the first tab, clicking the Previous tab goes to the last tab.<br>When the current tab is the last tab, clicking the Next tab goes to the first tab. | • BlackBerry<br>• Windows Mobile Professional |
| Save context | Allows the user to maintain context among screens when switching between screens.<br>**Note:** The screen you add the Save Context action to must have at least one table or one list detail control. | • BlackBerry<br>• Windows Mobile Professional |
| BlackBerry PIM | Add this action to integrate BlackBerry PIM applications such as calendar, address book, tasks, and memo on the device application. | BlackBerry |
| BlackBerry custom code | Allows you to create a custom coded action for any control that supports actions. | BlackBerry |

| Action | Description | Platform |
|---|---|---|
| Windows Mobile PIM | Add this action to integrate Windows Mobile PIM applications such as calendar, contacts, and tasks on the device application. | Windows Mobile Professional |
| Windows Mobile custom code | Allows you to create a custom coded action for any control that supports actions. | Windows Mobile Professional |

*Validation*

Drag and drop onto the edit box control to add pattern-matching rule validation.

Assign rule validation to the edit box control to validate user input.

| Option | Description |
|---|---|
| Rule validation | You can assign these pattern-matching datatypes:<br><br>• Numerical<br>• String/CHAR<br>• Date/Time/DateTime |

## **Selecting a Device**

Select a device to display an empty canvas in the Screen Design page, formatted for that device.

**1.** From the device application tool bar, click the down arrow to the right of the device icon.
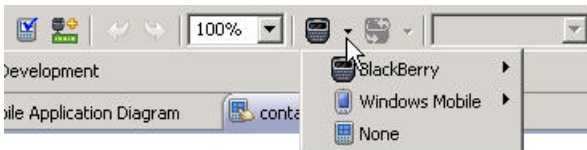


**2.** Select the device from the drop-down list.

If you select the BlackBerry Storm, the orientation icon is activated so you can select the orientation for the device:



**Note:** When you select None, you are using the device you set as the default device in the Device Application Designer preferences, but it displays a much larger canvas on the

Screen Design page. This makes it easier to add additional controls to screens that have containers that occupy most of the screen space.

### Style Sets

Style sets have their own look and feel and are used to design mobile applications that look good on multiple device platforms.

Style sets contain base styles, used for the system and user style. You can create new user styles and you can also define platform and device styles to use instead of the base style. When you create new styles, they are applicable to your current Device Application Designer project only, but you can export any styles you create for use in other projects. You can also import external styles for use in your current project.

Styles that you can create include:

- User – defines the style of the user interaction.
- Platform – when developing mobile applications, keep in mind that users of particular device platforms are accustomed to using the user interface in a certain way. Platform style can provide consistency in the user experience while ensuring the mobile application will work on a wide variety of different devices. If you do not have a platform style defined, the base style is used.
- Device – you can create styles for specific devices. If you do not have a device style defined, the platform style is used.

During screen design, styles automatically change according to the defined style set for the selected device. When you select a different device, the style set changes to reflect the new selection. If a style set is not defined for the selected device, the style set defined for the platform is used. If the style set is not defined for the platform, the base style set is used.

You can edit existing styles, including the system styles. When you edit a system style, the change is made to all controls using that system style.

#### Creating a New User Style

Use the Style Set References dialog to create new user style.

1.  Right-click in any page of the Device Application Designer (except the Introduction page), and select **Style References**.
2.  In the Style References dialog, click **New User Style**.
3.  In the Styles dialog, enter or select:

| Option | Description |
|--------|-------------|
| Name | Enter a name for the style. |
| Description | Enter a description for the style. |

| Option | Description |
|---|---|
| Colors | Select the colors for:<br>• Background Color<br>• Foreground Color<br>• Gradient Color |
| Font | Select the:<br>• Font Face<br>• Font Size<br>• Font Style |
| Preview | The style preview displays in the preview section. |

**4.** Click **OK**.

**See also**

• *Editing the Navigation Bar Style* on page 124

*Style Set References*

Use the Style References dialog to view, create, edit, and remove styles assigned to controls.

*Style Set References*

Defined system style sets, customized style sets, and user style sets are displayed in a tree. If the platform and/or the device are defined, those items can be expanded.

*Style References actions*

You can perform these actions in the Style Set References dialog:

| Action | Description |
|---|---|
| New User Style | Create a new base user style. |
| New Platform Style | Click to define the style for a different platform. For example, if you are working with a BlackBerry style set, you can use this option to define styles for Windows Mobile devices as well. |
| New Device Style | This is enabled when the selected item is a device platform, for example, BlackBerry. Click to define the style for a device belonging to that platform. |
| Edit | Edit the existing style. |

| Action | Description |
|---|---|
| Show Usage | Display information about which controls are using the selected style. |
| Remove | Remove the selected style.<br><br>**Note:** You cannot remove System styles. |
| Restore | Restore the customized style to default system styles. |
| Import | Browse the system for, and import, an existing style. The name of import style should be unique. |
| Export | Export the selected user styles or customized system styles. |
| Expand All | Click to expand all items in the tree. |
| Collapse All | Click to collapse all items in the tree. |
| Help | Click to view online help. |

*Editing Styles for Controls*

Use the Styles References dialog to make changes to an existing style.

1. Right-click in any page of the Device Application Designer (except the Introduction page), and select **Style References**.
2. In Style References, select the style you want to edit, for example, Button Focus Style, and click **Edit**.
3. In the Styles dialog, make your changes and click **OK**.

   **Note:** When editing system styles, the Name and Description fields are not editable.

*Importing Styles*

Import external styles .xml files from your system to use with your current Device Application Designer project.

1. Right-click in any page of the Device Application Designer (except the Introduction page), and select **Style References**.
2. In the Style References dialog, click **Import**.
3. In Import Styles, browse for the style .xml file to import, and click **Open**.

*Exporting Styles*
Export styles `.xml` files from your current Device Application Designer project to another
Device Application Designer project.

1. Right-click in any page of the Device Application Designer (except the Introduction
   page), and select **Style References**.
2. In the Style References dialog, select the user or customized system styles you want to
   export, and click **Export**.
3. In Export Styles, browse for the location to which you want to export the styles `.xml` file
   and click **Save**.

*Creating a New Platform Style*
Use the Style Set References dialog to create a new platform style.

1. Right-click in any page of the Device Application Designer (except the Introduction
   page), and select **Style References**.
2. In the Style Set References dialog, from the System Style Sets tree, select a base style set
   for which to define a new platform style and click **New Platform Style**.
3. In the Styles dialog, enter or select:

| Option | Description |
| --- | --- |
| Platform | Select the platform from the drop-down list. |
| Colors | Select the colors for:<br>• Background color<br>• Foreground color<br>• Gradient color |
| Font | Select the:<br>• Font face<br>• Font size<br>• Font style |
| Preview | The style preview displays in the preview section. |

4. Click **OK**.

*Creating a New Device Style*
Use the Style Set References dialog to create a new device style.

1. Right-click in any page of the Device Application Designer (except the Introduction
   page), and select **Style References**.

**2.** In the Style Set References dialog, select the platform, for example BlackBerry, for which to define the new device style and click **New Device Style**.

**3.** In the Styles dialog, enter or select:

| Option | Description |
|---|---|
| Device | Select the device from the drop-down list. |
| Colors | Select the colors for:<br>• Background color<br>• Foreground color<br>• Gradient color |
| Font | Select the:<br>• Font face<br>• Font size<br>• Font style |
| Preview | The style preview displays in the preview section. |

**4.** Click **OK**.

### Using Drag and Drop to Add a Data Source to a Screen

Drag and drop a mobile business object onto an empty screen to use the predefined data of the mobile business object, for example, a table.

**1.** In the Device Application Designer, click **Flow Design** to go to the Flow Design Canvas.

**2.** In the Palette, select **Empty Screen** and drag it onto the Flow Design canvas.

**3.** Double-click the empty screen to go to the Screen Design page.

**4.** In WorkSpace Navigator, locate the mobile business object for which to create a reference from the Mobile Application Project folder, and drag and drop it onto the empty screen. The screen is populated with the predefined data from the data source. You can view the mobile business object properties in the Properties view.

### Adding a Navigation Bar

Add a navigation bar to the screen or tab panel to enable navigation between different views in the device application.

**1.** In the Device Application Designer, go to **Screen Design > Palette**, and select **Navigation bar**.

**2.** Drag and drop the navigation bar onto the screen display or the tab panel.

#### *Navigation Bar*

A navigation bar helps users navigate client screens and keep track of where they are.

By default, the "Add a navigation bar to new screens" option is selected in Device Application Designer Preferences, so adding a new empty screen adds a navigation bar at the top of the new

screen automatically. If you unselect the "Add a navigation bar to new screens" option in Preferences, the navigation bar is not added automatically to the screen, so you have to add navigation bar manually.

When using navigation bars, keep in mind:

- The navigation bar is automatically positioned at the top of the screen and fills the width of the screen.
- You can customize the navigation bar style by clicking on the navigation bar and making your changes using the Properties view.

*Navigation Bar Definition*
Use the Properties page to configure the attributes for the navigation bar.

| Property | Description |
| --- | --- |
| Text | Enter or edit the navigation bar text. |
| Show back button | Select if you want to show the "back" button on the navigation bar. |
| Style | Select the style from the drop-down menu, or click **Search** to search available styles.<br><br>Click **Edit** to edit the selected style.<br><br>Click **Clear** to clear the style.<br><br>Click **Restore** to restore to the original style. |
| Border Style | The background color of the border style is used for the border of the navigation bar.<br><br>Click **Restore** to change the color back to its default style. |

*Editing the Navigation Bar Style*
Use the Styles dialog to edit the description, colors, and font characteristics of the navigation bar.

1. In the Screen Design page, click the navigation bar.
2. In the Properties page, click **Definition**.
3. Next to Style, click **Edit**
4. In the Styles dialog, enter:

| Option | Description |
| --- | --- |
| Name | This cannot be edited. |

| Option | Description |
|---|---|
| Description | This cannot be edited. |
| Colors | Click on the color button to select:<br>• Background Color<br>• Foreground Color<br>• Gradient Color |
| Font | Select the checkbox next to these options and select the style from the drop-down list:<br>• Font Face<br>• Font Size<br>• Font Style – select from these styles:<br>  • Bold<br>  • Italic<br>  • Underline |

You can see your changes in the Preview section of the Styles dialog.

**5.** Click **OK**.

**See also**
• *Creating a New User Style* on page 119

### Adding Items to the Menu or Table Context

Drag and drop a menu item or menu separator onto the Menu; or a context item onto the Table Context.

**1.** In the Device Application Designer, go to **Screen Design > Palette > Menu and Table Context** and select the item.

**2.** Drag and drop the selected item onto the appropriate area (Menu or Table Context), in the upper right corner of the screen display.

Context items can be used only with Grid Table or Cell Table controls.

**3.** (Optional) Use the Properties page to configure the appearance of the menu item and context item.

**Note:** There are no configuration properties for the menu separator.

*Menu*

Some devices have menus associated with each screen.

The menu is displayed in the upper right corner of the screen design page.

Menus can be different for different screens. Add menu items and menu separators to the screen menu by dragging and dropping those items from the Palette to the menu list. There is

no limit to the number of entries you can add to the menu, but consider user-friendliness when designing the menu.

*Menu Item Properties*

Use the Properties view to configure the properties for the menu item.

| Property | Description |
| --- | --- |
| Menu Item Name | Display name of the menu item. The name must be unique within a menu and must not be an empty string. |
| Use Variable | Select this checkbox to associate the text of the menu item with an existing variable. When selected, the menu item name field is disabled and the menu item name depends on the value of the selected variable. |
| | Click **Browse** to select the variable with which to associate the menu item. |
| Available on Control Selection | Select to make the menu items appear at runtime only when the selected control is in focus. This option is disabled if there are no list details on the screen. |
| | The drop-down list displays the list details to which you can associate this menu item. |

*Context Item Properties*

Use the Properties view to configure the context item.

| Property | Description |
| --- | --- |
| Context item name | Display name of the context item. The name must be unique within a context table and must not be an empty string. |
| Use variable | Select this checkbox to associate the text of the context item to an existing variable. When selected, the context item name field is disabled and the context item name depends on the value of the selected variable. |
| | Click **Browse** to select the variable with which to associate the context item. |

| Property | Description |
|---|---|
| Context item type | The type of the context item. Choose from Create, Update, Delete, Details, Relationship, or Other. |
| Default context item for the table | Select to make the context item the default context item to use when a table is acted upon on the device. For the BlackBerry Storm this is a touch or click and for other non-touch BlackBerry devices, it is a wheel-click. |

### Adding Controls to the Screen

Use the Screen Design Palette to add controls to your device application screen.

1. In the Device Application Designer, select **Screen Design > Palette > Controls** and then select the control you want to add to the screen.
2. Drag and drop the control onto the screen display.
3. (Optional) Use the Properties page for the control to configure the control.

#### Selecting a Variable to Use for a Control Property

Use the Variable Selection dialog to select variables for many of the control properties.

1. In the Properties page of the control, select Use Variable.
2. (Optional) In the Variable Selection dialog, filter the variable selection using the Limit To section.
3. Select the variable you want to use for the property from the list, and click **OK**.

### Creating Widget Events

Use the Coding tab to assign widget events to a control.

1. Go to the Properties view of the control for which you are creating a widget event.
2. Click the **Coding** tab, and enter an optional base name for the event.

   The base name represents the selected item in the generated code, for example, "SubmitButton."

   **Note:**
   - The base name must begin with a letter or underscore.
   - The base name can consist of alphanumeric characters (a–z, A–Z, 0–9) and underscores only, and must also be a unique name within the screen.
   - The base name cannot be localized.
   - The base name for display or tab panels must be unique within the screen as well as unique within the application.

- You can have a display or tab panel base name that is the same as a control base name on another screen, but it cannot be the same name as another display/tab panel.

**3.** Select the events to assign to the control individually, or click **Select All**.

**4.** From the main menu, select **File > Save**.

**5.** Use the Device Application Designer to generate the code for the device application.

**6.** The generated widget event code is saved in the Custom folder, where you can then customize it.

*Widget Event Support by Platform*
Widget event controls

*BlackBerry*

| Con-trols | onLoad | onClick | onValue-Change | onRecord-Change | onOrienta-tionChange | on-Draw |
|-----------|--------|---------|----------------|-----------------|----------------------|---------|
| **User input** | | | | | | |
| Check box | X | X | X | | X | X |
| Choice | X | X | X | | X | X |
| Edit box | X | X | X | | X | X |
| Radio group | X | X | X | | X | X |
| **Table controls** | | | | | | |
| Grid ta-ble | X | X | | X | X | X |
| List de-tail | X | X | | X | X | X |
| Cell table | X | X | | X | X | X |
| **Cell controls** | | | | | | |
| Cell label | | | | | | X |
| Cell im-age | | | | | | X |
| **General controls and containers** | | | | | | |
| Button | X | X | | | X | X |

| Con-trols | onLoad | onClick | onValue-Change | onRecord-Change | onOrienta-tionChange | on-Draw |
|---|---|---|---|---|---|---|
| Hyper-link | X | X | | | X | X |
| Image | X | X | | | X | X |
| List item | X | X | | | X | X |
| Menu item | X | X | | | X | X |
| Naviga-tion bar | X | X | | | X | X |
| Radio button | X | X | | | X | X |
| Tab fold-er | X | X | | | X | X |
| Toolbar item | X | X | | | X | X |
| **Controls with no focus** | | | | | | |
| Display | X | | | | X | X |
| Label | X | | | | X | X |
| List group | X | | | | X | X |
| Region | X | | | | X | X |
| Separator | X | | | | X | X |
| Tab panel display | X | | | | X | X |
| Toolbar | X | | | | X | X |

*Windows Mobile*

| Con-trols | onLoad | onClick | onValue-Change | onRecord-Selection-Change | onOrienta-tionChange | on-Draw |
|---|---|---|---|---|---|---|
| **User input** | | | | | | |

| Controls | onLoad | onClick | onValue-Change | onRecord-Selection-Change | onOrienta-tionChange | on-Draw |
|----------|--------|---------|----------------|---------------------------|----------------------|---------|
| Check box | X | X | X | | | |
| Choice | X | X | X | | | |
| Edit box | X | | X | | | |
| Radio group | X | X | X | | | |
| **Table controls** | | | | | | |
| Grid table | X | X | | | | |
| List detail | X | | | | | |
| Cell table | X | X | | X | | |
| **Cell controls** | | | | | | |
| Cell label | | | | | | |
| Cell image | | | | | | |
| **General controls and containers** | | | | | | |
| Button | X | X | | | | |
| Hyperlink | X | X | | | | |
| Image | X | X | | | | |
| List item | X | X | | | | |
| Menu item | X | X | | | | |
| Navigation bar | X | X | | | | |
| Tab folder | X | X | | | | |

| Con-trols | onLoad | onClick | onValue-Change | onRecord-Selection-Change | onOrienta-tionChange | on-Draw |
|---|---|---|---|---|---|---|
| Toolbar item | X | X | | | | |
| **Controls with no focus** | | | | | | |
| Toolbar | X | | | | | |
| Display | X | | | | | |
| Label | X | | | | | |
| List group | X | | | | | |
| Region | X | | | | | |
| Separator | X | | | | | |
| Tab panel display | X | | | | | |

*Widget Event Limitations on BlackBerry*
Widget event limitations on the BlackBerry device platform.

- The BlackBerry platform does not support menu events.
- On touch-screen devices, NavigationClick and NavigationUnclick methods are not supported on the check box, edit box, and radio group controls.
- The BlackBerry **navigationMovement** method is invoked when a navigation motion occurs inside a field. Since you cannot move inside a control, this event does not normally occur. However, if you add the control as the last or first control on the current screen and scroll up or down, the **navigationMovement** method is called. Therefore, you must scroll the control to trigger the **navigationMovement** method. Using the arrow key on the Simulator scrolls the control.
- The navigation bar control is unique in that it can be focused on only when it has a Back action assigned. When the navigation bar is clicked, the current screen pops from the screen stack, and as a result, the **NavigationUnclick** event is not called.
- The toolbar item control cannot be focused on on BlackBerry devices, so **OnFocus**, **OnUnfocus**, **NavigationClick**, **NavigationUnclick**, **NavigationMovement**, and **DrawFocus** are not invoked.
- **DrawFocus** is not supported for the display, region, radio groups, list detail, tab folder, toolbar, and tab panel controls, and also for controls with no focus, such as the label, separator, and toolbar item.

The onDraw event has two methods: **paint()** and **drawFocus()**. Since the toolbar cannot be focused on, only the paint method is invoked with the onDraw event.

### onLoad Event
The onLoad widget event is called when a control or container is loaded.

For example, you can use the onLoad event to indicate a new e-mail message by showing a badge with the number of unread messages before the control is displayed.

Use the onLoad event with these controls:

| Control type | Control |
|---|---|
| User input | <ul><li>Check box</li><li>Choice</li><li>Edit box</li><li>Radio group</li></ul> |
| Table | <ul><li>Grid table</li><li>List detail</li><li>Table cell</li></ul> |
| General controls and containers | <ul><li>Button</li><li>Hyperlink</li><li>Image</li><li>List item</li><li>Menu item</li><li>Navigation bar</li><li>Radio button</li><li>Tab folder</li><li>Toolbar item</li><li>Toolbar</li></ul> |
| Controls with no focus | <ul><li>Display</li><li>Label</li><li>List group</li><li>Region</li><li>Separator</li><li>Tab panel display</li></ul> |

### onClick Event

The onClick event is called when the control is acted upon.

For example, you can use this event to show a tooltip or expand a table cell to show more information for an item.

Use the onClick event with these controls:

| Control type | Control |
|---|---|
| User input | • Check box<br>• Choice<br>• Edit box<br>• Radio group |
| Table | • Grid table<br>• List detail<br>• Table cell |
| General controls and containers | • Button<br>• Hyperlink<br>• Image<br>• List item<br>• Menu item<br>• Navigation bar<br>• Radio button<br>• Tab folder<br>• Toolbar item |

### onValueChange Event

Use the onValueChange event when the value of an input control changes.

This can be useful when using linked parameters, because the values of a control change are based on the selected value of another control.

Use the onValueChange event with these user input controls:

• Check box
• Choice
• Edit box
• Radio group
• Radio button

### onRecordChange Event

The onRecordChange event is called when a user scrolls through records.

For example, it can be used for enabling or disabling menu items, activating phone actions, expanding the content of a cell, showing cell content in a fish-eye view, and so on.

Use the onRecordChange event for these table type controls.

- Grid table
- List detail
- Table cell

### onOrientationChange Event

The onOrientationChange event is called when the orientation of the device changes.

For example, this event can be called to redraw the screen on a BlackBerry Storm device when the orientation is changed.

**Note:** This event is supported only on the BlackBerry Storm 9530.

Use the onOrientationChange event with these controls:

| Control type | Control |
|---|---|
| User input | - Check box<br>- Choice<br>- Edit box<br>- Radio group |
| Table | - Grid table<br>- List detail<br>- Table cell |
| General controls and containers | - Button<br>- Hyperlink<br>- Image<br>- List item<br>- Menu item<br>- Navigation bar<br>- Radio button<br>- Tab folder<br>- Toolbar item |

| Control type | Control |
|---|---|
| Controls with no focus | • Display<br>• Label<br>• List group<br>• Region<br>• Separator<br>• Tab panel display<br>• Toolbar |

### *onDraw Event*

The onDraw event is called after the control or container is called to paint the object.

**Note:** This event is not supported on Windows Mobile devices.

Use the onDraw event with these controls:

| Control type | Control |
|---|---|
| User input | • Check box<br>• Choice<br>• Edit box<br>• Radio group |
| Table | • Grid table<br>• List detail<br>• Table cell |
| General controls and containers | • Button<br>• Hyperlink<br>• Image<br>• List item<br>• Menu item<br>• Navigation bar<br>• Radio button<br>• Tab folder<br>• Toolbar item |

| Control type | Control |
|---|---|
| Controls with no focus | • Display<br>• Label<br>• List group<br>• Region<br>• Separator<br>• Tab panel display<br>• Toolbar |
| Cell | • Cell label<br>• Cell image |

### Configuring Control Properties

Use the Properties page of any control to configure control properties.

1. In the Screen Design page, select the control for which to configure the properties, right-click, and select **Show Properties View**.
   The configurable properties for the selected control display in the Properties view.
2. Click the **Definition** tab to define properties for the control.
3. (Optional) Click the **Platform-specific** tab to configure platform-specific properties.
4. (Optional) Click the **Coding** tab to assign an event to the control.
   This tab appears only for controls that support events.
5. Configure the properties for the control and select **File > Save**.

#### Properties View for Controls

Each control has a Properties view where you can view and configure properties for the control.

The tabs shown in the Properties view vary depending on the control.

#### Definition

Define the propertis for the control, such as alignment, text, style, span, and so on. The properties shown depend on the selected control.

#### Platform-specific

Some controls have a Platform-specific tab that display only the properties that are platform-specific. The property values displayed in the Platform-specific tab can be read-only or configurable.

Use the Platform-specific tab to generate custom code subclasses for the generated code. Custom code extends the generated screen classes, which you can modify to further customize screens and controls. When you set the **Generate a custom coding subclass** option to "true,"

---

a "custom" folder is generated, where the custom subclasses are stored. This "custom" folder will not be overwritten upon subsequent code generation of the device application, so any customizations you have made to the code will remain intact.

**Note:** Platform-specific properties should have unique names within the application. If you give a property a duplicate name, the property is automatically removed and does not show on the device.

### Coding

The Coding tab is displayed for controls and containers that support events. Events allow you to specify custom events on screens and screen controls during design time. For example, a table of e-mails can display records differently (maybe with an icon for unread mail) depending on the state of the e-mail.

| Property | Description |
|---|---|
| ID | The internal ID of the control. This is a read-only field for informational purposes. Not all controls have IDs. This is left blank for controls that do not have an ID. |
| Base name | An optional name for the control or display, for example, SubmitButton for a button control. This name can be empty, but if specified, it must be unique within a screen. This name is not localizable. Only alphanumeric (a-z, A-Z, 0-9) and underscores (_) are allowed in the name. |
| Events | Lists the available events for the selected control. The list of events varies according to the type of control selected. Select the events to customize by selecting one or more of the checkboxes. |

### Label Properties

Use the Label control to display text on the device screen.

**Table 4. Definition**

| Property | Description |
|---|---|
| Label Text | Enter the text for the label. |
| Use Variable | Select to use a variable for the label text. Click **Browse** to find available variables. |
| Wrap to Multiple Lines | Select to truncate and wrap to multiple lines text that is too long to fit on one line. |

| Property | Description |
|----------|-------------|
| Style | Click **Search** to search available styles. |
|  | Click **Restore** to restore the default style. |
|  | Click **Edit** to edit the selected label style. |
|  | Click **Clear** to clear the label style. |
| Span | Choose: |
|  | • Horizontal – enter the number of columns the widget will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display or container. |
|  | • Vertical – enter the number of rows the widget will occupy. |
| Alignment | Choose the way the label displays on the screen: |
|  | Horizontal |
|  | • Left – left-justified alignment. |
|  | • Center – centered alignment. |
|  | • Right – right-justified alignment. |
|  | Vertical |
|  | • Top – positions the control at the top of the span. |
|  | • Center– positions the control in the center of the span. |
|  | • Bottom – positions the control at the bottom of the span. |

### *Edit Box Properties*

Use the Edit Box control to input text and assign values to strings, dates, decimals, integers, and so on.

**Table 5. Edit Box control properties**

| Property | Description |
|----------|-------------|
| Edit Box Text | The text that appears in the edit box. |
| Use Variable | Select to use a variable for the edit box text. Click **Browse** to find available variables. |
| Password | Select if the edit box is a password field. |

| Property | Description |
|---|---|
| Show Border | Select to show the edit box border on the device.<br><br>**Note:** If you are developing an application for the BlackBerry 9530, you must select the Show Border checkbox to ensure the screen displays properly. |
| Read Only | Sets the text in the edit box to read-only on the device. |
| Empty String Means NULL | When selected, if the user leaves the input field empty, a NULL value is passed to the server. This is required by some back-end systems.<br><br>**Note:** This option is enabled only for the STRING data type. |
| Data Type | Select the data type from the drop-down list. Use this option on the device to give menu assistance to inputs, for example, selecting a calendar for date. The device retrieves the information and performs validation on the input, for example, INT must be a number. It is also used to map the control to the operation parameter correctly. |
| Logical Type | Sets the logical type for the edit box. This can be used on the device to give menu assistance to input fields, for example, task, calendar, e-mail, or phone. |
| Validation Message | Sets the validation message that will display for data type or logical type mismatches. Click **Restore** to return the message to the default message.<br><br>**Note:** Validation messages should be constructed using the proper pattern format for the MessageFormat class (**MessageFormat.format()**). |
| Style | Click **Search** to search available styles.<br><br>Click **Restore** to restore the default style.<br><br>Click **Edit** to edit the selected edit box style.<br><br>Click **Clear** to clear the edit box style. |

| Property | Description |
|---|---|
| Span | Choose:<br><br>• Horizontal – enter the number of columns the edit box will occupy. The number must be greater than zero, and less than or equal to the number of columns in the container.<br>• Vertical – enter the number of rows the edit box will occupy. |

*Hyperlink Properties*
Use the hyperlink control to attach to an action for screen transitions.

| Property | Description |
|---|---|
| Text | Enter the text for the hyperlink. Select the **Use variable** checkbox to use a variable for the hyperlink text. Click **Browse** to find available variables. |
| Text equals value | Select so the text entered in the **Hyperlink text** edit box is treated as a hyperlink value. If the entered hyperlink value is the same as that stored in the Hyperlink text field, or if the same variable is chosen for both the Hyperlink text and value, the checkbox is automatically selected. |
| Hyperlink value | Set the value of hyperlink if it is different from the hyperlink text. An empty string or no value is acceptable. Select the **Use variable** checkbox to use a variable for the value of the hyperlink. Click **Browse** to find available variables. |
| Logical type | Sets the logical type for the hyperlink. This can be used on the device to give menu assistance to input fields, for example, e-mail, phone, image, or URL.<br><br>When a logical type is defined for the hyperlink, a decorator is shown.<br><br>Hyperlink 1 ▸ |

| Property | Description |
|---|---|
| Focus style | Select how the hyperlink appears when it is the object of focus. Click **Search** to find available focus styles. |
| | Click **Restore** to restore the default focus style. |
| | Click **Edit** to edit the focus style. |
| | Click **Clear** to clear the focus style. |
| Unfocus style | Select the appearance of the hyperlink when it is not the object of focus. Click **Search** to find available styles. |
| | Click **Restore** to restore the default style. |
| | Click **Edit** to edit the style. |
| | Click **Clear** to clear the style. |
| Span | Choose: |
| | • Horizontal – enter the number of columns the hyperlink will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen container. |
| | • Vertical – enter the number of rows the hyperlink will occupy. |
| Alignment | Choose the way the hyperlink displays on the screen: |
| | Horizontal |
| | • Left – left-justified alignment. |
| | • Center – centered alignment. |
| | • Right – right-justified alignment. |
| | Vertical |
| | • Top – positions the control at the top of the span. |
| | • Center– positions the control in the center of the span. |
| | • Bottom – positions the control at the bottom of the span. |

*Checkbox Properties*

Use the checkbox control to assign or display Boolean values.

| Property | Description |
| --- | --- |
| Text | Enter text for the check box.<br><br>Select **Use Variable** to use a variable for the check box text. Click **Browse** to find available variables. |
| Control state | Select to display a check in the check box by default.<br><br>Select **Use Variable** to use a variable for the check box state.<br><br>Click **Browse** to find available variables.<br><br>Select **Read Only** to make the check box uneditable on the device screen. |
| Logical type | Sets the logical type for the checkbox. This can be used on the device to give menu assistance to input fields, for example, task, calendar, e-mail, or phone. |
| Style | Click **Search** to search available styles.<br><br>Click **Edit** to edit the selected style.<br><br>Click **Clear** to clear the style. |
| Span | Choose:<br><br>• Horizontal – enter the number of columns the check box will occupy. The number must be greater than zero, and less than or equal to the number of columns in the container.<br>• Vertical – enter the number of rows the check box will occupy. |

| Property | Description |
|---|---|
| Alignment | Choose the way the check box displays on the screen:<br><br>Horizontal<br><br>• Left – left-justified alignment.<br>• Center – centered alignment.<br>• Right – right-justified alignment.<br><br>Vertical<br><br>• Top – positions the control at the top of the span.<br>• Center– positions the control in the center of the span.<br>• Bottom – positions the control at the bottom of the span. |

*Radio Button Properties*
Use the Properties page for radio button to edit the attributes for the radio button.

| Property | Description |
|---|---|
| Text | Enter the text for the radio button. |
| Use variable | Select to use a variable for the radio button text. Click **Browse** to find available variables. |
| Text equals value | Select so the text entered in the **Radio button text** edit box is treated as the radio button value. If the entered radio button value is the same as the one stored in the Radio button text field, or if the same variable is chosen for both the Radio button text and value, the checkbox is automatically selected. |
| Radio button value | The value to pass to any action when the radio button is selected. If not set, empty string or null will be used as the value. |

| Property | Description |
|----------|-------------|
| Span | Choose: <br><br>• Horizontal – enter the number of columns the radio button will occupy. The number must be greater than zero, and less than or equal to the number of columns in the container. <br>• Vertical – enter the number of rows the radio button will occupy. |
| Alignment | Define the alignment of the radio button within the radio group. <br><br>Horizontal <br><br>• Left – left-justified alignment. <br>• Center – centered alignment. <br>• Right – right-justified alignment. <br><br>Vertical <br><br>• Top – positions the control at the top of the span. <br>• Center– positions the control in the center of the span. <br>• Bottom – positions the control at the bottom of the span. |

*List Item Control Properties*
Use the Properties page for the List Item control to configure attributes for the list item.

**Table 6. Definition**

| Property | Description |
|----------|-------------|
| List Item Text | The text string that will display on the device screen for the list item. |
| Use Variable | Select to use a variable for the list item text. Click **Browse** to find available variables. When selected, the list item text field shows the name of the selected variable and has the variable decorator. |
| Assign Image | Click **Browse** to find an image from the Image References. |

| Property | Description |
|---|---|
| Focus Style | Select how the list item appears when it is the object of focus. Click **Search** to find available styles. |
| | Click **Restore** to restore the default style. |
| | Click **Edit** to edit the style. |
| | Click **Clear** to clear the style. |
| Unfocus Style | Select how the list item appears when it is not the object of focus. Click **Search** to find available styles. |
| | Click **Restore** to restore the default style. |
| | Click **Edit** to edit the style. |
| | Click **Clear** to clear the style. |

### *Choice Properties*

Use the Choice control to display a collection of items as a drop-down list.

**Table 7. Choice control properties**

| Property | Description |
|---|---|
| Mobile business object | Click **Search** to find the mobile business object with which to associate the selected control. |
| Object query | The object query to associate with the selected control. Leave empty for no filtering of the mobile business object. |
| Display name attribute | This option is enabled when this control is associated with a mobile business object. It is populated with the selected mobile business object attributes. This provides content to the control during run time. |
| Value attribute | This field is enabled when the Mobile Business Object field is not empty. It is populated with the mobile business object attributes. By default, the first attribute is selected. This attribute can be the same as the Display Name Attribute. |

| Property | Description |
|---|---|
| Linked value | Enter a value for linking this choice to other choice elements on the screen. The value must be from the data available in the "display_name" attribute of the mobile business object. |
| | The linked value is cleared and disabled if the mobile business object does not contain the necessary attributes with which to link parameters. |
| | The device display shows the link decorator 🔗 for any choice fields that are linked. |
| Items | Click **Add** to add the items for the choice drop-down list. |
| | Click **Edit** to edit the selected item. |
| | Click **Remove** to remove the selected item. |
| Initial value | Select the initial value for the choice drop-down menu. The choices in this drop-down list are set in the Items section. |
| Use variable | Use a variable to determine the initial value of the choice. Click **Browse** to find available variables. |
| Read only | Select if the choices displayed on the screen should be read-only. |
| Empty string means NULL | When selected, if the user leaves the input field empty, a NULL value is passed to the server instead of an empty string. This is required by some back-end systems. |
| | **Note:** This option is enabled only for the STRING data type. |
| Data type | Select the data type from the drop-down list. Use this option on the device to give menu assistance to inputs, for example, selecting a calendar for date. The device retrieves the information and performs validation on the input, for example, INT must be a number. It is also used to map the control to the operation parameter correctly. |
| Logical type | Sets the logical type for the choice. This is used on the device to give menu assistance to input fields, for example, task, calendar, e-mail, or phone. |

| Property | Description |
|---|---|
| Style | Click **Search** to search available styles.<br><br>Click **Restore** to restore the default style.<br><br>Click **Edit** to edit the selected choice style.<br><br>Click **Clear** to clear the choice style. |
| Span | Choose:<br><br>• Horizontal – enter the number of columns the widget will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display.<br>• Vertical – enter the number of rows the widget will occupy. |

### Customizing the Choice Control Items

Use the Choice Item Configuration dialog to enter items that will appear on the choice control drop-down list.

1. On the Properties page for the choice control, in the Items section, click **Add**.
2. In the Customize Choice Item dialog, enter:
   • Display Name – the text that appears on the display screen of the client.
   • Value – the internal string to be sent by the client when the choice item is selected by the user.
   • Use Variable – select to associate the item value to an existing variable.
3. (Optional) Click **Browse** to launch the variable selection dialog if you selected Use Variable in the previous step.

   This button is enabled only if you select Use Variable.
4. Click **OK**.
   The new choice item appears in the table and on the screen display.

### Editing Choice Control Items

Edit choice control items using the Properties page.

1. On the Properties page for the choice control, in the Items section, click **Edit**.
2. In the Customize Choice Item dialog, edit the values, and click **OK**.

### Button Control Properties

Use the button control Properties page to configure button appearance on the device screen.

| Property | Description |
|---|---|
| Text | Enter the text for the button. |
| Use Variable | Use a variable for the button text. Click **Browse** to find available variables. |
| Assign Image | Click **Browse** to find an image with which to associate the button control. Click **Clear** to clear the selected image. |
| Place Image at Beginning | Select to place the button image before the text. |
| Focus Style | Click **Search** to launch the Style Search dialog and select the appearanc the button will acquire when it is the object of focus. Click **Restore** to restore the default style. Click **Edit** to edit the style. Click **Clear** to clear the style. |
| Unfocus Style | Click **Search** to launch the Style Search dialog and select how the button appears when it is not the object of focus. Click **Restore** to restore the default style. Click **Edit** to edit the style. Click **Clear** to clear the style. **Note:** The unfocus style is not supported on Windows Mobile platforms. |
| Span | Choose: • Horizontal – enter the number of columns the button will occupy. The number must be greater than zero, and less than or equal to the number of columns in the container. • Vertical – enter the number of rows the button will occupy. |

| Property | Description |
|---|---|
| Alignment | Choose the way the button displays on the screen: |
| | Horizontal |
| | • Left – left-justified alignment. |
| | • Center – centered alignment. |
| | • Right – right-justified alignment. |
| | Vertical |
| | • Top – positions the control at the top of the span. |
| | • Center– positions the control in the center of the span. |
| | • Bottom – positions the control at the bottom of the span. |

*Image Control Properties*

Use the Image Properties page to configure the properties for the image widget.

| Property | Description |
|---|---|
| Span | Choose: |
| | • Horizontal – enter the number of columns the widget will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display. |
| | • Vertical – enter the number of rows the widget will occupy. |
| Assign Image | Click **Browse** to find an image to associate with the image widget. |
| Use Variable | Select to assign a table context variable to the image. |
| | Click **Browse** to open the Variable Selection dialog. Only table context variables of image types are listed in the table. This button is disabled if Use Variable is not selected. |
| | Variable Type displays only when a variable is selected and shows the name of the selected variable. |
| Show Spacing Occupied | Select if you want the image to fill in the region of the display it occupies. |

*Spacer Properties*

Use the Properties page for the spacer control to configure the spacer attributes.

| Property | Description |
|---|---|
| Span | Choose:<br><br>• Horizontal – enter the number of columns the spacer will occupy. The number must be greater than zero, and less than or equal to the number of columns in the container.<br>• Vertical – enter the number of rows the spacer will occupy on the device screen. |
| Show Spacing Occupied | Select if you want the spacer to fill in the region of the display it occupies. |

*Separator Properties*

Use the Properties page for the separator control to configure the separator attributes.

| Property | Description |
|---|---|
| Separator Thickness | Select the thickness of the separator line, in pixels. |
| Horizontal Span | Enter the number of columns the separator will occupy. The number must be greater than zero, and less than or equal to the number of columns in the container. |
| Style | The foreground color of the separator style is used for the color. The default is Separator Style. |

*Grid Table Properties*

Use the Properties page for the grid table control to configure the appearance of the grid table.

| Property | Description |
|---|---|
| Header style | Click **Search** to find and select from available styles.<br><br>Click **Restore** to restore the default style.<br><br>Click **Clear** to clear the selected style.<br><br>Click **Edit** to edit the style. |

| Property | Description |
| --- | --- |
| Odd row style | Click **Search** to find and select from available styles for the odd rows of the grid table.<br><br>Click **Restore** to restore the default style.<br><br>Click **Clear** to clear the selected style.<br><br>Click **Edit** to edit the style. |
| Even row style | Click **Search** to select from available styles for the even rows of the grid table.<br><br>Click **Restore** to restore the default style.<br><br>Click **Clear** to clear the selected style.<br><br>Click **Edit** to edit the style. |
| Pending style | Click **Search** to find and select from available styles for pending records. A pending record is a record that has been changed but not synchronized with the server.<br><br>Click **Restore** to restore the default style.<br><br>Click **Clear** to clear the selected style.<br><br>Click **Edit** to edit the style. |
| Border style | Click **Search** to find and select from available styles.<br><br>Click **Restore** to restore the default style.<br><br>Click **Clear** to clear the selected style.<br><br>Click **Edit** to edit the style. |
| Focus style | Select the appearance of the table when it is the object of focus. Click **Search** to find available styles.<br><br>Click **Restore** to restore the default style.<br><br>Click **Edit** to edit the style. |
| Context menu options | The Sort Menu option is selected by default. This adds the sort option to the context menu for the available columns in a table with data that can be sorted.<br><br>The default text for the name of the context menu item when it appears on the device is "Sort on Column." |

| Property | Description |
| --- | --- |
| Support paging | Enables paging on the record data for a table, which can reduce the memory usage for large tables.<br><br>• Start Paging Record Number – enter the number of records that should be in the table before the table performs paging. Normally this would be twice the paging size.<br>The minimum number is 25, and the maximum is 10,000.<br>• Paging Record Size – enter the number of records to store in a table page. The minimum number is 25, and the maximum is 10,000.<br>• Number of Pages to Cache – enter the number of table pages to cache in memory. This number must be 2 or more, but typically not more than 3, which conserves memory and also allows for good performance. The maximum allowed is 10 pages. |

### List Detail Properties
Use the list detail Properties page to configure the list detail attributes.

The default style for the list detail is:

• Gradient color is applied to each row.
• Odd row style is applied to all rows, even row style is ignored.
• The row value style uses the focus style.
• When focus is applied to the row, the row value uses the inverted focus style.

| Property | Description |
| --- | --- |
| Odd Row Style | Click **Search** to find and select from available styles for the odd rows of the list detail.<br><br>Click **Restore** to restore the default style.<br><br>Click **Clear** to clear the selected style.<br><br>Click **Edit** to edit the selected style. |

| Property | Description |
| --- | --- |
| Even Row Style | Click **Search** to find and select from available styles for the even rows of the list detail. |
| | Click **Restore** to restore the default style. |
| | Click **Clear** to clear the selected style. |
| | Click **Edit** to edit the selected style. |
| Pending Style | Click **Search** to select from available styles for pending records. A pending record is a record that has been changed but not synchronized with the server. |
| | Click **Restore** to restore the default style. |
| | Click **Clear** to clear the selected style. |
| | Click **Edit** to edit the selected style. |
| Border Style | Click **Search** to select from available border styles. The background color of the border style is used for the border of the list details. |
| | Click **Restore** to restore the default style. |
| | Click **Clear** to clear the selected style. |
| | Click **Edit** to edit the selected style. |
| Focus Style | The predefined style for the focus cell in the list detail. |
| | Click **Search** to preview and select from the available styles. |
| | Click **Restore** to restore the default style. |
| | Click **Edit** to edit the selected style. |
| Assign Percentages | The width of each column in the list detail, in percentages. The defaults are 50% for the first column and 50% for the second column. |
| Horizontal Alignment | Choose the way the list detail displays on the screen for each column: |
| | • Left – left-justified alignment. |
| | • Center – centered alignment. |
| | • Right – right-justified alignment. |

| Property | Description |
|---|---|
| Use Hot Keys | Select to allow users to customize hot keys for Previous and Next operations. You can select hot keys from characters a through z. The defaults are p and n, for Previous and Next, respectively. |
| Show Previous and Next Options in Menu | Display Previous and Next operations in the device menu. This option is selected by default. These menu options are displayed dynamically at runtime on the device. |
| Previous Menu Label | The label of the Previous menu. The menu label displays only when the Previous menu is shown dynamically at runtime on the device. |
| Next Menu Label | The label of the Next menu. The menu label displays only when the Next menu is shown dynamically at runtime on the device. |

*Toolbar Item Properties*
Use the Properties page to configure the appearance of the toolbar item.

**Note:** Toolbars and toolbar items are not supported on some devices.

The toolbar always appears on the bottom of the device screen.

| Property | Description |
|---|---|
| Toolbar item name | The name of the toolbar item that appears in the outline view. |
| Assign image | Click **Browse** to locate an image to associate with the toolbar item. |
| | **Note:** If you do not assign an image to the toolbar item, a warning is shown when you verify or generate the code for the device application. |
| Available on control selection | Select to show the toolbar item on the device only when something is selected on the table. |
| | **Note:** This property is not supported on Black-Berry and Windows Mobile. |

**Note:** Use the Properties view for the Toolbar container to align toolbar items left, center, or right within the toolbar.

*Selecting the Data Type for Controls*
Use the Properties view to select the data type for the control.

1. In the Properties page, from Data Type, select the data type you want to use for the control and click **Select** if it is enabled.

   The Select button is enabled if you choose any of these data types:
   - Date
   - Time
   - Datetime
2. (Optional) If you clicked **Select** in the previous step, in the dialog, select the values you want to use for the data type, and click **OK**.

*Setting the Logical Type for Controls*
Use the Properties view for the selected control to set the logical type.

1. In the Properties page, from Logical Type, select the logical type category from the first drop-down list.
2. Select the logical type from the second drop-down list.

*Logical Types*
Assign logical types to controls and mobile business object attributes.

Assigning logical types to the screen controls and mobile business object attributes allows the Device Application Designer to know how to map screen fields to device applications correctly. Logical types are divided into these categories:

- General – contains general types that are not used by any PIM applications, such as e-mail, phone, URL, and image.
- Calendar – contains types used by the calendar, such as start date, end date and so on.
- Address Book – contains types relevant to a contact, for example, first name, last name, and address.
- Task – contains types of a task, for example, priority, due date and so on.
- Memo – contains types of a memo, such as title and note.

*Configuring Cells Properties*
Use the Properties page of the cell item (cell table, cell label, cell image) to configure its properties.

1. On the Screen Design page, right-click the cell item for which you want to configure the properties and select **Show Properties View**.
   The configurable properties for the selected control display in the Properties view.
2. Configure the properties for the cell item and select **File > Save**.

*Cell Table Properties*
Use the Properties page for the cell table control to configure the appearance of the cell table.

**Note:** When using a cell table, keep in mind that, on the device, the cell table displays only text labels and images. If you want to use binary data, use the grid table control.

| Property | Description |
| --- | --- |
| Focus style | Select how the cell appears when it is the object of focus. Click **Search** to find available focus styles. |
| | Click **Restore** to restore the default focus style. |
| | Click **Edit** to edit the focus style. |
| | Click **Clear** to clear the focus style. |
| Unfocus style | Select the appearance of the cell when it is not the object of focus. Click **Search** to find available styles. |
| | Click **Restore** to restore the default style. |
| | Click **Edit** to edit the style. |
| | Click **Clear** to clear the style. |
| Number of columns | Choose the number of columns that will be in each cell. |
| Assign percentages | The width of each column in the cell, in percentages. The defaults are 33% for the first column, 34% for the second column, and 33% for the third column. |
| Support sorting | This adds the sort option to the context menu for the available columns in a table with data that can be sorted. |
| | The default text for the name of the context menu item when it appears on the device is "Sort on Column." |

| Property | Description |
|---|---|
| Support paging | Enables paging on the record data for a table, which can reduce the memory usage for large tables. |
| | **Note:** Paging is not supported on Windows Mobile devices. |
| | • Start Paging Record Number – enter the number of records that should be in the table before the table performs paging. Normally this would be twice the paging size. The minimum number is 25, and the maximum is 10,000. |
| | • Paging Record Size – enter the number of records to store in a table page. The minimum number is 25, and the maximum is 10,000. |
| | • Number of Pages to Cache – enter the number of table pages to cache in memory. This number must be 2 or more, but typically not more than 3, which conserves memory and also allows for good performance. The maximum allowed is 10 pages. |

*Configuring Cell Images*
Use the configuration dialog to choose where the image cell gets its image from.

**1.** To configure the image:
   - In the Properties view for the cell image, click **Edit**, or
   - Double-click the cell image in the Screen Design editor.
**2.** In the Image Item configuration dialog, select from these options:

| Option | Description |
|---|---|
| Mobile business object attribute | This option is enabled only if the cell table is bound to a mobile business object (MBO) with at least one attribute of the binary data type. Only MBO attributes of binary data type and image logical type can be assigned to a cell image. |

| Option | Description |
|---|---|
| Value mapping | Allows you to represent text values with images. For example, department IDs can be represented by different icons that that are more meaningful to users and more economical in terms of the limited screen space. |
| | **Note:** This option is enabled only if the cell table is bound to a mobile business object and if the mobile business object has an attribute that is not of the binary type. The image is based on the text value of the attribute. The actual image shown on the device client depends on the value of the selected mobile business object attribute at runtime. |
| State indicator | A mobile business object record can be in different states. You can use a state indicator to show these states:<br>• New – the record is new and was never in the data set.<br>• Changed – the record has changed on the server since the last synchronization.<br>• Pending – changes to the record have not yet been synchronized to the server. |
| | **Note:** Only one state indicator is allowed per cell table. If there is already a state indicator, this option is disabled. |
| Variable | Click **Search** to open the Variable Selection dialog. Only table context variables of binary + image type are listed. |
| Image reference | Cick Search to open the Image Selection dialog, where you can bind the image to an image reference. |

**3.** Click **OK**.

*Configuring the State Indicator*
Use a state indicator to represent records that are in certain states—new, changed, or pending.

You can assign multiple images to the state. The actual image shown on the device client depends on the state of the associated record.

**1.** In the Image Item configuration dialog, next to the State option, click **Configure**.
**2.** In the State Indicator dialog, next to the state you want to configure, click **Search**.

By default, a blue dot for New, a purple dot for Changed, and an orange dot for Pending is assigned. You can change the images to something else, or you can choose to show no icon for a certain state. However, at least one image should be specified.

3. In the Image Selection dialog, select the image you want to assign to the indicated state and click **OK**.

4. When you are finished configuring the state indicators, click **OK**.

*Configuring an Image to Use With a Value*
Configure images to use for values in the mobile business object attribute.

1. In the Image Item configuration dialog, select **Value Mapping**, then select the attribute to which you want to assign an image.

2. Click **Configure**.

**Note:** At least one mapping must be defined. The same image can be assigned to multiple values if needed.

3. In the Value Mapping dialog:

| Option | Description |
|---|---|
| Mobile business object attribute | This is a read-only field. |
| Default image | You can choose to display a default image if the value of the mobile business object attribute does not match any specified values. By default, **No image** is selected. Unselect the check box to enable the image selection. |
| Image assignment | This table contains the mapping of values to assigned images. Use the buttons on the right to edit the entries in the table, create new mappings, or delete mappings. |

4. Click **New** to create a new mapping.

5. In the Image Selection dialog:

| Option | Dialog |
|---|---|
| Mobile business object attribute | This is a read-only field. |
| Value | Enter the value that will be replaced by the image on the device. This value is case sensitive, which means that only a value that exactly matches this string will be replaced during runtime. You can leave this field empty to map the image to empty strings and NULLs. |

| Option | Dialog |
|--------|--------|
| Image | Click **Search** to select an image reference that represents the specified value. |

**6.** Click **OK**.

**7.** When you are finished configuring value mappings, click **OK**.

### Cell Image Properties

Use the Properties view to configure the cell image.

| Property | Description |
|----------|-------------|
| Image | A read-only field that shows the image assigned to the cell image. Click **Edit** or double-click the cell image on the screen display to make changes to the value. |
| Scale proportionally | If selected, the aspect ratio of the image is preserved during scaling. The size occupied by a cell image is restricted by the horizontal and vertical spans assigned to it. An image larger than the allocated space is scaled down to fit.<br><br>**Note:** This option is not supported on Windows Mobile devices. |
| Span | Choose:<br><br>• Horizontal – enter the number of columns the cell image will occupy. The number must be greater than zero, and less than or equal to the number of columns in the cell.<br>• Vertical – enter the number of rows the cell image will occupy. |

| Property | Description |
|---|---|
| Alignment | Choose the way the cell image displays in the cell:<br><br>Horizontal<br><br>• Left – left-justified alignment.<br>• Center – centered alignment.<br>• Right – right-justified alignment.<br><br>Vertical<br><br>• Top – positions the control at the top of the span.<br>• Center– positions the control in the center of the span.<br>• Bottom – positions the control at the bottom of the span. |

*Adding Cell Items to Cell Labels*
Use the configuration dialog to add cell items. Cell labels consist of one or more cell items.

1. Drag and drop the the **Cell label** from the Palette onto the cell table. The configuration dialog opens.

   You can also open the configuration dialog by clicking **Edit** in the Properties view for the cell label.
2. Click **Add** to create a new cell label item.
3. In the Label Item configuration dialog, select from these options:

| Option | Description |
|---|---|
| Mobile business object attribute | This option is enabled only if the cell table is bound to a mobile business object (MBO) with at least one attribute of a non-binary data type. |
| Variable | Click **Search** to open the Variable Selection dialog. Only non-binary variable types are listed for table context variables. |
| Literal | A cell label with one literal item, where the literal item is an empty string, can be used as a spacer. |

4. Click **OK**.
5. To edit a label item, select a label item from the table and click **Edit**.
6. To delete a label item, select a label item from the table and click **Remove**.

7. Change the order of label items by selecting a label item and clicking either **Move Up** or **Move Down**.

8. Preview the label text in **Label text preview**.

9. When you are finished, click **OK**.

*Cell Label Properties*
Use the Properties view to configure properties for the cell label.

| Property | Description |
|----------|-------------|
| Text | A read-only field that shows the cell label value. Click Edit, or double-click the cell label in the editor to make changes to the value. |
| Wrap to multiple lines | Select to truncate or wrap to multiple lines of text that is too long to fit on one line. |
| Focus style | Select how the cell label appears when it is the object of focus. Click **Search** to find available focus styles. |
| | Click **Restore** to restore the default focus style. |
| | Click **Edit** to edit the focus style. |
| | Click **Clear** to clear the focus style. |
| Unfocus style | Select the appearance of the cell label when it is not the object of focus. Click **Search** to find available styles. |
| | Click **Restore** to restore the default style. |
| | Click **Edit** to edit the style. |
| | Click **Clear** to clear the style. |
| Span | Choose: |
| | • Horizontal – enter the number of columns the cell label will occupy. The number must be greater than zero, and less than or equal to the number of columns in the cell. |
| | • Vertical – enter the number of rows the widget will occupy. |

| Property | Description |
|---|---|
| Alignment | Choose the way the cell label displays in the cell:<br><br>Horizontal<br><br>• Left – left-justified alignment.<br>• Center – centered alignment.<br>• Right – right-justified alignment.<br><br>Vertical<br><br>• Top – positions the control at the top of the span.<br>• Center– positions the control in the center of the span.<br>• Bottom – positions the control at the bottom of the span. |

**Note:** Only one style per cell label is supported when using the Device Application Designer. For example, you cannot create labels like "**Name:** Label" (with bold and regular text in a single label). You can use custom coding to have multiple styles in a single label, or create multiple cell labels. If you create multiple cell labels, the cell labels are in separate columns.

*Configuring Cell Items*

### Assigning Rule Validation to the Edit Box
Validate user input on the device application by assigning validation rules to the edit box control.

Rule validation is supported only for the edit box control.

1. In the Device Application Designer, select **Screen Design > Palette > Controls > Edit Box**.
2. Drag and drop the **Edit Box** onto the screen display.
3. In the **Properties > Definition** view for the edit box, select the data type from the **Data type** list.
4. In the Palette, drag and drop **Rule Validation** onto the edit box.
5. In the rule dialog, configure the validation rule and click **OK**.

### Creating Numerical Rule Validation

Use the Numerical Rule dialog to set the numerical validation rule for the edit box control.

Before a numerical validation item is added to the control, these checks are performed to enforce the integrity of all the validation rules:

- "equals" is mutually exclusive to "is greater than." "is greater than or equal to," "is less than" and "is less than or equal to."
- "is greater than" is mutually exclusive to "is greater than or equal to."
- "is less than" is mutually exclusive to "is less than or equal to."
- If both "is greater than" or "is greater than or equal to" and "is less than" or "is less than or equal to" rules are added, a check on the range is performed and vice versa.

**1.** In the Numerical Rule dialog:

| Option | Description |
|---|---|
| Rule Validation Type | Select the validation type. |
| Value | Enter the value for the validation rule. The value you enter is tested against the default of the edit box if it has one. |
| Error Message | Enter the error message that appears when rule validation fails on the client. |

**2.** Click **OK**.

### Validation Rules for Numerical Datatypes

Validation rules for numerical datatypes (INT, DOUBLE, FLOAT, DECIMAL, SHORT, LONG and INTEGER).

You can assign each of these rules only once to the numerical datatype input on an Edit Box control.

| Rule | Description |
|---|---|
| equals | Validates that the input is equal to the given number |
| is greater than | Validates that the input is greater than the given number |
| is greater than or equal to | Validates that the input is greater than or equal to the given number |
| is less than | Validates that the input is less than the given number |

| Rule | Description |
|------|-------------|
| is less than or equals to | Validates that the input is less than or equal to the given number |

### Creating String Rule Validation
In the String Rule Validation dialog, set the string validation rule for the edit box.

**1.** In the String Rule Validation dialog:

| Option | Description |
|--------|-------------|
| Rule Validation Type | Select the validation type. |
| Value | Enter the value for the validation rule. The value is tested against the default value of the Edit Box, if it has one. If the default value does not conform to the validation rule, an error message appears in the dialog.<br><br>If you select "matches regular expression" as the rule type, validate the expression by clicking **Test Regular Expression**. |
| Error Message | Enter the error message that appears when validation fails on the client. |

**2.** Click **OK**.

### Validation Rules for String Datatypes
Validation rules for STRING datatypes.

You can add multiple "contains" rules to the STRING datatype input on an Edit Box, but the rest of the validation rules can be assigned only once.

Before a string validation item is added to the control, these checks are performed to enforce the integrity of all the validations rules:

- Each type of pattern match rule, except for the "contains" rule can be added only once to the Edit Box.
- The "matches regular expression" rule supersedes other rules.
- The "length is equal to" rule is mutually exclusive to "length is greater than," "length is greater than or equal to," "length is less than," and "length is less than or equal to."
- The "length is greater than" rule is mutually exclusive to "length is greater than or equal to."
- If both "length is greater than" or "length is greater than" and "length is less than" or "length is less than or equal to" rules are added, a check on the range is also performed, and vice-versa.
- The "length is less than" rule is mutually exclusive to "length is less than or equal to."

| Rule | Description |
|------|-------------|
| contains | Validates that the input contains the given string. |
| starts with | Validates that the input starts with the given string. |
| ends with | Validates that the input ends with the given string. |
| length is equal to | Validates that the length of the input string is equal to the given length. |
| length is greater than | Validates that the length of the input string is greater than the given length. |
| length is greater than or equal to | Validates that the length of the input string is greater than or equal to the given length. |
| length is less than | Validates that the length of the input string is less than the given length. |
| length is less than or equal to | Validates that the length of the input string is less than or equal to the given length. |
| matches regular expression | This rule supersedes other rules. Do not assign any other rules to the control. Validates that the input conforms to the given regular expression. |

*Testing a Regular Expression*
Use the Regular Expression Tester to validate a regular expression.

The Test Regular Expression button is enabled in the String Rule Validation dialog when you select "matches regular expression" as the Rule Validation Type.

1. In the String Rule Validation dialog, click **Test Regular Expression**.
2. In the Regular Expression Tester dialog:

| Option | Description |
|--------|-------------|
| Validation Expression | Enter the regular expression you defined. |
| Test Value | Enter the value used to test whether or not it conforms to the regular expression defined in Validation Expression. |

3. Click **Test** to perform the regular expression test.
   The results of the validation test display in the Test Result field.
4. Click **OK**.

### Validation Rule for CHAR data

The only validation rule for the CHAR datatype is " matches regular expression."

It validates that the input CHAR conforms to the given regular expression.

### Creating DateTime Rule Validation

In the Calendar Rule dialog, set validation rules for calendar (date/time/datetime) type data.

1. In the Calendar Rule dialog:

| Option | Description |
| --- | --- |
| Rule Validation Type | Select the validation type for calendar data. |
| Select | Click to select a DATE, TIME, or DATETIME datatype value. Depending on the selected datatype of the Edit Box, a corresponding dialog is launched where you can select the data. |
| Error Message | Enter the error message that appears when validation fails on the client. |

2. In the DateTime, Time, or Date dialog, make your selections and click **OK**.
3. In the Calendar Rule dialog, click **OK**.

### Validation Rules for DateTime Datatypes

Validation rules for the DateTime datatype (DATE, TIME, and DATETIME).

You can assign only one "after" or "before" rule to a DateTime datatype input on an Edit Box.

| Rule | Description |
| --- | --- |
| after | Validates that the input is after the given time |
| before | Validates that the input is before the given time |

### Editing a Rule Validation

Use the Properties view to edit the rule validation for the edit box control.

1. In the edit box Properties view, click **Validation**.
2. In the Validation page, select the rule and click **Edit**.
3. Edit the rule validation and click **OK**.

### Removing a Rule Validation

Use the Properties view to remove rule validations from the edit box control.

1. In the edit box Properties view, click **Validation**.

**2.** In the Validation page, select the rule to remove and click **Remove**.

### Adding Controls to Containers

Group controls together by adding them to containers.

**1.** In the Device Application Designer Screen Design page, from **Palette > Containers**, select the container with which to group the controls.

**2.** Drag and drop the selected container onto the device screen.

**3.** From **Palette > Controls**, select the corresponding control.

**4.** Drag and drop the selected control onto the container. You can add as many controls as you want.

**Note:** When containers occupy the entire screen design canvas, it is difficult to add additional controls behind the container. Switch to the None setting for the canvas and use the drag-and-drop method of adding controls to the screen.
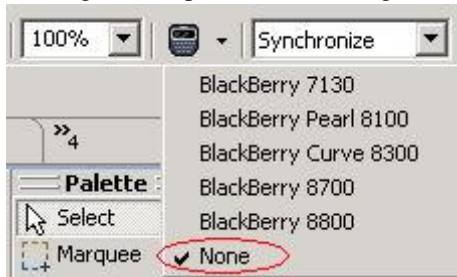


**5.** (Optional) Right-click the control and select **Show Properties View** to configure the properties for the control or container.

**6.** Select **File > Save** when you are finished.

*Region Properties*

Use the Properties page to edit the attributes for the Region container.

| Property | Description | Valid children |
|---|---|---|
| Number of Columns | The number of columns in a region. You can have 1 – 3 columns in the region. The default is 2 columns. | • Image<br>• Label<br>• Edit Box<br>• Choice<br>• Hyperlink<br>• Checkbox<br>• Spacer<br>• Button<br>• Separator |

| Property | Description | Valid children |
|---|---|---|
| Assign Percentages | The width of each column in the region, in percentages. The defaults are:<br><br>• 1 Column – 100%.<br>• 2 Columns – 35% for the first column, 65% for the second column.<br>• 3 Columns – 33% for the first column, 34% for the second column, and 33% for the third column. | |
| Span | Choose:<br><br>• Horizontal – enter the number of columns the widget will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display.<br>• Vertical – enter the number of rows the widget will occupy. | |
| Show Border | Select to display a border around the region. The foreground color of the style is used for the border color. | |
| Style | Select the style to use for the region. Controls inside the region inherit the region style for the parts of the style not set on the controls. For example, if you define green as the background color for the region, the region's background is set to green, and each control in the region will also have a green background. | |

### *Radio Group Properties*

Use the Properties page to edit the attributes for the Radio Group container control.

| Property | Description |
|---|---|
| Number of Columns | The number of columns in the radio group. You can separate tabs in to 1 – 3 columns. The default is 1. |

| Property | Description |
| --- | --- |
| Assign Percentages | The width of each column in the radio group, in percentages. The defaults are:<br><br>• 1 Column – 100%.<br>• 2 Columns – 35% for the first column, 65% for the second column.<br>• 3 Columns – 33% for the first column, 34% for the second column, and 33% for the third column. |
| Selected Radio | Select the radio button that is shown as selected on the device initially. |
| Use Variable | Select to associate the selected radio button with the initial value of an existing variable. The initial value is what determines which radio button in the group is selected by default when the screen is initially seen.<br><br>**Note:** If the value variable of the selected button is changed, the radio group variable does not automatically change. At least one button without a value variable must be available to be selected to unset the value variable for the radio group.<br><br>Click **Browse** to find existing variables. |
| Read only | Sets the radio button to read-only on the device. The radio button appears as light gray on the device. |
| Empty button value means NULL | When selected, if the user leaves the input field empty, a NULL value is passed to the server instead of an empty string. This is required by some back-end systems.<br><br>**Note:** This option is enabled only for the STRING data type. |
| Data Type | Select the data type from the drop-down list. Use this option on the device to give menu assistance to inputs, for example, selecting a calendar for date. The device retrieves the information and performs validation on the input, for example, INT must be a number. It is also used to map the control to the operation parameter correctly. |

| Property | Description |
|---|---|
| Logical Type | Sets the logical type for the radio group. This is used on the device to give menu assistance to input fields, for example, task, calendar, e-mail, or phone. |
| Selected control style\
Unselected control style | Select the styles for the way the radio buttons will appear when they are selected or unselected on the user interface. Click **Search** to search available styles.\
Click **Edit** to edit the selected style.\
Click **Clear** to clear the style. |
| Span | Choose:\
• Horizontal – enter the number of columns the radio group will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display.\
• Vertical – enter the number of rows the radio group will occupy. |

### Tab Folder and Tab Panel Properties

Use the Properties page to edit the attributes for the Tab Folder and Tab Panel container controls.

### Tab Folder properties

| Property | Description |
|---|---|
| Current panel | Shows the currently selected active tab. Select the active tab from the drop-down list. |
| Panel order | Drag and drop the tab names to arrange the order in which the tabs appear. Click **New** to create a new tab in the tab folder.\
Click **Remove** to remove the selected tab from the tab folder. This button is only activated if there is more than one tab in the tab folder. |

### Tab Folder styles properties

These properties allow you to select the colors for the active tab and the inactive tab.

| Property | Description |
|---|---|
| Active tab | Select the style for the active tab. |

| Property | Description |
|---|---|
| Inactive tab | Select the style for the inactive tab. |
| Tab border style | Select the style for the tab border. |

*Current Tab Panel properties*

| Property | Description |
|---|---|
| Panel title | The name of the tab. |
| Assign image | Adds an image to the selected tab of the panel. |
| Show pattern | Select to add a pattern to the background color. <br><br>**Note:** This option is not supported on Windows Mobile devices. |
| Background image | Select a background image to show in the tab. This will show behind the widgets. |
| Scale image to fill display size | Stretch the image to fit the display size. This is not always desired for logos, but is good for pattern-based images. |
| Style | Select the style to use for the tab panel. Controls inside the tab panel inherit the style for style parts not set on the controls. |
| Number of columns | The number of columns in an active tab. You can separate tabs in to 1 – 3 columns. The default is 2. |
| Assign percentages | The width of each column in the active tab, in percentages. The defaults are: <br><br>• 1 Column – 100%. <br>• 2 Columns – 35% for the first column, 65% for the second column. <br>• 3 Columns – 33% for the first column, 34% for the second column, and 33% for the third column. |
| Horizontal alignment | • Left – left-justified alignment. <br>• Center – centered alignment. <br>• Right – right-justified alignment. |

### *List Group Properties*

Use the Properties page for List Group to edit the attributes for the List Group container control.

| Property | Description |
|---|---|
| Span | Choose:<br>• Horizontal – enter the number of columns the list group will occupy. The number must be greater than zero, and less than or equal to the number of columns in the screen display.<br>• Vertical – enter the number of rows the list group will occupy. |

### *Toolbar Properties*

Use the Properties page to edit the attributes for the toolbar container control.

**Note:** The toolbar is not supported on some devices. If the toolbar control displays in the Palette, it is supported for the selected device.

| Property | Description |
|---|---|
| Style | Select the default toolbar style. Click **Search** to launch the Style Search dialog, where you can search and select from available styles.<br>Click **Restore** to restore the default style.<br>Click **Edit** to edit the selected label style.<br>Click **Clear** to clear the label style. |
| Border style | Select the default style for the toolbar border. Click **Search** to launch the Style Search dialog, where you can search and select from available styles.<br>Click **Restore** to restore the default style.<br>Click **Edit** to edit the selected label style.<br>Click **Clear** to clear the label style. |
| Horizontal alignment | The toolbar is always centered and occupies the full width of the device screen. For toolbar items within the toolbar container, you can choose Left, Center, or Right alignment. |

### Assigning an Action to a Control

You can assign actions to controls that lead to the performance of actions.

### Prerequisites

You must create a control that supports actions, for example, a hyperlink or list item.

### Task

1. Drag and drop the selected action from **Screen Design > Palette > Actions** on to the control on the device screen.

   **Note:** If the control does not allow the drop action, it is a control that does not support actions.

   You can also use the Property page to add actions to controls:

   1. In the Property page for the Action, in the drop-down list to the right of the **New** button, select the type of action you want to create.
   2. Enter the information in the Action dialog, and click **OK**.

      **Note:** You can highlight an action in the property page and right-click to copy and paste the action.

2. (Optional) Enter the information for the action in the action dialog.

   The Exit, Logout, and Refresh actions do not have properties that you can configure, so do not have configuration dialogs.

#### *Creating an Alert Action*

Use the Screen Design page palette to create an alert action for the selected control.

1. In the Screen Design page of the Device Application Designer, under Palette, select **Actions > Alert**.
2. Drag and drop the alert action onto the control.

   **Note:** You can drop actions only on to controls that accept actions. When you drop the action on to the control, the action decorator appears on the control.

3. In the Create an Action dialog, select the type of alert from the drop-down list:
   - Error
   - Information
   - Question
4. Click **New**.
5. Enter or select:

| Option | Description |
|--------|-------------|
| Screen | The screen with which to associate the alert from the drop-down list. This drop-down list is populated with the screens in the Flow Design page. |
| Message | The text for the alert message to be displayed. |
| OK Button Label | The text for the button the user clicks when the alert is displayed. |
| Cancel Button Label | This field is shown only if you are creating a Question alert. |

6. Click **OK**.
7. In the Create an Alert Action dialog, you can assign more alert actions to screens. When you are finished, click **OK**.
   The alerts display in detail pane of the Properties page.

   **Note:** Any action that navigates to another screen must be the last action in the listed actions.

### *Creating an Object Query Action*
Add an object query action to a table or list detail control.

The object query action has these limitations:

- Paging is not supported for object queries. If you add an object query action to the table, all data is shown.
- Sort and Search features are not supported.
- Object queries with a return type of result set are not supported.
- If the object query parameter is of the DateTime datatype, the millisecond length must be two characters, for example, `2010-02-06 06:46:15.16`. If you use a longer millisecond length, the record cannot be fetched.

   **Note:** An object query is not supported on child mobile business objects

- Personalization keys are not supported.

1. In the **Screen Design > Palette**, select the control to which you want to assign the object query action, for example a grid table, and drag and drop it onto the screen.
2. In WorkSpace Navigator, locate the mobile business object or synchronization group that contains the object query you want to assign to the control, and drag and drop it directly onto the control.
3. From **Palette > Controls**, select any action-enabled control, for example, a button or menu item and drag and drop it onto the mobile business object screen.

**4.** From **Palette > Actions**, select the **Object Query** action and drag and drop it onto the control you added in the previous step.

**5.** In the Object Query Action dialog, enter or select:

| Property | Description |
|---|---|
| Control | Select the control to which to assign the object query. This updates the grid table or list view to use the values of the object query. This over-writes whatever is set in the initial configured object query in the mobile business object. |
| Mobile business object | This is filled in based on the selected control. |
| Object query | Select the object query. The object query pa-rameters are filled in automatically. You can configure any of the parameters by double-clicking on the selected parameter. |
| Required field alert message | The alert message the user sees if a required field is not filled out. For example, if Name is a required field, and the user does not enter any information, an alert message tells the user to enter the required information. |
| Exceeded field alert message | You can limit the string length parameter in mobile business object forms. Enter the alert message to display when the user exceeds the maximum string length allowed when the form is submitted. |

**6.** Click **OK**.

*Configuring the Initial Object Query Parameters*
Set parameter values for the object query that is initially loaded with the table data.

These are the values that are used in the object query unless an object query action is applied to the control.

**1.** In the Mobile Business Object Properties view, select the object query from the Object Query drop-down list.

**Note:** The Mobile Business Object property page displays when the selected control, such as a table, list detail, is associated with a mobile business object.

**2.** Click **Initial Parameter Values**.

**3.** Select the object query value to configure and click **Configure**.

**4.** In the Input dialog, select from these options:

| Option | Description |
|---|---|
| Variable | Select to use a variable. Either click **Browse** to browse for and select existing variables to associate with the parameter, or click **New** to define a new variable to associate with the object query parameter. |
| Literal | Select to use a fixed value know as a literal. |
| Personalization key | Not supported for object queries. |
| NULL | Use a NULL value assigned to the parameter. When a parameter has a personalization key or default value, and you set the parameter value to null, the explicit sync parameter always overrides the personalization value and default value.<br><br>**Note:** Null does not override personalization key or default values for create, update, and delete operation parameters. |

**5.** Click **OK**.

*Creating Object Query Input Parameters Using Content Assist*
Create input parameters on the screen where the user can enter values that can be executed on the client database to fetch a record.

**Prerequisites**
The device application must reference a deployed mobile business object that has an object query with a parameter defined.

**Task**

**1.** Drag and drop the mobile business object onto the Flow Design page to create the device application.
**2.** Double-click the screen that references the mobile business object.
**3.** In the Screen Design page, click the mobile business object icon.

A list of options is displayed, which includes the option to create the object query inputs, as shown in this example, where the object query is performing a search on the first and last name of an employee:
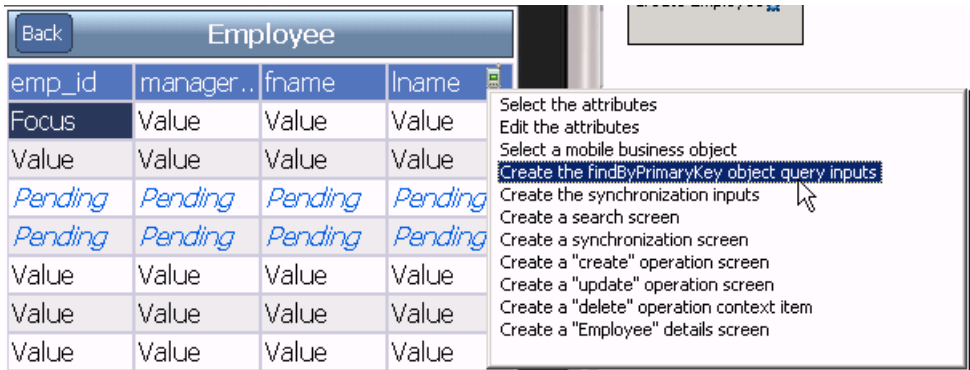
**4.** Double-click the option to create the object query inputs.

The object query inputs display on the device screen as shown in this example, where the input fields for fname (first name) and lname (last name) are added above the table, as well as a Submit button with the object query action assigned.
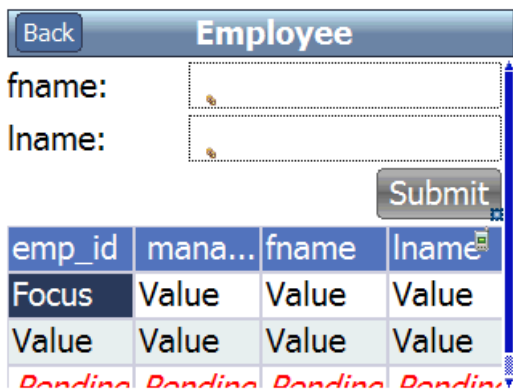


After the user clicks **Submit**, results are returned and the table is refreshed with the results.

**5.** Select **File > Save**.

### Creating an Operation Action

You can create an action based on the operation of a selected mobile business object.

**1.** From WorkSpace Navigator, select the operation for which you want to create an action, and drag and drop the operation onto the control.

**Note:** You can drop actions only on to controls that accept actions. When you drop the action onto the control, the action decorator appears on the control.

**2.** The Operation Action dialog displays and shows the information for the operation.

| Option | Description |
|---|---|
| Synchronization group | Shows the name of the synchronization group that contains the mobile business object that executes the operation. |
| Mobile business object | The mobile business object with which the operation is associated. |
| Operation | The name of the operation. You can select a different operation from the drop-down list. |
| Place the operation in pending state | If this option is selected, the operation is not sent to the server automatically (for both message based and replication based mobile business objects). To synchronize the mobile business object, you must perform the synchronization action or use the synchronization stock screen.<br><br>If this is not selected, the operation is synchronized immediately, automatically. |
| Required field alert message | This is the alert message the user sees if a required field is not filled out. For example, if Name is a required field, and the user does not enter any information, an alert message tells the user to enter the required information. |
| Exceeded length alert message | Limit the string length parameter in mobile business object forms. Enter the alert message to display when the user exceeds the maximum string length allowed when the form is submitted.<br><br>In the message, the {0} will be replaced with the parameter name and the {1} will be replaced with the size of the parameter. |

3. (Optional) Select a parameter and click **Configure** to configure the input for the CGI parameters of the operation.

   Click **Restore** to restore default values.

4. Click **OK**.

*Searching for an Operation*

Use the Search for Operation dialog in mobile buisness objects to search for an operation with which to associate an action. In local business object, the search operation acts as a filter.

1. In the Operation Action dialog, click **Search**.
2. In the Search for Operation dialog, enter the search criteria:

**Table 8. Search criteria**

| Option | Description |
|--------|-------------|
| Name | Enter the name of the mobile business object. |
| Project | Select the project in which you want to search from the drop-down list, or select All Mobile Application Projects to search all mobile application projects. |
| **Advanced** | |
| Synchronization group | Enter the name of the synchronization group in which to search, or select the group from the drop-down list. |
| Data source reference | Filter a search based on data source information; search for items that are bound to a data source, data source type, or connection profile name. |
| Role assignments | Select role mode and enter specific role names. Use wildcards, "*" ; matches all, "?" matches single character. |
| Personalization key reference | Select an existing personalization key from the drop-down list. |

Click **OK**.

*Configuring Parameters For an Operation Action*
Use the Operation Action dialog to configure the parameters of the operation action.

1.
   In the Screen Design page, click on the action decorator ⚙ for which to configure the operation parameter.
2. In Content Assist, double-click **Edit Operation Action**.
3. In the Operation Action dialog, select the parameter to configure and click **Configure**.
4. In the Parameter dialog, select:

| Option | Description |
|--------|-------------|
| Variable | Select and click **Browse** to browse for available variables to associate with the parameter. Click **New** to define a new variable to associate with the parameter. |
| Literal | Select to use a fixed value. |
| Control | Select to use a control that is defined on the screen |
| Personalization key | This is enabled only if the mobile business object parameter has a personalization key associated with it. |

| Option | Description |
|---|---|
| NULL | Use a NULL value assigned to the parameter. When a parameter has a personalization key or default value, and you set the parameter value to null, the explicit sync parameter always over-rides the personalization value and default value. |
| | **Note:** Null does not override personalization key or default values for create, update, and delete operation parameters. |
| Mobile business object | If enabled, the parameters can be pointed to the structured mobile business object shown. |

**5.** (Optional) Click **Next** to configure another operation parameter.

**6.** Click **OK**.

*Editing an Operation Action*
Use the Operation Action dialog to edit an operation action.

**1.** In the Screen Design page, click on the action decorator 🔰 for which to configure the operation parameter.

**2.** In Content Assist, double-click **Edit Operation Action**.

**3.** Make your changes and click **OK**.

**4.** You can also edit actions using the Properties view:

    a) Select the control to which the operation action is assigned.

    b) In the Properties view, click the **Actions** panel.

    c) Select the operation action to configure and click **Edit**.

*Creating a Tab Activation Action*
You can add a tab activation action to controls if the current screen contains a tab folder.

**1.** In the**Properties > Action** page of the control for which you want to create the tab activation action, click the drop-down list next to New, and select Tab Activation.

**2.** In Tab Activation Action, select from:

| Option | Description |
|---|---|
| Previous Tab Panel | Select to set the action to go to the tab panel on the left of the current tab panel in the tab folder. |

| Option | Description |
|---|---|
| Next Tab Panel | Select to set the action to go to the tab panel to the right of the current tab panel in the tab folder. |
| Tab Panel | Select to set the action to go to a specific tab panel, selected from the drop-down list. |

**3.** Click **OK**.

### Creating a Connection Action
Use the Connection Action dialog to edit the connection type and screen for the action.

**1.** In the **Screen Design > Palette > Actions**, select Connection and drag and drop it on to the control.
**2.** In the Connection Action dialog, from the Connection drop-down list, select the type of connection you want to assign to the action:
   • Go To
   • Cancel
   • Error
   • Pending
   • Success
**3.** Select the screen to which you want the connection action to point from the drop-down list on the right. If no screens appear in the drop-down list, click **New**.
**4.** (Optional) If no screens appear in the drop-down list, click **New**.
**5.** (Optional) In the Select Screen dialog, select the screen from the drop-down list, and click **OK**.
**6.** In the Connection Action dialog, click **OK**.

### Creating a Synchronize Action
Use the Synchronization Action dialog to select a synchronization group and configure its mobile business object synchronization inputs.

**1.** In the **Screen Design > Palette**, select the control to which you want to assign a synchronization action, for example a button, and drag and drop it onto the screen.
**2.** In WorkSpace Navigator, locate the mobile business object or synchronization group that contains the synchronization parameters you want to assign to the control, and drag and drop it directly onto the control.

When using content assist, the option to create a "synchronize <MBOname>" menu is automatically added only if the mobile business object has no synchronization parameters mapped. If there are synchronization parameters already configured in the mobile business object, you can select the "create inputs" option to create the form data with a synchronization button.

**Note:** If the mobile business object to which you are adding a synchronization action has a child mobile business object, you must add a Refresh action to the screen of the child mobile business object so that the screen refreshes with the updated data.

    a) (Optional) An alternate method for assigning a synchronization action to a control is to select the control from the Palette to which you want to assign the synchronization action and drag it onto the screen.

    b) From Actions, select **Synchronize**, and drag and drop it onto the control.

    c) In the Synchronization Action dialog, click **Search** to launch the Search for Synchronization Group dialog.

    d) Select the synchronization group with the mobile business object that contains the synchronization parameters you want to assign to the control from the list, and click **OK**

    If parameters are defined for the mobile business object, the parameter table is populated.

The Synchronization Action dialog launches, with the synchronization parameters automatically listed.

**3.** (Optional) Select the checkbox next to the synchronization parameter if the input is required by the mobile business object synchronization. If selected, a value is required.

**4.** Review the synchronization parameters and click **OK**.

*Synchronization Action Inputs*
The Synchronization Action dialog shows the synchronization action inputs that you can configure.

| Option | Description |
|---|---|
| Synchronization group | Click **Search** to search for a synchronization group. Mobile business objects contained in the synchronization group are listed in the Mobile Business Object drop-down list in the Customize section. |
| Customize synchronization | If not selected (default), synchronization parameters are used ot synchronize according to the following order: user explicit value > pk explicit value > pk default value > parameter default value. If selected, you can customize the synchronization parameters of the selected mobile business object from the Mobile Business Object drop-down list, and synchronize the synchronization group. |
| Mobile Business Object | The drop-down list contains all the mobile business objects of the specified synchronization group. |

| Option | Description |
|---|---|
| Required Field Alert Message | The alert message the user sees if a required field is not filled out. For example, if Name is a required field, and the user does not enter any information, an alert message tells the user to enter the required information. |
| Exceeded Length Alert Message | You can limit the string length parameter in mobile business object forms. Enter the alert message to display when the user exceeds the maximum string length allowed when the form is submitted. |
| Save synchronization parameters | Select to save the synchronization parameters of mobile business objects for later use. In a replication-based application, if this option is selected, synchronization will not occur immediately.<br><br>On messaging-based applications, synchronization is performed immediately, regardless. |
| Input | This lists all the parameters and filter by (sync) attributes associated to the mobile business object. Double-click on any row to open the Input dialog. This list is empty if there are no parameters defined for the mobile business object. |

*Configuring Synchronize Action Inputs*
Use the Synchronization Action dialog to edit the parameters of the synchronize action.

1. In the Properties page for Actions, select the synchronization action for which you want to configure the inputs and click **Edit**.
2. In the Synchronization Action dialog, select the parameter you want to configure input for and click **Configure**.
3. In the Parameter dialog, select from these options and click **OK**.

| Option | Description |
|---|---|
| Variable | Select and click **Browse** to browse for available variables to associate with the parameter. Click **New** to define a new variable to associate with the parameter. |
| Literal | Select to use a fixed value. |
| Control | Select to use a control that is defined on the screen |

| Option | Description |
|--------|-------------|
| Personalization key | This is enabled only if the mobile business object parameter has a personalization key associated with it. |
| NULL | Use a NULL value assigned to the parameter. When a parameter has a personalization key or default value, and you set the parameter value to null, the explicit sync parameter always overrides the personalization value and default value. |
|  | **Note:** Null does not override personalization key or default values for create, update, and delete operation parameters. |
| Mobile business object | If enabled, the parameters can be pointed to the structured mobile business object shown. |

**4.** Click **OK**.

*Creating a Persist Action*

Use the Persist Action dialog to configure the persistent values for variables.

**1.** In the **Screen Design > Palette > Actions**, select **Persist** and drag and drop it on to the control.

**2.** In the Persist Action dialog, click:

- **New** – to create a new variable with persistent values, or
- **Configure** – to configure the persistent values for the selected variable.

**3.** Click **OK**.

*Configuring a Persist Action*

Use the Persist Action dialog to configure persist actions for selected variables.

**1.** In the Screen Design page, click on the action decorator for which you want to edit the persist action values.

**2.** In Content Assist, double-click **Edit Persist Action**.

**3.** In the Persist Action dialog, select the variable for which you want to configure the persist action, and click **Configure**.

**4.** In the dialog, select:

| Option | Description |
|---|---|
| **Set as Persistence** | Select to set the selected variable for persistence when the action is run. The literal, or control, value will be assigned to the variable. |
| | **Note:** Only user variables can be assigned in a persistence action. |
| **Literal** | Select to use a fixed value. |
| **Control** | Select to use a control that is defined on the screen. |

5. (Optional) Click **Next** to configure another persist action.
6. Click **OK**.

### Creating a Save Context Action

Add a Save Context action to a control.

As with other actions, the control to which you add the Save Context action must support the action. The screen must contain at least one table or list detail widget.

1. From **Screen Design > Palette > Actions**, select Save Context.
2. Drag and drop the Save Context action onto the appropriate control on the Screen Design page.
3. In the Save Context Action dialog, select:
   • Control – select the table or list detail for which to save the context.
   • Relationship – (optional) select the relationship to use when switching screens.
4. Click **OK**.

### Save Context Action

Assign the Save Context action to screens with a table or list detail widget to maintain context when navigating between screens on the device.

Add a Save Context action to a control so that when navigating from one table screen to another on the device, the device client remembers what table row was in focus before it switches to the next screen to maintain the context and display the correct information on the second screen. If you do not explicitly add this action to the control, context is not automatically saved from screen to screen.

When using the Save Context action, keep in mind the following:

• The screen must contain at least one table or list detail widget.
• Each widget can have only one Save Context action assigned.
• When a table or list detail widget is deleted, the related Save Context action is also deleted.
• User selects the relationship when Save Context action is defined. When the relationship that was defined for the Save Context action is changed or removed, the problem check shows a warning in the Problems view. Update the Save Context when you change or remove relationships from the Save Context action.

### *Editing a Save Context Action*
Edit the control and relationship for a Save Context action.

1. Click on the Save Context action icon on the control.
2. Double-click **Edit Save Context Action**.
3. In the Save Context Action dialog, make the changes then click **OK**.

### *Creating a Navigate Back Action*
Add a navigate back action to a control so the user can go back to the original screen.

1. In the Screen Design page, from **Palette > Actions** click **Navigate back**.
2. Click the control on which you want to drop the action.

   **Note:** There are no configuration properties for the navigate back action since it performs only one action—returning to the original screen.

### *Integrating PIM Applications*
Adding the BlackBerry or Windows Mobile PIM action to your controls allows your application to integrate with PIM applications like the calendar, address book, tasks, and memo.

Assign logical types to controls and mobile business object attributes that the PIM action will reference. The PIM action requires that the controls to which it is mapped have input fields, for example, edit box, choice, checkbox, and radio group.

PIM context menu items are ordered by alpha within a logical type scope. For example, if there are two phone numbers, one URL, and one e-mail address, the context menu appears like this on the device:

- Call Phone1
- Call Phone2
- Email to address1
- Go to URL1
- SMS Phone1
- SMS Phone2

Once you map the fields and mobile business object attributes to logical types, add the PIM action to a control to which actions can be assigned, such as the button, hyperlink, list item, and menu item.

For example, to integrate a device application with the BlackBerry address book application, assign the Address Book as the logical type category and then select from the logical types such as First Name, Last Name and so on.

*Adding a PIM Action*

Add the PIM action to a control to integrate the control with a Windows Mobile or BlackBerry PIM application.

You can assign only one BlackBerry PIM action per control.

1. Drag and drop the BlackBerry or Windows Mobile PIM action from the Palette onto a control or menu item that accepts an action. You can also add the PIM action using the Properties view for the selected control:

   a) Select the control to which you want to add the BlackBerry or Windows Mobile PIM action.
   b) In the Properties view for the control, select **Actions** from the left pane.
   c) In the Actions dialog, click the down arrow next to **New** and select **BlackBerry PIM** or **Windows Mobile PIM**.

2. In the PIM Action dialog, select from these configuration options:

| Option | Description |
|---|---|
| BlackBerry/Windows Mobile PIM Application | Select the PIM application with which the action will interact. |
| Read from Application | By default, this option is selected, and indicates that the action should retrieve information from the PIM application. When selected, the PIM application is launched and a custom menu called "Import to <client name>" is added to all the PIM applications. This menu appears on the context menu of each PIM application on the device. |
| | When the PIM action is called, the specified PIM application is launched. The user selects a specific contact, calendar, memo, or task and clicks the "Import to <client name>" menu item, which activates the PIM action on the Device Application Designer, where it loops through the list of PIM controls on the current screen. PIM controls that have valid and compatible PIM fields and attributes are populated with the corresponding value from the PIM application in which the menu was clicked. |

| Option | Description |
| --- | --- |
| Write to Application | Select this option for the action to save information to the PIM application. |
| | Values are taken from the PIM controls, table, or list detail layout. The logical types are then checked to see which PIM fields and attributes are valid. The information is then saved into the corresponding PIM application. |
| Data Source | When writing information to the PIM application, this option indicates where the client should get information from. For example, if you select Display, the client searches the entire screen display for controls that have logical datatypes mapped. If you select Table, the client searches only in the table. You can select Data Source only if Write to Application is also selected. |
| Launch PIM Application | This option is available only if Write to Application is selected. If selected, the corresponding application is launched after writing the PIM action. |
| | **Note:** BlackBerry only. |

**3.** Click **OK**.

**4.** Select **File > Save**.

*Editing a PIM Action*
Use the Properties view to edit the PIM action.

**1.** In the Properties view for the selected control, click **Actions** in the left pane.

**2.** From the list of actions, select the PIM action to edit and click **Edit**.

**3.** Make your changes and click **OK**.

*Creating a BlackBerry Custom Action*
Create a user coded action for BlackBerry devices.

**1.** In the Palette, select **BlackBerry Custom Code** and drag and drop it onto the control that supports the action.

**2.** In the BlackBerry Custom Action dialog, browse for the class file that contains the custom coding that implements the action.

**3.** (Optional) Click **Create Action Class** to create a new Java class for the action.

a) In the New Java Class wizard, enter:

| Option | Description |
| --- | --- |
| Source Folder | By default, this is empty. Click **Browse** to locate the source folder for the Java class. |
| Package | Click **Browse** to locate the package for the new Java class. |
| Name | Enter a name for the custom action. |
| Superclass | By default, this is com.sybase.uep.bob-client.actions.Action.<br>1. Click **Browse**.<br>2. In the Superclass Selection dialog, enter:<br>    • Choose a Type<br>    • Matching Items<br>3. Click **OK**. |
| Interfaces | By default, this is populated with the com.sybase.uep.bobclient.actions.IBOBAction interface. Click **Add** to select interfaces implemented by the new class. |
| Which Method Stubs Would You Like to Create | • Public Static Void Main – leave this option unselected.<br>• Constructors From Superclass<br>• Inherited Abstract Methods |
| Do You Want to Add Comments | Select **Generate Comments** to add comments. |

    b) Click **Finish**.

**4.** (Optional) Expand the **Advanced** section to specify advanced settings for:

| Option | Description |
| --- | --- |
| Source package | Select the folder to include any of the source packages to compile into the client for the source screen. |
| Extra source package | Select the folder that contains any additional source package to compile into the client. |
| Resources | Select a folder to include any resources, for example, images, in the client. |
| Third party libraries | Select a folder with libraries (*.jar files) to compile against. |

**5.** Click **OK**.

### Editing a BlackBerry Custom Action

Use the Edit the BlackBerry Custom Action dialog to edit BlackBerry custom actions.

1. In the Properties view for the selected control, click **Actions** in the left pane.
2. From the list of actions, select the custom action you want to edit and click **Edit**.

   The BlackBerry Custom Action dialog appears. If there are advanced settings specified for the action, the Advanced section is expanded.
3. Make the changes to the action and click **OK**.

### Removing an Action

Use the Properties view to remove actions from controls.

1. In the Properties view for the selected control, click **Actions** in the left pane.
2. From the list of actions, select the action you want to remove and click **Remove**.

### Creating a Windows Mobile Custom Action

Create a user coded action for Windows Mobile devices.

1. In the Palette, select **Windows Mobile Custom Code** and drag and drop it onto the control.
2. In the Windows Mobile Custom Action dialog, enter the C# class name and click **OK**.

### Creating a Screen Template

Create templates from screens so that the information for the selected screen is saved for re-use.

1. In the Flow Design page of the Device Application Designer, select the screen for which you want to create a template and right-click.
2. In the menu, select **Create Screen Template**.
   The screen name appears in the **Palette > Screens** menu. This allows you to drag and drop this screen, containing all of its defined items, onto the Flow Design

### Deleting a Screen Template

Use the Customize Palette to delete user-defined screens from the Palette menu.

1. Right-click anywhere on the Palette and select **Customize**.
2. In the Customize Palette dialog, select the screen you want to delete from the left pane, right-click and select **Delete**.
3. Click **OK**.

### Creating Unwired Platform Profile References

Use the Unwired Platform Profile References dialog to create new profile references or to edit existing profile references.

1. Right-click in any page of the Device Application Designer (except the Introduction page) and select Unwired Platform Profile References.
2. In the Unwired Platform Profile References dialog, click **New**.
3. In the Unwired Platform Profile dialog, enter:

| Option | Description |
|---|---|
| **Name** | Enter a name for the profile reference. |
| **Sybase Unwired Platform Profile** | Select the Unwired Platform profile for which you want to create a reference from the drop-down list. |
| **Set as Login Profile** | Select to use this profile as the login profile. |
| **Create Profile** | Click to launch the dialog to create a new connection profile if you do not want to use an existing profile. |

4. Click **OK**.

*Unwired Platform Profile References*

Use the Unwired Platform Profile References dialog to view, create, edit, and remove profile references.

*Unwired Platform Profile References table*

| Column | Description |
|---|---|
| Display Name | Shows the display name of the profile. The default login profile is marked with an icon  . |
| Unwired Platform Profile | Shows the Unwired Platform connection profile to which the profile reference is mapped. |

*Unwired Platform Profile References Actions*

You can perform these actions in the Unwired Platform Profile References dialog:

| Action | Description |
|---|---|
| New | Launches the Unwired Platform Profile dialog and create a new profile reference. |
| Edit | Launches the Unwired Platform Profile dialog and edit an existing profile reference. |

| Action | Description |
|---|---|
| Remove | Removes the selected profile reference. |
| Create Profile | Launches the New Unwired Server Connection Profile to create a new connection profile. |
| Help | Launches system help. |

*Editing Unwired Platform Profile References*
Use the Unwired Platform Profile References dialog to edit profile references.

1. Right-click in any page (except the Introduction page) of the Device Application Designer editor, and select **Unwired Platform Profile References**.
2. In the Unwired Platform Profile References dialog, select the profile reference you want to edit from the list and click **Edit**.
3. In the Edit Profile References dialog, edit the information for the profile reference, then click **OK**.

## Creating a New Mobile Business Object Reference

Use the Mobile Business Object References dialog to view, edit, and create new mobile business object references. This allows you to create a screen with a predefined mobile business object as the source, for example, a virtual table.

1. Right-click in any page of the Device Application Designer (except Introduction) and select **Mobile Business Object References**.
2. In the Mobile Object References dialog, click **New**.
3. Click **Search** to find the mobile business object for which you want to create a reference.
4. In the Search for Mobile Business Object dialog, enter your search criteria, and click **Search**.
5. Select the mobile business object from the list, and click **OK**.
6. In the Mobile Business Object References dialog, in Display Name, enter the name that is used to show the device application on the device.
7. (Optional) Select a profile from the Profile drop-down list, or click **New** to create a new Unwired Platform profile.
8. (Optional) Select **Use login profile** to use the login profile. If you select this option, the Profile drop-down is disabled.
9. Click **OK**.

### Mobile Business Object References

Use the Mobile Business Object References dialog to create new mobile business object references, edit existing mobile business object references and attributes, or to remove existing mobile business object (MBO) references.

### Mobile Business Object References

| Column | Description |
|---|---|
| Name | Name of the mobile business object. |
| Display Name | The display name of mobile business object when deployed to the device or simulator. |
| Profile | The Unwired Platform profile associated with the mobile business object. |
| Version | The version of the mobile business object. |
| Package | The package that contains the mobile business object. |
| | **Note:** You must use all the deployed MBOs in a package on the device application. If you do not want to use all the MBOs in one project, you can select only the MBOs you want to use and deploy those to the server, or separate the MBOs you want to use from the MBOs you do not want to use into separate mobile application projects. |
| Synchronization group | The synchronization group that contains the mobile business object. |

### Mobile Business Object References Actions

You can perform these actions in the Mobile Business Objects References dialog:

| Action | Description |
|---|---|
| New | Create a new mobile business object reference. |
| Edit | Edit the mobile business object reference. |

| Action | Description |
|---|---|
| Initial Sync | This option applies to all syncable mobile business objects. It allows different default values to be passed to the mobile business objects with parameters for startup synchronization. You can assign these input types to an initial sync parameter:<br><br>• Variable (only system and user variables can be assigned)<br>• Literal<br>• Personalization key<br>• NULL value |
| Edit Attributes | Edit the display and attribute type settings for the mobile business object attributes. |
| Restore | Restore the mobile business object to the original version. |
| Show Usage | Display information about which controls under which screens are using the selected reference. |
| Remove | Remove the selected mobile business object reference.<br><br>**Note:** Only mobile object references that are not being used in the current application can be removed. |
| Help | Launch the online help. |

*Searching for Mobile Business Objects*
Use the Search for Mobile Business Object dialog to find a mobile business object to assign to a table or synchronize action, or when you create a mobile business object reference.

1. In the Search for Mobile Business Object dialog, define your search criteria and click **Search**.

**Table 9. Search criteria**

| Option | Description |
|---|---|
| Name | Enter the name of the mobile business object. |

| Option | Description |
|---|---|
| Project | Select the project in which you want to search from the drop-down list, or select All Mobile Application Projects to search all mobile application projects. |
| **Advanced** | |
| Data source reference | Filter a search based on data source information; search for items that are bound to a data source, data source type, or connection profile name. |
| Role assignments | Select role mode and enter specific role names. Use wildcards, "*" ; matches all, "?" matches single character. |
| Personalization key reference | Select an existing personalization key from the drop-down list. |
| Synchronization group | Filter based on the synchronization group name used by the mobile business object. |

Search results are displayed.

**2.** Select the mobile business object from the list and click **OK**.

### Editing a Mobile Business Object Reference

Use the Mobile Business Object References dialog to edit the mobile business object reference.

**1.** Right-click in any page (except Introduction) of the Device Application Designer and select **Mobile Business Object References**.

**2.** In the Mobile Business Object References dialog, select the user-defined mobile business object to edit, and click **Edit**.

**3.** In the Mobile Business Object References dialog, edit the information and click **OK**.

### Editing Mobile Business Object Attributes

Use the Mobile Business Object Attributes dialog to edit the name and logical type attributes for a mobile business object.

**1.** Right-click in any page (except the Introduction page) of the Device Application Designer editor, and select **Mobile Business Object References**.

**2.** In the Mobile Business Object References dialog, select the mobile business object reference for which to edit the attributes from the list, and click **Edit**.

To edit more than one attribute, click **Edit Attributes** to open the Mobile Business Object Attributes dialog, where you can edit all of the attributes.

**3.** In the Mobile Business Objects Attributes dialog, select the field to edit:

| Option | Description |
|---|---|
| ID | This cannot be edited. |
| Display Name | Enter the display name of the mobile business object attributes. |
| Logical Type Category | Select the logical type category from the drop-down list. |
| Logical Type | Select the logical type from the drop-down list. |
| Data Type | This cannot be edited. |

**4.** Click **OK**.

### Deleting a Mobile Business Object Reference
Mobile business object references are deleted when you delete the control with which the mobile business object is associated.

**1.** Right-click in any page of the Device Application Designer (except Introduction) and select **Mobile Business Object Reference**.

**2.** In the Mobile Object Reference dialog, select the mobile business object you want to remove and click **Remove**.

You cannot delete mobile business object references that are referenced in the screen design.

**3.** (Optional) Or, in the Screen Design page, right-click the control with which the mobile business object is associated, and select **Delete**.
The mobile business object reference is deleted.

**Note:** When you delete a screen that is associated with a mobile business object in the Flow Design page, the mobile business object reference is not deleted. You can manually delete the mobile business object reference using the Mobile Business References dialog. Additionally, when you verify the device application, the unused mobile business object warning displays in the Problems view.

### Configuring the Initial Synchronization Parameter
Configure an initial synchronization parameter for a mobile business object (MBO).

The initial synchronization parameter determines what is sent to the server when the data is synchronized the first time the application is launched. If you do not set the initial synchronization parameter in the Device Application Designer, then the settings specified in the mobile business object are used.

Setting the initial synchronization parameter for the device application is useful when a mobile business object contains a lot of data so instead of returning all data during the initial synchronization, you can filter records so only the most relevant data is returned. This can make the device application data load much faster.

**Note:** The initial synchronization parameter is not supported on Windows Mobile.

1. Right-click in any page (except Introduction) of the Device Application Designer and select **Mobile Business Object References**.
2. In the Mobile Business Object References dialog, select the user-defined mobile business object for which you want to configure an initial synchronization parameter, and click **Initial Sync**.
3. (Optional) In the Configure Initial Synchronization Parameter dialog, unselect **Use Server Default Configuration for Startup Synchronization**. When this is selected, you cannot configure an initial synchronization parameter and the server default configuration is used for the startup synchronization.

   This is enabled only if the MBO has a synchronization parameter defined. Once the user explicitly sets synchronization parameter values, the initial synchronization paramemeter settings no long take effect.
4. Select the Input parameter to configure and click **Configure**.
5. In the Input dialog, select from these options:

| Option | Description |
|---|---|
| Variable | Select and click **Browse** to browse for available system or user variables to associate with the parameter. Click **New** to define a new variable to associate with the operation parameter. |
| Literal | Select to use a fixed value. For the calendar (date/time/datetime) data type, click on **Select** to choose a value. |
| Personalization key | This is not enabled. |
| NULL | Use a NULL value assigned to the parameter. This does not override a personalization key value or a parameter default value. |

6. Click **OK**.

## Creating Variable References
Use the Variable References dialog to create a new variable to which to assign a control.

1. Right-click in any Device Application Designer page (except the Introduction Page) and select **Variable References**.
   The Variable References dialog is launched.
2. Click **New**.
   The Variables dialog launches.
3. Enter:

| Option | Description |
|---|---|
| Name | Name of the variable reference. |
| Data Type | Select the data type for the variable reference. |
| Default Value | Default value for the variable reference. |
| Description | Description for the variable reference. |

**4.** Click **OK**.

### Variables

Use the Device Application Designer editor to assign controls to get their initial values from variables.

You can assign user or system variables to controls. You can also define user variables that can store values from controls and be reused.

### Personalization key variables

This type of variable allows you to assign personalization keys to controls like edit box, choice, and so on. Personalization keys are maintained in the mobile business object. You can use only the personalization keys defined in Unwired Platform projects referenced by the mobile business object application.

**Note:** Authentication personalization keys, for example user name and passoword personalization keys, and structured personalization keys are not supported.

### Table context variables

When you associate a mobile business object to a table widget or list detail, the Device Application Designer automatically creates a list of table context variables for all of the columns in the mobile business object. These variables are listed in the Variable References dialog and the Variable Selection dialog.

You can see table context variables throughout the workbench displayed in this format: {mbo-name}.{attribute-display-name}. For example:

**Figure 2: Table context variables**
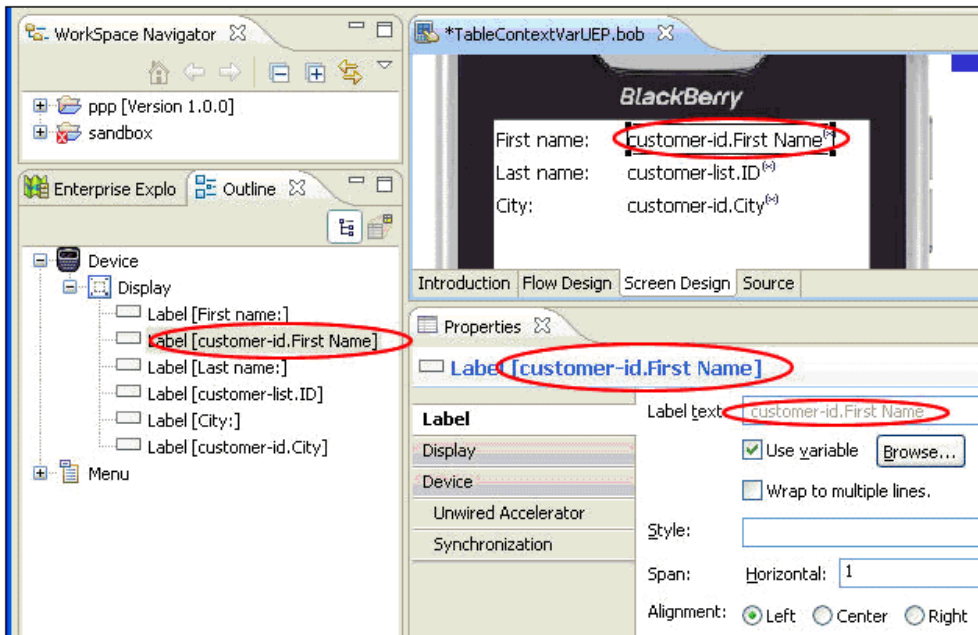


*Filtering Variable References*

Use the Variable References dialog to filter your search for a specific variable reference.

1. Right-click in any page (except the Introduction page) of the Device Application Designer, and select **Variable References**.

2. In the Variable References dialog, select the type of variable you want to specify:
   - User variable
   - System variable
   - Personalization key variable
   - Table context variable

3. (Optional) In Mobile Application Project, for the table context variable type, enter the project in which the variable is defined.

   This field accepts wildcard characters, for example, "*" for any string and "?" for any character. This field is enabled only when you select the table context or personalization key variable option in Show.

4. (Optional) In Mobile Business Object, for the table context variable type, enter the mobile business object in which the variable is defined.

   This field accepts wildcard characters, for example, "*" for any string and "?" for any character. This field is enabled only when you select the Table Context Variable option in Show.

### Variable References
Use the Variable References dialog to view, create, edit, and remove variables.

### Variable references filtering
The Limit To section of the Variable References dialog contains these options for filtering the variable references:

| Option | Description |
|---|---|
| Show | Select the type of variable you want to filter:<br><br>• User variable<br>• System variable<br>• Personalization key variable<br><br>**Note:** You can define a personalization key variable that is not bound to a data source parameter.<br><br>• Table context variable |
| Mobile Application Project | The mobile application project in which the variable is defined<br><br>**Note:** This field is enabled only when you select the table context or personalization key variable option in Show. |
| Mobile Business Object | The mobile business object in which the variable is defined<br><br>**Note:** This field is enabled only when you select the table context variable option in Show. |

### Variable References table

| Column | Description |
|---|---|
| Name | Name of the variable |
| Type | System, user, or table context |
| Data Type | The data type of the variable |
| Mobile Application Project | The mobile application project in which the table context or personalization key variable reference is defined |

| Column | Description |
|---|---|
| Mobile Business Object | The mobile business object in which the variable is defined |
| Initial Value | The initial value assigned to the variable reference |
| Description | The variable reference description |

*Variable References Actions*

You can perform these actions in the Variable References dialog:

| Action | Description |
|---|---|
| New | Create a new variable reference. |
| Edit | Edit an existing variable reference. |
| Show Usage | Display information about which controls or menus are using the selected variable reference. |
| Remove | Remove the selected user variable reference. |
| Help | Displays help. |

*Editing Variable References*

Use the Variable References dialog to edit variables.

1. Right-click in any page (except the Introduction page) of the Device Application Designer editor, and select **Variable References**.
2. Select the user or table context variable reference you want to edit from the list and click **Edit**.

   **Note:** You cannot edit system variables.

3. Edit the variable reference name, default value, and description for user variables, or logical type and display name for table context variables, then click **OK**.

## Creating Image References

Drag and drop an image from your workspace onto controls that support image references, or use the Images dialog to create new, user-defined image references to associate with controls.

1. Right-click in any page (except Introduction) of the Device Application Designer and select **Image References**.
2. In the Image References dialog, click **New**.
3. In the Images dialog, select the location of the image, and click **Browse**:

- Workspace – brings up a file selection dialog that shows only the projects and folders of your current workspace. This is useful for source code control versions since the images always have the same relative path in the same projects as the Device Application Designer.

  **Note:** This option is not supported for the Windows Mobile platform.

- Desktop – brings up the desktop file selection so you can select any file on your machine.

  The File field is filled in automatically based on the file you select. This name is the reference point for the image from the code.

4. In **Name**, the name of the image file you selected is filled in by default. You can enter a different name to be used internally for the image.

   **Note:** Image names must be unique.

5. (Optional) In **Description**, enter a description for the image.
6. Click **OK**.

### Editing Image References

Use the Image References dialog to edit user-defined image references.

1. Right-click in any page (except Introduction) of the Device Application Designer and select **Image References**.
2. (Optional) In the Image References dialog, unselect **System image** to filter out system images that cannot be edited.
3. Select the user-defined image reference you want to edit from the list, and click **Edit**.
4. In the Images dialog, edit the information, and click **OK**.

### Assigning an Image to a Control

Assign an image to a control.

You can assign an image to these controls:

- Tab panel
- Button
- List item
- Image
- Toolbar item

You can also assign an image to the Tab folder and Display as a background image.

**Note:** For Windows Mobile and BlackBerry devices, the image you are assigning cannot be larger than the display area.

1. Drag and drop the image from your WorkSpace to the control or display area.

---

**2.** Alternatively, in the Properties page for the control, next to **Assign image** or **Background image**, click **Browse**.

**3.** In the Image Selection dialog, select the image you want to assign to the control and click **OK**.

## Using Linked Parameters

Linked parameters allow you to set up chained relationships among field and parameter values, which makes it easier to create and update device applications.

### Linked Parameters

Linked parameters use the dynamic format of the mobile business object to drive the content of the drop-down values on the device application.

Many applications have drop-down lists of values you can choose, and the values you select can affect the choices in other lists.

Linked parameters make it easier to build entry forms for device applications that allow dependent drop-down choices. For example, if a user selects "California" for the state field, the country field is automatically set to "USA." This makes the input data consistent, with less chance of errors. For example:

| Field name | Values |
|---|---|
| Field 1 | Accepts values A, B, or C |
| Field 2 | Allows values dependent on Field 1:<br><br>`A->1,2,3`<br>`B->4, 5, 6`<br>`C->7, 8, 9` |
| Field 3 | Allows values dependent on Field 1:<br><br>`A->100, 101, 102`<br>`B->200, 201, 202`<br>`C->300, 301, 302` |
| Field 4 | Allows values dependent on Field 2:<br><br>`1->11,111`<br>`2->22,222`<br>`3->33,333`<br>`4->44,444`<br>`5->55,555`<br>`6->66,666`<br>`7->77,777`<br>`8->88,888`<br>`9->99,999` |

When the user selects value "A" in Field 1, the allowed values listed in Field 2 change to 1, 2, 3; the Field 3 choices become 100, 101, 102.

**Note:** For each change to Field 2, there is a cascading effect to Field 4.

If the user chooses value C in Field 1, the Field 2 choices become 7, 8, 9; Field 3 choices become 300, 301, 302; Field 4 choices become 77,777, 88,888, 99,999.

### Transforming a Data Source for Linked Parameters

You must transform, or generate, the data source to produce data in the required format for linked parameters.

Sybase Unwired Platform includes two XSLT files (`CategoryTypeItem.xsl` and `RegionSiteDepartment.xsl`) located in `<SUP Installation Directory>\UnwiredPlatform\Unwired_Workspace\samples`, which you can modify and use with Remedy Web Services. Use these XSLT files to build mobile business objects to produce the Category-Type-Item or Region-Site-Department dependent field values (linked parameters) that are common in the Remedy HelpDesk and other applications.

If you use a different application or data source, you can use your own transformation or generation techniques to produce the data in a similar format to that required for linked parameters.

This example shows the CategoryTypeItem XSLT:
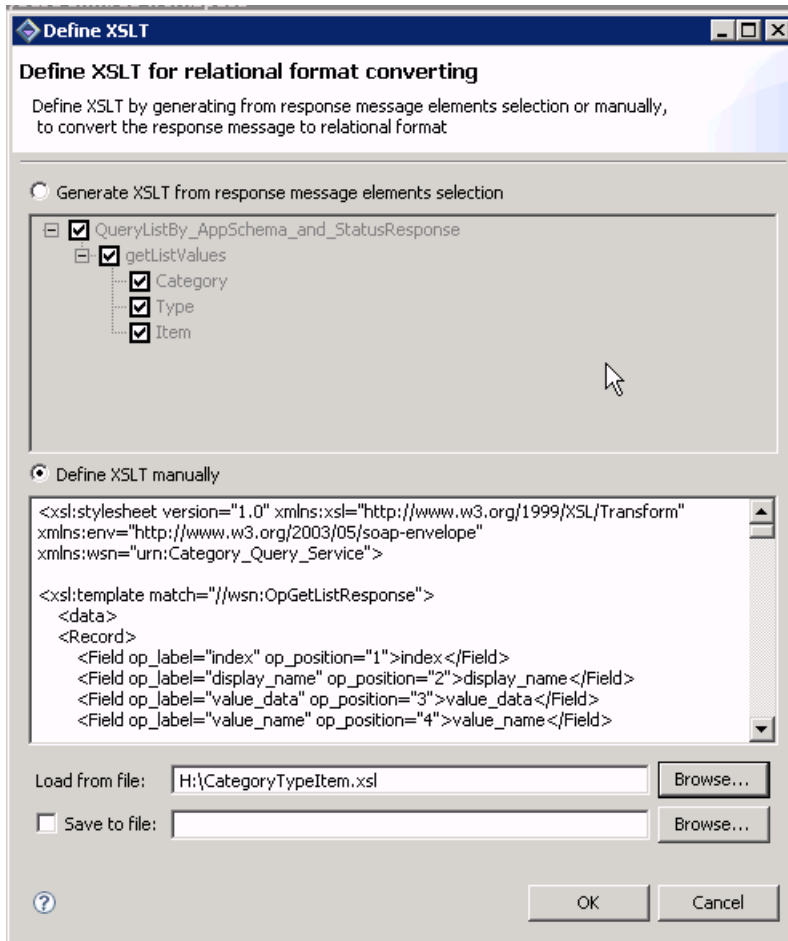
This is a preview of a mobile business object that uses linked parameters:
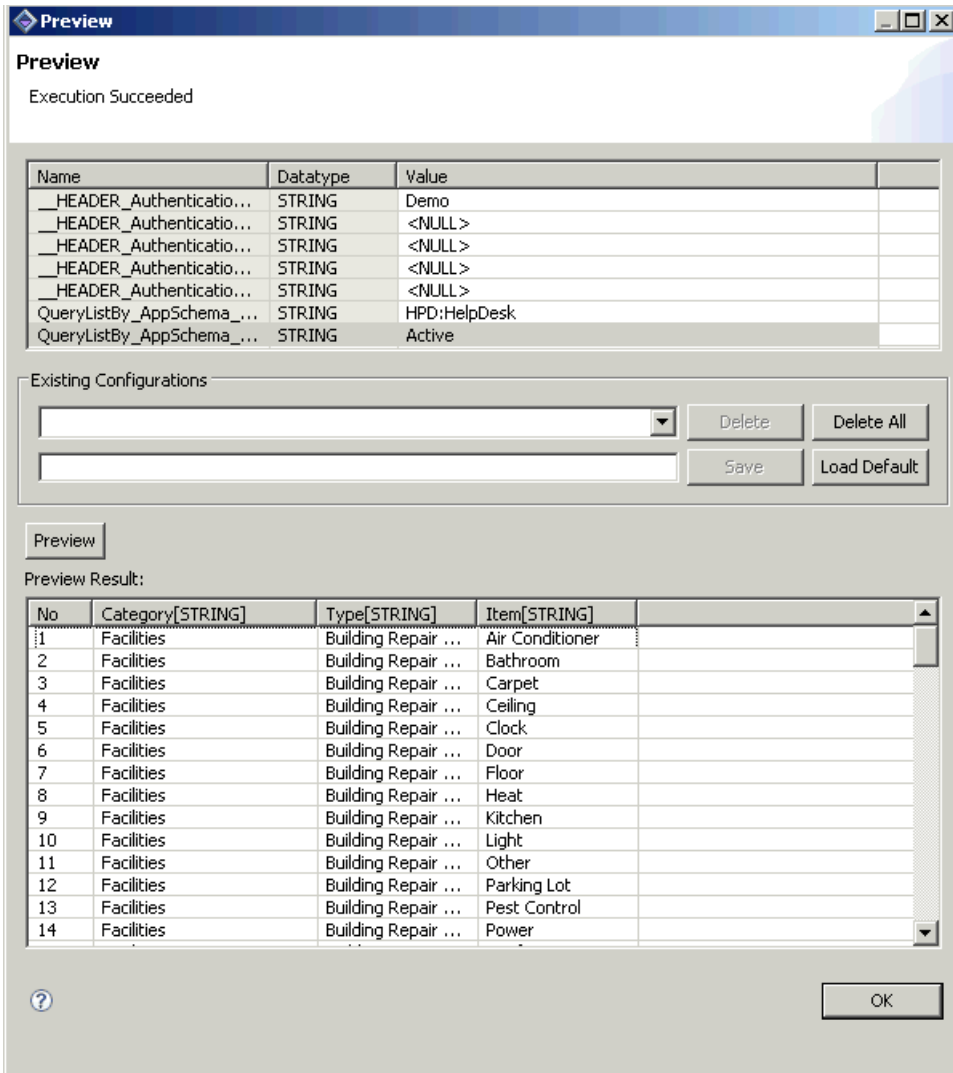
*Example of a database source*

This sample creates a linked table using Remedy data.

```
CREATE TABLE "dba"."linked_params" (
"index" INT NOT NULL,
"display_name" VARCHAR(50) NOT NULL,
"value_data" VARCHAR(50) NOT NULL,
"value_name" VARCHAR(50) NOT NULL,
"link" INT NULL,
)
IN SYSTEM
;
```

```
ALTER TABLE "dba"."linked_params"
    ADD CONSTRAINT "ASA105" PRIMARY KEY CLUSTERED ("index")
;
INSERT INTO dba.linked_params
    ("index",
     "display_name",
     "value_data",
     "value_name",
     "link")
    VALUES
    (1,
    'Category',
    'default',
    'Default',
    null);
INSERT INTO dba.linked_params
    ("index",
    "display_name",
    "value_data",
    "value_name",
    "link")
    VALUES
    (2,
    'Category',
    'hardware',
        'Hardware',
    null);
INSERT INTO dba.linked_params
    ("index",
    "display_name",
        "value_data",
    "value_name",
    "link")
    VALUES
    (3,
    'Category',
    'software',
    'Software',
    null);
INSERT INTO dba.linked_params
("index",
    "display_name",
        "value_data",
    "value_name",
    "link")
    VALUES
    (4,
    'Type',
    'default',
    'Default',
    1);
INSERT INTO dba.linked_params
("index",
    "display_name",
        "value_data",
    "value_name",
```

```
    "link")
   VALUES
    (5,
    'Type',
    'drive',
    'Drive',
    2);
INSERT INTO dba.linked_params
("index",
    "display_name",
       "value_data",
    "value_name",
    "link")
   VALUES
    (6,
    'Type',
    'laptop',
    'Laptop',
    2);
INSERT INTO dba.linked_params
("index",
    "display_name",
       "value_data",
    "value_name",
    "link")
   VALUES
    (7,
    'Type',
    'memory',
    'Memory',
    2);
INSERT INTO dba.linked_params
("index",
    "display_name",
       "value_data",
    "value_name",
    "link")
   VALUES
    (8,
    'Type',
    'email',
    'Email',
    3);
INSERT INTO dba.linked_params
("index",
    "display_name",
       "value_data",
    "value_name",
    "link")
   VALUES
    (9,
    'Type',
    'internet',
    'Internet',
    3);
INSERT INTO dba.linked_params
```

```
("index",
    "display_name",
        "value_data",
    "value_name",
    "link")
 VALUES
 (10,
 'Item',
 'default',
 'Default',
 4);
INSERT INTO dba.linked_params
("index",
    "display_name",
        "value_data",
    "value_name",
    "link")
 VALUES
 (11,
 'Item',
 'dvd drive',
 'DVD Drive',
 5);
INSERT INTO dba.linked_params
("index",
    "display_name",
        "value_data",
    "value_name",
    "link")
 VALUES
 (12,
 'Item',
 'hard drive',
 'Hard Drive',
 5);
INSERT INTO dba.linked_params
("index",
    "display_name",
        "value_data",
    "value_name",
    "link")
 VALUES
 (13,
 'Item',
 'zip drive',
 'Zip Drive',
 5);
INSERT INTO dba.linked_params
("index",
    "display_name",
        "value_data",
    "value_name",
    "link")
 VALUES
 (14,
 'Item',
```

```
    'apple',
    'Apple',
  6);
INSERT INTO dba.linked_params
("index",
    "display_name",
        "value_data",
    "value_name",
    "link")
  VALUES
  (15,
    'Item',
    'deli',
    'Deli',
  6);
INSERT INTO dba.linked_params
("index",
    "display_name",
        "value_data",
    "value_name",
    "link")
  VALUES
  (16,
    'Item',
    'ibm',
    'IBM',
  6);
INSERT INTO dba.linked_params
("index",
    "display_name",
        "value_data",
    "value_name",
    "link")
  VALUES
  (17,
    'Item',
    'memory',
    'Memory',
  7);
INSERT INTO dba.linked_params
("index",
    "display_name",
        "value_data",
    "value_name",
    "link")
  VALUES
  (18,
    'Item',
    'outlook',
    'MS Outlook',
  8);
INSERT INTO dba.linked_params
("index",
    "display_name",
        "value_data",
    "value_name",
```

```
       "link")
    VALUES
    (19,
    'Item',
    'outlook express',
    'MS Outlook Express',
    8);
INSERT INTO dba.linked_params
("index",
    "display_name",
        "value_data",
    "value_name",
    "link")
    VALUES
    (20,
    'Item',
    'access',
    'Access',
    9);
INSERT INTO dba.linked_params
("index",
    "display_name",
        "value_data",
    "value_name",
    "link")
    VALUES
    (21,
    'Item',
    'ftp',
    'FTP',
    9);
INSERT INTO dba.linked_params
("index",
    "display_name",
        "value_data",
    "value_name",
    "link")
    VALUES
    (22,
    'Item',
    'other',
    'Other',
    9);
```

### Creating a Device Application With Linked Parameters

Create a device application with linked parameters.

You can use the same method with different data sources to create a device application with linked parameters. This procedure uses examples using a Remedy Web Service mobile business object as the data source

1. Select **File > New > Device Appication Designer**.
2. When the Device Application Designer opens, click **Flow Design**.

3. In the Palette, select the **Empty** screen, and drag it onto the Flow Design page.

4. Double-click the Empty screen to open the Screen Design page.

5. From the Palette, select **Label** and click on the empty screen in the screen design.

6. From Palette, select **Choice** and click on the empty screen.

The screen will look similar to this:
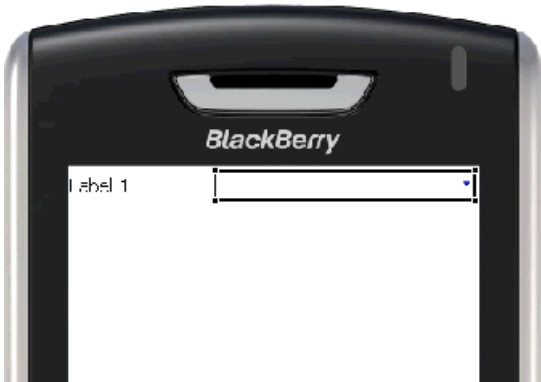


7. From WorkSpace Navigator, select the mobile business object that contains the parameters you want to link and drag and drop it onto the Choice control.
When you drag and drop the data source that contains the required columns for linked parameters, the Display Name Attribute and the Value Attribute are automatically populated. The Properties view for the Choice control displays.

The linked value is cleared and disabled if the mobile business object does not contain the required attributes to do linked parameters.

8. In the Properties view, enter the value for the linked parameters in the Linked Value field.

The value for the Linked Value is the label from the table column. Preview the MBO to get the value for the linked parameter. For example:

The Linked Value can be any or all of these:

* Category
* Type
* Item

For example:
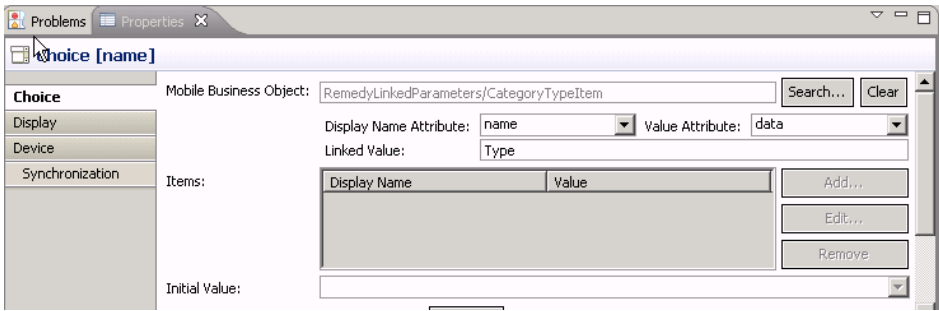
The linked parameter icon  displays on the items with linked parameters.

**9.** Assign a Persist action to each item.

**10.** When you are finished adding linked parameters, select **File > Save**.

## Configuring Stock Screens

You can customize the stock screens originally developed for the container client.

Stock screens show the stock screen icon in the upper left corner of the screen displayed in the display pane on the Screen Design page, and the screen appears gray instead of white, which indicates you cannot configure the screen. You can, however, configure some of the properties of the stock screens and add menu groups to the stock screens.

When you configure the stock screen properties, the configuration dialog box you see depends on the type of the property selected for configuration. For example, if you select a String type property, you will see a String configuration dialog.

**1.** In Flow Design, double-click the stock screen for which to configure the properties. The stock screen opens in the Screen Design page.

**2.** In the Properties view, select the property to configure from the list and click **Configure**.

**3.** Enter the information in the configuration dialog and click **OK**.

### Stock Screens

Stock screens are default screens provided by Unwired Platform that consist of some of the most commonly used device application screens.

| Stock Screen | Function |
|---|---|
| Login | A login screen where the user must enter their user name and password. The login screen is associated with the connection profile chosen as the login profile. |
| Logs | Allows you to view transactional information on insert, update and delete transactions. |

| Stock Screen | Function |
|---|---|
| Pending Operations | This screen lists all operations that have been added but are pending (have not been synchronized). This means that changes do not take place until the next synchronization. You can delete an operation, which means the changes are deferred and will not be sent for that operation when the operations are synchronized.<br><br>**Note:** The pending operations screen does not display complex type parameter values. |
| Personalization | Allows the user to get or set all the personalization keys referenced in the MBO package. A list of personalization keys is returned based on the MBO package used in the device application. |
| Search | The search screen allows you to search synchronized data in mobile business objects on the device.<br><br>**Note:** Binary search is not supported on BlackBerry devices. On BlackBerry devices, in the search screen, all binary columns are filtered out. |
| Synchronize | This screen lists the mobile business objects that are referenced by the device application. The user can select individual mobile business objects, or all mobile business objects, upon which to perform a synchronization action.<br><br>The user can also view information about the selected mobile business object and select from these synchronization options:<br><br>• Sync immediately<br>• Prompt and sync if there is no response<br>• Prompt but do not Sync if there is no response<br>• Do not Prompt and do not sync<br><br>**Note:** The synchronization from the synchronization stock screen does not impact the current display screen of the mobile business object. If another user updates the mobile business object, the user must use the synchronization operation that is directly linked to the mobile business object to get the updated results. |

| Stock Screen | Function |
|---|---|
| BlackBerry Custom Code | Allows you to design custom code for the screen and attach it to the selected platform. The platform plug-in determines how the source file is built. |
| Windows Mobile Custom Code | Allows you to design custom code for the screen and attach it to the selected platform. The platform plug-in determines how the source file is built. |

### Stock and Settings Screen Properties

Use the Properties page in Screen Design to configure stock and settings screen properties.

*Property types*

**Note:** Properties that you can configure for the stock and settings screens are platform-specific so configurable properties vary depending on the platform.

| Type | Description |
|---|---|
| String | Defines the text for stock screen properties. |
| Boolean | Set to true to display, or false to not display, the selected stock screen property. |
| Integer | Defines the integer associated with stock screen properties. |
| Image | Defines the image associated with stock screen properties, for example, a banner. |
| Style | This property defines the display style for stock screen properties, for example labels. |
| Action | Defines the actions for the stock screen property. The configurable actions for stock screen properties are restricted to navigation type actions. For example, you can assign control from the stock screen for persisting or submitting an application. |
| Folder | For the BlakcBerry custom code screen, you can select the folder for where to compile resources, source packages, and third party libraries. |
| File | For the BlackBerry custom code screen, you can select the class file that implements the screen. |
| Option | Sets the default log level for the logging screen. |

Click the button to perform the corresponding action:

| Button | Description |
|---|---|
| Configure | Launches the configuration dialog for the selected property. |
| Customize | Enabled for screens with a custom action. The name of the button varies depending on the name given to the custom action. "Customize" is the default. |
| Create Screen Class | This button appears only for Custom Code stock screens. Launches the Custom Code Screen Properties dialog. |
| Restore | Restores the selected item to its default properties. |
| Restore All | Restores all properties to their default settings. |

### Configuring the Action Property

Use the screen properties dialog to configure action properties for the stock screen.

1. In the stock screen Properties view, select the action you want to configure and click **Configure**.
2. In the property dialog, select the Value from the drop-down list.

   **Note:** If you select the Alert or Connection action, the dialog for creating an alert or connection action launches.

3. Click **OK**.

### Configuring the Boolean Property

Use the screen properties dialog to configure the boolean properties for the stock screen.

1. In the stock screen Properties view, select the boolean property to configure and click **Configure**.
2. In the property dialog, select the check box next to the property to set to `true`.

   Leaving the check box unselected means the property is set to `false`.

3. Click **OK**.

### Configuring the Image Property

Use the screen properties dialog to select the image to associate with the image properties for the stock screen.

1. In the stock screen Properties view, select the image property to configure and click **Configure**.

2. In the Screen Image Property dialog, click **Browse**.

3. In the Image Selection dialog, select the image you want to use for the image property from the list, and click **OK**.

4. Click **OK**.

### Configuring the Integer Property

Use the Screen Properties dialog to configure integer properties for the stock screen.

1. In the stock screen Properties view, select the integer to configure and click **Configure**.

2. In the property dialog, in Value, enter the value for the integer.

3. Click **OK**.

### Configuring the Style Property

Use the screen properties dialog to configure the style properties for the stock screen.

1. In the stock screen Properties view, select the style property to configure and click **Configure**.

2. In the Style dialog, click **Browse**.

3. In the Style Search dialog, select the style you want to use for the style property from the list, and click **OK**.

4. In the Style dialog, click **OK**.

### Configuring the String Property

Use the screen properties dialog to configure the string properties for the stock screen.

1. In the stock screen Properties, select the string property to configure and click **Configure**.

2. In the string dialog, in Value, enter the text for the string and click **OK**.

## Configuring Settings Screens

Configure properties for settings screens.

When you configure the settings screen properties, the configuration dialog box you see depends on the type of the property. For example, if you select a String type property, you see a String configuration dialog.

1. In Flow Design, double-click the settings screen for which to configure the properties. The settings screen opens in the Screen Design page.

2. In the Settings Screen Properties view, select the property to configure and click **Configure**.

3. Use the configuration dialog to make the changes, and click **OK**.

### Settings Screens

Settings screens are default screens provided by Unwired Platform that consist of some of the most commonly used settings screens for device applications.

Only settings screens with settings supported by the selected profile are shown in the palette. Settings screens are not connected to other screens in the Flow Design.

**Note:** Some settings screens take effect only after restarting the application, for example, the Profile settings screen.

| Screen | Description |
|---|---|
| About | Use to design your own "About screen," that can contain information like version, copyright, author and so on. |
| Profile | Use this screen to list and define connection profiles to the data sources. |
| | If you change connection information in the profile settings, such as the domain, server, or port, you must drop and re-create the data source to which the profile points, otherwise, the profile will point to the wrong data source. |
| | **Note:** The Profile settings screen is not supported for messaging-based applications. |
| Push Settings | Use this screen to configure push HTTP notifications. |
| | **Note:** The Push settings screen is not supported on messaging_based applications. |
| Screen Saver | The screen saver screen requires a login screen since it locks out with a password, based on a user's inactivity. |
| | **Note:** The screen saver screen is not supported on the Windows Mobile platform. |

| Screen | Description |
|---|---|
| Logging | Configure the amount of client log you want to see on the client, by setting debug levels to Off, Fatal, Error, Warning, Information, and Debug.<br><br>When set to Off, you do not see the log. The Debug level shows the most amount of information in the client log. It also specifies the label for the default log level.<br><br>It can be configured to remove logs for both the client and server log after exiting the application.<br><br>You can also specify the label for the default log level. |

### Setting the Logging Level for the Client Log

Use the Logging settings screen to configure the log level for the client log generated by the device.

1. In the Screen Design page, click on the Logging screen.
2. In the Properties view for the Logging screen, select the **Default log level** row in the table, and click Configure.
3. Choose the log level and click **OK**.

| Log level | Description |
|---|---|
| Fatal | Shows fatal errors. |
| Error | Shows errors. |
| Warning | Shows warnings. |
| Information | Shows information. |
| Debug | Shows the most information. |
| Off | Turns the log off. |

### *Specifying a Label for the Client Log*

Specify the label for the default log level.

1. In the Screen Design page, click the Logging screen.
2. In the Properties view for the Logging screen, select the **Label for the log level** row in the table, and click **Configure**.
3. Enter the value for the label and click **OK**.

### Configuring the Profile Screen

Configure the labels for the settings that appear on the Profile screen

Configure the labels for settings such as user name, password, server name, server port number, and so on.

1. In the Screen Design page, click on the Profile screen.
2. In the Properties view, select the setting to configure from the list and click **Configure**.
3. In the dialog, configure the property, and click **OK**.

### Configuring the Push Settings Screen

Define the style and configure labels and messages for the Push Settings screen .

Configure the labels and messages that appear on the Push Settings screen, such as MDS server and port, HTTP push label and description, and so on.

1. In the Screen Design page, click on the Push Settings screen.
2. In the Properties view, select the setting to configure from the list and click **Configure**.
3. In the dialog, configure the property, and click **OK**.

#### *Push Synchronization*

Push synchronization is based on the synchronization group. A synchronization group specifies the synchronization behavior for every mobile business object (MBO) within that group.

See *Sybase Unwired WorkSpace - Mobile Business Object Development > Develop > Working with Mobile Business Objects > Mobile Business Object Mobility Properties > Synchronization > Synchronization Groups*.

**Note:** BlackBerry devices support only HTTP push synchronization.

The client application parses the push notification and retrieves the push information, such as which synchronization group under which package should perform the push synchronization. The response type of each synchronization group is persisted on the device. The response types include:

| | |
| --- | --- |
| Sync immediately | Synchronize the corresponding synchronization group immediately when push synchronization notification arrives |

| | |
|---|---|
| Prompt and sync if there is no response | When push synchronization notification arrives, a dialog appears with a "Continue" button and "Cancel" button and displaying a message saying, "There is a push request to refresh a synchronization group. Would you like to proceed?" If there is no response from the user within the specified time interval, synchronize the corresponding synchronization group. |
| Prompt but don't sync if there is no response | When push synchronization notification arrives, a dialog appears with a "Continue" button and "Cancel" button and displaying a message saying, "There is a push request to refresh a synchronization group. Would you like to proceed?" If there is no response from the user within the specified time interval, the corresponding synchronization group is not synchronized. |
| Don't prompt and don't sync | When push synchronization notification arrives, there is no prompt and no synchronization. |

The time interval between data change in the back-end data source and push notification arrival on the device is dependent on:

- The scheduled cache interval set in the cache policy.
- The synchronization group's change detection interval. After the check interval, a scheduled task finds clients that need to be notified of the change.
- The notification interval set in the Synchronization Group Info screen on the BlackBerry device.

These three things dictate how long the device will receive a push notification.

### *Enabling Push Synchronization on BlackBerry*
This is an example of the general procedure for configuring push synchronization on BlackBerry.

**Prerequisites**
The mobile business object (MBO) used for the device application must be associated with a cache group for which the cache policy is set to "scheduled." See *Sybase Unwired WorkSpace - Mobile Business Object Development > Develop > Working with Mobile Business Objects > Mobile Business Object Mobility Properties > Cache Groups > Creating Cache Groups* and *Sybase Unwired WorkSpace - Mobile Business Object Development > Develop > Working with Mobile Business Objects > Mobile Business Object Mobility Properties > Cache Groups > Assigning Mobile Business Objects to a Cache Group.*

**Task**

1. Right-click the Mobile Application Project for which to create the device application and select **Create Device Application Designer**.
2. Select BlackBerry as the device platform.

   The screens and connections for the application are created automatically.
3. From the Flow Design Palette, drag and drop the **Push Settings** screen onto the **Settings Screens** box.
4. Verify and generate the code for the device application.
5. On the Simulator or device, go the Push Settings screen in the BlackBerry Options menu.
6. Enter the MDS host and port number and save the information.
7. Open the application, then open the **Synchronize** screen.
8. Select the synchronization group for which you want to enable push, then select **Synchronization Group Info**.
9. Select the **Push enabled** option and enter the configuration settings.
10. Save the push settings.
11. Synchronize the synchronization group on the BlackBerry device to register the push synchronization settings with the server.

    After synchronization, the client can receive the push notification when data changes in the server database.

## Configuring the About Screen
Configure the settings for the About screen.

Set properties to configure settings such as style, the amount of allocated memory in RAM, device information, and so on.

1. In the Screen Design page, click on the Screen Saver screen.
2. In the Properties view, select the setting to configure from the list and click **Configure**.
3. In the dialog, configure the property, and click **OK**.

## Configuring the Screen Saver Screen
Configure the settings for the Screen Saver screen.

Set properties for the Screen Saver screen such as allowing the user to change the idle time, style, button labels, and so on.

1. In the Screen Design page, click on the Screen Saver screen.
2. In the Properties view, select the setting to configure from the list and click **Configure**.
3. In the dialog, configure the property, and click **OK**.

### Accessing the Settings Screen on the Device

How to access settings screens depends on which device you are using.

On Windows Mobile devices, a **Settings** item is added to the application's Menu. In the mobile application, select **Menu > Settings**.



On BlackBerry devices, access the settings screens in the BlackBerry **Options** menu. The settings screen appears in Options as the name of the device application, for example **custinfo**.
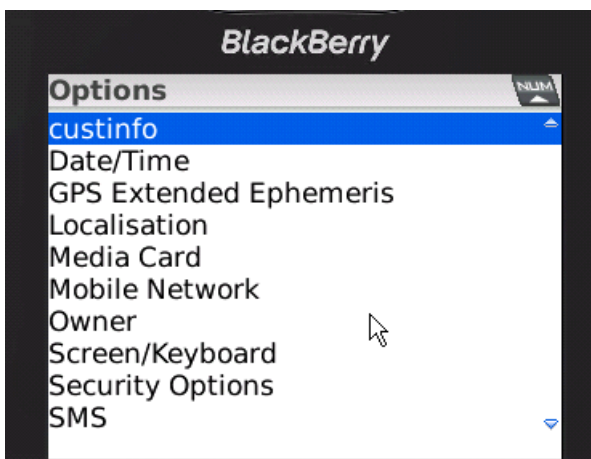
## Designing a BlackBerry Custom Code Screen

Use the BlackBerry Custom Code stock screen to design device application screens with custom coding.

1. Create a new device application.
2. Select BlackBerry as the device platform. This determines which stock screens appear.
3. Click **Flow Design**.
4. From **Palette > Stock Screens**, select **BlackBerry Custom Code**.
5. Drag and drop the BlackBerry Custom Code screen onto the Flow Design page.
6. In the Screen Design Properties view, click **Stock Screen**. The available properties are determined by the device platform you selected.

   You can also drag and drop a source file that implements the correct screen interface. This creates the stock screen item in the flow and automatically sets up all the fields.
7. Click **Create Screen Class** to create a new Java class for the custom screen.

### Creating a Java Screen Class

Use the Custom Code Screen Properties dialog to create a new screen class.

1. In the Flow Design page, double-click the **BlackBerry Custom Code** screen.
2. Right-click in the Screen Design page and select **Create Screen Class**.

   You can also click **Create Screen Class** in the Screen Design Properties view when the custom code screen is selected.
3. In the New Java Class wizard, enter:

| Option | Description |
|---|---|
| Source Folder | Click **Browse** to locate the source folder for the Java class. |
| Package | Click **Browse** to locate the Package for the new Java class. |
| | **Note:** Do not leave this field blank, as the JDK 1.4 Java class in the default package cannot be resolved in other packages. |
| Name | Enter the name of the Java class. |

| Option | Description |
|---|---|
| Superclass | The default is com.sybase.uep.bob-client.screens.BaseBOBScreen.<br>1. Click **Browse**.<br>2. In the Superclass Selection dialog, enter:<br>   • Choose a Type<br>   • Matching Items<br>3. Click **OK**. |
| Interfaces | The default is com.sybase.uep.bob-client.screens.IBOBScreen.<br>1. Click **Add**.<br>2. In the Implemented Interfaces Selection dialog, enter:<br>   • Choose Interfaces<br>   • Matching Items<br>3. Click **OK**.<br><br>To remove an interface:<br>1. In the Implemented Interfaces Selection dialog, select an interface.<br>2. Click **Remove**. |
| Which Method Stubs Would You Like to Create | Select from:<br>• Public Static Void Main – leave this option unselected.<br>• Constructors From Superclass<br>• Inherited Abstract Methods |
| Do You Want to Add Comments | Select **Generate Comments** to add comments. |

**4.** Click **Finish**.

## Configuring BlackBerry Custom Code Screen Properties

Use the BlackBerry Custom Code screen properties dialog to configure properties for the Custom Code screen.

**1.** In Flow Design, double-click the **BlackBerry Custom Code** screen.

**2.** In the **Properties > Screen Design > Stock Screen** view, select the property to configure and click **Configure**.

| Property | Description |
|---|---|
| Class | Click Browse to select the java file that implements the screen. |

| Property | Description |
| --- | --- |
| Extra source package | Select the folder that contains any additional source package to compile into the client. |
| Resources | Select a folder to include any resources, for example, images, in the client. |
| Source package | Select the folder to include any of the source packages to compile into the client for the source screen. Generally this property is set automatically when you select the source file. |
| Third party libraries | Select a folder with libraries (*.jar files) to compile against. |

3. A configuration dialog window appears for each property you select. Enter the information in the dialog window, and click **OK**.

4. You can restore the default settings for all custom code screen properties by clicking **Restore All**. Or, you can restore the default settings for selected properties by selecting the property and clicking **Restore**.

5. Click **Create Screen Class** to create a new Java class for the custom screen.

## Editing Device Applications

Use the Device Application Designer to edit device applications.

1. In WorkSpace Navigator, open the project that contains the device application you want to edit, right-click the device application, and select **Open With > Device Application Designer**.
The Device Application Designer opens to the Introduction page.

2. Click the **Flow Design** or **Screen Design** tab to make your changes.

3. Select **File > Save**.

## Deleting Device Applications

Delete a device application using the Device Application Designer.

1. In WorkSpace Navigator, open the project that contains the device application you want to delete.

2. Right-click the device application you want to delete and select **Delete**.

3. Click **Yes** in the confirmation dialog to proceed with the deletion.

## i18n Support

Sybase Unwired Platform i18n support includes the following functionality.

- Sybase Unwired Platform allows entry of international data at runtime for mobile applications.
- Sybase Unwired Platform correctly displays international data.
- Sybase Unwired Platform correctly stores and retrieves international data from data sources.
- All Sybase Unwired Platform user interfaces, graphical or command line, are localizable except the CLIs for Unwired Server.
- Formatting of data (for presentation to the user) is locale sensitive.
- GUI layouts will accommodate localized data so there is no overlap.
- International data is processed correctly by Sybase Unwired Platform, for example, parsing of strings.
- Transformation of data is handled correctly by Sybase Unwired Platform. For example, converting characters encoded as cp936 to Unicode UTF-8.
- File input and output correctly handles international data for the contents of files as well as the directory paths to the files.
- All online help content, such as tool tips and cheat sheets, is localizable.
- All data conduits in Sybase Unwired Platform correctly transport international data.

### Localization

Generate localized device applications by localizing objects in the Device Application Designer.

The Device Application Designer contains various items that you can localize, such as the names of screen controls, screens, and alerts. You can localize the device application by creating locale properties files at the Localization Property tab. You can then load, update, and generate localized device applications.

You can update resource bundles for the Device Application Designer by editing the values in the localization properties file. The resource keys are defined by the corresponding elements in the current Device Application Designer. You can also upload specific locale resource bundles into the Device Application Designer.

Some limitations for the locale properties files apply:

- In controls, only the literal-type of label item is localizable. When creating a new locale for a control, a resource bundle file with an entry similar to this example is created for the literal string:
  Screen.1.CellTable.1.CellLabel.6.LabelItemLiteral.0=Hello
- When you specify a country for the language, the basic language locale must also be available. For example, if you create a locale and specify English as the language and the

United States as the country, then a locale for English (the basic language) must also be available.

- If you create a locale that specifies language, country, and variant, the locale for the basic language and the locale for the basic language and the country must be available. For example, if you create a locale and specify English as the language, United States as the country, and WIN as the variant, then English (United States) and English locales must also be available.
- When you select the language and country from the drop-downs that list the full names, the codes in the property file are generated according to the language and country selected.

All locale properties files are saved in the same directory as the device application.

When localizing the Device Application Designer:

- Device application files that have names that begin with numbers or special characters cannot be localized. You will receive an error when you generate the code. Make sure that any device application you want to localize does not have a file name that begins with a number or special character.
- If you localize the locale properties file manually with an external editor, you must make sure the file is encoded in ASCII so that the content can be correctly read and converted to Unicode.
- When you localize a device application containing controls cannot have duplicate names, otherwise, controls that are named identically do not behave correctly. For example, if one of the radio buttons in the original device application is selected, the other radio button with the same name is also selected.

### *Creating a New Locale*
Create a new locale with resource keys and values extracted from localizable strings in the current device application file.

When you create a new locale, keep in mind:

- When you specify a country for the language, the basic language locale must also be available. For example, if you create a locale and specify English as the language, then there must also be a locale for English (the basic language).
- If you create a locale that specifies language, country, and variant, the locale for the basic language and the locale for the basic language and the country must be available. For example, if you create a locale and specify English as the language, United States as the country, and WIN as the variant, then English (United States) and English locales must also be available.

1. In the Properties view, click **Localization** in the left pane.
2. In the Localization page, click **New**.
3. In the New Locale wizard:

| Option | Description |
|---|---|
| Language | Select the language. |
| Country | Select the country. |
| Variant | Enter the variant, which is the vendor or browser-specific code. For example, enter "WIN" for Windows, "MAC" for Macintosh and "POSIX" for POSIX. If there are two variants, separate them with an underscore, and put the most important one first. For example, a Traditional Spanish collation might construct a locale with parameters for language, country, and variant as: "es", "ES", "Traditional_WIN". |
| Overwrite existing file | Overwrite an existing localization file. |
| Automatically create default locale | Automatically create the default locale properties file. For example, if you specify the language as "English" and the country as the "United States" for a device application called `test`, then both `test_en_uS.properties` and `test.properties` files are created. |

**4.** Click **Finish**.
The properties file is created in the same directory as the device application file.

### Editing a Locale Properties File
Use the Properties File Editor to edit the localization properties file.

**1.** In the Screen Design page Properties view, click **Localization**.

**2.** Select the locale to edit and click **Edit**.

**Note:** If you use an external editor to manually edit the locale properties file, make sure the file is encoded in ASCII so that the content can be correctly read and converted to Unicode.

**3.** Make the changes to the file.

**4.** Select **File > Save**.

### Updating the Current Locale
Update the currently loaded locale properties file with the resource keys from the current Device Application Designer.

If the locale properties file does not already exist, it is created. If the current locale is not defined in the device application file, the updated locale is used as the default, and the file name is *{device_application}*.properties. Otherwise, the locale defined in the device application file is updated.

1. In the Screen Design page Properties view, click **Localization**.
2. Click **Update**.

### *Validating the Locale Properties Files*
Validate locale properties files.

Based on Device Application Designer Errors/Warnings General preference settings, the validation verifies that all the resource keys for the current device application file exist in the locale properties file.

1. In the Screen Design page Properties view, click **Localization**.
2. Select the locale for which to validate the properties file, and click **Validate**.
   You see a message if there are missing resource keys.
3. (Optional) Click **Yes** to automatically fix the properties file.

### *Removing a Locale*
Remove locale properties files.

1. In the Screen Design page Properties view, click **Localization**.
2. Select the locale to remove and click **Remove**.
3. Click **Yes** to confirm the deletion.

### *Loading a Locale Properties File*
Load a locale properties file.

1. In the Screen Design page Properties view, click **Localization**.
2. Select the locale properties to load and click **Load**.
   You see a message if there are missing resource keys.
3. (Optional) Click **Yes** to automatically fix the properties file.
   If you choose to not fix the locale properties file, loading does not continue.

   Loading the locale property file updates the current locale of the device application file.

### *Localization Properties File*
This is an example of the localization properties file.

The localization properties file is encoded in standard ASCII. All non-ASCII character values from the device application file are converted to escaped Unicode hexadecimal values before they are written to the properties files. The resource keys in the properties file are grouped into four sections:

- Global – anything that is global in the device application file. Currently, the ribbon name and platform-specific properties.
- Flow – the flow of the screens.
- Screen – the screens contains widgets, menu items, and actions.

• Meta – metadata information such as variables, images, styles, connection profiles, the synchronization progress message, and mobile business objects.

## Syntax

Variable assignment:

```
SettingsScreen.
16.StockProperties.com.sybase.uep.bob.PROFILE_UPDATE_SCREEN_USERNAM
E_INPUT_LABEL.value=User Name
SettingsScreen.
16.StockProperties.com.sybase.uep.bob.rim.PROFILE_SCREEN_CLOSE_MENU
_LABEL.value=Close
SettingsScreen.
16.StockProperties.com.sybase.uep.bob.rim.PROFILE_SCREEN_EDIT_PROFI
LE_MENU_LABEL.value=Edit profile
SettingsScreen.
16.StockProperties.com.sybase.uep.bob.rim.PROFILE_SCREEN_ERROR_SAVI
NG_PROFILE_MSG.value=Profiles could not be saved due to an internal
error
SettingsScreen.
16.StockProperties.com.sybase.uep.bob.rim.PROFILE_SCREEN_LOGIN_PROF
ILE_MENU_LABEL.value=Set as Login
SettingsScreen.
16.StockProperties.com.sybase.uep.bob.rim.PROFILE_UPDATE_SCREEN_CAN
CEL_MENU_LABEL.value=Cancel
SettingsScreen.
16.StockProperties.com.sybase.uep.bob.rim.PROFILE_UPDATE_SCREEN_ONL
INE_LOGIN_ERROR_MSG.value=Encountered error while authenticating the
new changes to the profile :
SettingsScreen.
16.StockProperties.com.sybase.uep.bob.rim.PROFILE_UPDATE_SCREEN_SAV
E_MENU_LABEL.value=Save
SettingsScreen.
16.StockProperties.com.sybase.uep.bob.rim.PROFILE_UPDATE_SCREEN_SAV
E_PROFILE_SUCCESS_SUGGESTION_MSG.value=Profile info modification may
cause data synchronization exception, please clear data in
synchronization screen first.
SettingsScreen.
16.StockProperties.com.sybase.uep.bob.rim.PROFILE_UPDATE_SCREEN_TIT
LE.value=Profile Update
SettingsScreen.
16.StockProperties.com.sybase.uep.bob.rim.PROFILE_UPDATE_SCREEN_UPD
ATE_DUPLICATE_PROFILE_NAME_MSG.value=The profile name you have
chosen is already in use
SettingsScreen.
16.StockProperties.com.sybase.uep.bob.rim.PROFILE_UPDATE_SCREEN_UPD
ATE_VALIDATING_MSG.value=Validating...
SettingsScreen.
16.StockProperties.com.sybase.uep.bob.wm.PROFILE_SCREEN_CLOSE_MENU_
LABEL.value=Close
SettingsScreen.
16.StockProperties.com.sybase.uep.bob.wm.PROFILE_SCREEN_EDIT_PROFIL
E_MENU_LABEL.value=Edit Profile
```

### Examples

- **Ribbon name** – The resource key for the ribbon name must be defined as ribbon_name:

  ```
  ribbon_name=myBob
  ```
- **Alerts** – Currently only alerts in flows contain localizable strings. Resource keys for an alert can be defined as {ConnectionType}.{sourceScreen}.{targetScreen}. {connectionSubType}.{attrName}. For example, the OK label of an information alert from screen 1 to screen 2 looks like:

  ```
  ALERT.Screen.1.Screen.2.INFO.ok_label
  ```
- **Screen widgets** – Resource keys for screen widgets can be defined as {screen}.{widget}. {attributeName}. For example, a button with the ID of 1 in screen 1 looks like:

  ```
  Screen.1.Button.1.name
  ```
- **Stock properties** – For stock properties, the keys look similar to:

  ```
  {screen}.StockProperties.{stockProperty} .{attributeName }
  ```
- **Labels** – The resource key for the About screen close menu label can be defined as:

  ```
  Screen.1.StockProperties.
  {platformid}.ABOUT_SCREEN_CLOSE_MENU_LABEL.value
  ```
- **Menus** – Resource keys for menus can be defined like this:

  ```
  {screen}.{menuItem}.{attributeName }  e.g. Screen.1.Menu.2.name
  ```
- **Variables** – Resource keys for variables are formatted as {variable}.{attributeName}, for example:

  ```
  VariableReferences.1.display_name, VariableReferences.
  1.description
  ```

### *Localization Limitations*

This topic describes the limitations of device application localization.

- The screen title and menu of the Push Logs screen cannot change language (for example, Chinese) after selecting a new device language even though the value in the localization file itself has been changed.
- Some properties are not included in the localization file, so some characters that should be localized on a BlackBerry application, are not. For example:
  - Mobile Application Object (MBO) package name in the Synchronize screen
  - Synchronization Group name in the Synchronize screen
  - Continue and No buttons in the Push Request screen
  - Table header text (Time, Status) in Push Logs screen
  - Save message in Logs setting screen
  - Some client log menus in the Logs screen
  - Some error messages are not localized
  - In the Push Logs screen, when you delete the push logs, the pop-up message confirming the deletion is not localized.
- On BlackBerry device applications, you cannot change the language for the HTTP MDS server name in the Push Settings screen after selecting the device language, for example

---

Chinese, even though the MDS server name property value (SettingsScreen. 12.StockProperties.com.sybase.uep.bob.rim.PUSH_SETTINGS_SC REEN_HTTP_MDS_SERVER_NAME.value=localhost) has been changed in the localization properties file.

- On BlackBerry devices, for the date/time/datetime picker screen, there are no properties for the message in the pick screen in the localization properties file.
- Using the PIM action to perform a write event on **Tasks > Priority/Completed**, the localized value cannot be written correctly. Using the PIM action to perform a read event on **Tasks > Priority/Completed**, the localized value can be read only as English.

## Verifying a Device Application

Before generating the device application, verify it and fix any errors that are found.

1. In WorkSpace Navigator, open the project that contains the device application for which you want to generate code, right-click the device application and select **Verify Device Application Designer**.

   **Note:** If there are errors, you must fix them before you can generate the device application. Check the Problems view for errors and warnings.

2. Or, click on the Verify icon ▣ in the Device Application Designer toolbar. You can also right-click on any page of the Device Application Designer (except the Introduction page) and select **Verify**.

### Using Content Assist to Identify and Fix Problems

Content assist identifies and displays a list of problems when you verify your device application. You can access content assist from any decorator item.

1. Right-click in the Flow Design or Screen Design display pane, and select **Verify** from the context menu.

   A list of problems is generated.

2. In the Problems view, double-click the problem.
   The element with the problem is displayed. The cursor changes to the content assist cursor when you move over the problem.

3. Click the problem decorator in the lower left-hand corner of the object. For example, ▣ the error decorator.
   A content assist window appears and gives you available options (with tool-tips) for fixing the problem.

4. Select the option for the action you want to perform.

   **Note:** You cannot generate a device application if there are errors. If you want to generate the device application with the errors, you can change the error to a warning in the Error and Warning Preferences.

### *Content Assist Warning and Error Decorators*
Use Content Assist to identify problems and find options for fixing them.

Content assist displays warning and error decorators in the lower left corner of most objects in the Device Application Designer editor Screen Design and Flow Design pages. Clicking on the decorator displays options for fixing the problem, as well as a description of the problem. The problems are linked to the Problems view. The objects show decorators based on the problem reported. Warning and error decorators are used on the object. The tool-tip is the actual message in the problems view.
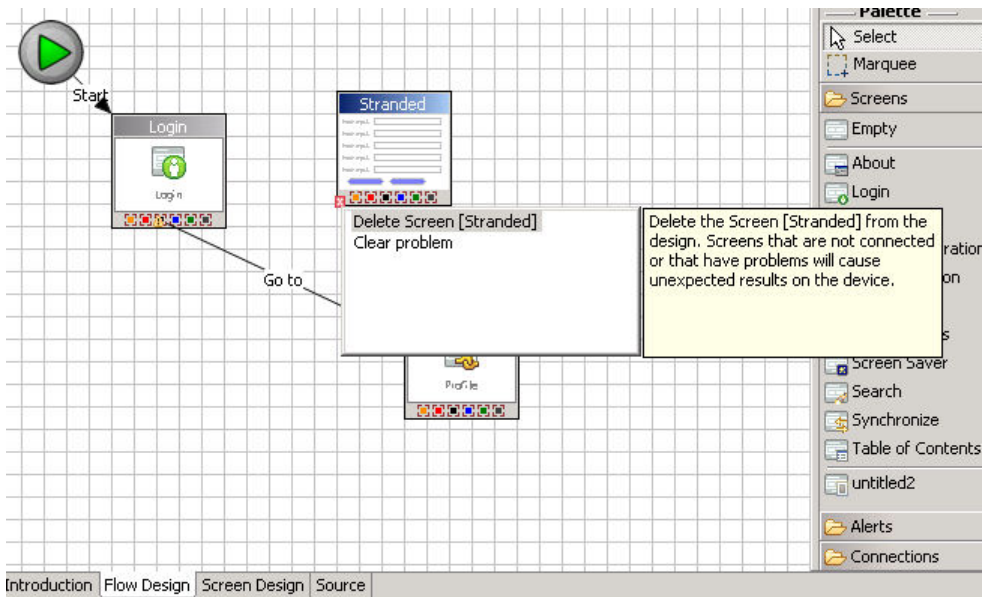


**Note:** Clearing the problem clears it only until the next Verify is performed. This is useful when you are finished with a model and don't want the decorators in the project files under the navigator.

One of the more advanced flow design content assists creates menu items and actions for unused connections and alerts. The more advanced content assist for the screen design creates the login screen (if you have a logout action in use), or creates an action for a control that requires an action. This also shows up in the right-click context menu on a connection not in use before the Verify is performed.

## Generating Device Application Code

Use the Device Application Designer Code Generation wizard to generate and compile code for Windows Mobile and BlackBerry device applications.

**Note:** Before generating code and deploying the application for testing on a device or simulator, you must deploy the mobile business objects referenced in the device application, to the Unwired Server. All the mobile business objects in the deployed package must be used in the device application.

### Generating Code For a BlackBerry Device Application

Use the Generate Device Application wizard to generate the device application code.

#### Prerequisites

Verify the device application and fix any errors that are found. Device applications with errors cannot be generated.

#### Task

1. Click the Verify icon ☑ on the toolbar to verify the device application has no errors.
2. Click the code generation icon 📇 on the toolbar.
3. In the Generate Device Application wizard, in Device Platform, select **BlackBerry**, and, optionally, select:

| Option | Description |
|---|---|
| Server domain | Select the domain to use for the connection profile. The profiles used in the design appear in the Profile column. The initial value of "default" appears under the domain. Select the Domain column to choose a different domain. |
|  | You can enter any value you want or select one that is available. The list of available domains are returned from the profile. If you have previously connected, it caches the list of last known domains. If you have never connected, "default" is returned. |
| Locale | Expand this section to see a list of available locales from which you can select. |

| Option | Description |
|---|---|
| Advanced | Expand this section for advanced options:<br>• Check mobile business object on Sybase Unwired Platform Server – select to verify that the mobile business objects that are used in the device application exist on the corresponding Unwired Server.<br>• Mobile Business Object Group – the mobile business object group that contains the mobile business objects you want to verify. Click **Generate Code** to launch the Generate Code wizard. |

**4.** Click **Next**.

**5.** In the Generate Device Application wizard, enter the information, then click **Finish**:

| Option | Description |
|---|---|
| Favorite Configurations | (Optional) Select a saved configuration from the drop-down list.<br><br>**Note:** The **Remove the custom folder** option state is not saved in a favorite configuration. You must explicitly choose that option when you want to remove the custom folder that contains any custom coding you have added. |
| Locations | • Generate code only – if this option is selected, the code is not compiled.<br>• Deploy the BlackBerry application – select to deploy the BlackBerry application to the specified location. Selecting this option activates the Deploy Configurations section, where you can set the locations for the BlackBerry rapc compiler, simulator location, and so on. |

| Option | Description |
|---|---|
| Deploy Configurations | • BlackBerry rapc compiler – the BlackBerry JDE rapc compiler location. This determines the operating system version of the generated BlackBerry application.<br>• Copy to Simulator location – the filepath to the location of the compatible simulator you want to use for development and testing.<br>   • Start the BlackBerry Simulator after copying – select to start the simulator after you generate the device client application.<br>   • Start MDS automatically – select if a connection to the mobile business object is required. This is a one-time action. A check is performed to see whether the MDS is started. If it is, it will not be started twice. Once the MDS is running, it does not have to be restarted.<br>   • Disable signing – disables the signing process for testing only. This option is available only when you are copying the generated COD files to a simulator location for testing.<br>• Copy to desktop location – the location where you want to copy the generated .cod files.<br>• Copy to Workspace location – the location for the workspace to which you want to copy the device client application. |
| Debug the BlackBerry application | Debug configurations:<br>• Client project<br>• Options project<br>• Launch the BlackBerry project<br>   • Launch configuration<br><br>**Note:** This option is enabled only if you have installed the BlackBerry Java Plug-in for Eclipse. |

| Option | Description |
|---|---|
| Advanced | • Open the generated folder under Windows Explorer – select this option to open the generated folder under Windows Explorer when code generation is complete. This is useful to locate the generated source so you can move it to a JDE project or for custom coding. |
| | • Generated artifacts location – this option indicates the location of the folder where all the code is generated. The default is <workspace>\<dad_project_name>\DAD Generated Code\<dag_file_name>\Blackberry. Click **Browse** to change the location. Click **Restore** to restore to the default location. |
| | • Remove the Custom folder – select this option to remove any previously generated custom code placed in the Custom folder. |
| | • Use JDK path – use this JDK path for the **rapc.exe** compiler from RIM. This must be in the following format: `c:\program files\java\jdk1.6.0_16\bin`. |
| | • Configure package for generated code – select this option to specify one or more of the following package names. |
| |    • Package for the client code - The package for the code in `XX.cod`. |
| |    • Package for the option code - The package for the UI code in `XXOptions.cod`. |
| |    • Package for the object API code - The package for the object API. |

**See also**
• *API Documentation* on page 333

### *Generating Code For Testing on a BlackBerry Simulator*

Use the Generate Device Application wizard to generate a device application for development and testing on a simulator.

**Prerequisites**

You must have an existing mobile application project and the BlackBerry JDE or the BlackBerry Java plug-in installed.

**Task**

1. Right-click on any page (except the Introduction page) of the Device Application Designer and select Generate Device Application. Or, click on the Generate Device Application icon ![icon] in the toolbar.
2. In the Generate Device Application Designer wizard, under Device Platform, select BlackBerry and click **Next**.
3. In the next page, select from these options: **Copy to Simulator location** and accept the default, or click **Browse** to select the location to which to copy the generated code.

| Option | Description |
|---|---|
| Deploy the BlackBerry application | Select this option to deploy the device application to the BlackBerry Simulator for testing. |
| BlackBerry rapc compiler | The BlackBerry JDE rapc compiler location. This determines the operating system version of the generated BlackBerry application. |
| Copy to Simulator location | The Simulator location is the filepath to the location of the compatible simulator you want to use for development and testing. <br>• Start the BlackBerry Simulator after copying – select to start the BlackBerry Simulator after the files are copied to the specified location. <br>• Start MDS automatically – select to connect to the mobile business object for testing. If the MDS is already running, you do not have to restart it. <br>• Disable signing – disables signing for testing purposes. |

4. Click **Finish**.

The code is generated and copied to the specified simulator location.

### BlackBerry Java Plug-in for Eclipse Integration

The Device Application Designer Code Generation wizard is integrated with the BlackBerry Java Plug-in for Eclipse.

You can launch a BlackBerry project directly from Eclipse after code generation. This allows you to debug generated device application code that contains the user interface framework within a BlackBerry project.

**Note:** To use this feature, you must first install the BlackBerry Java Plug-in for Eclipse.

See the documentation on the BlackBerry developer Web site *http://docs.blackberry.com/en/ developers/?userType=21* for more information about the BlackBerry Java Plug-in for Eclipse.

### Generating BlackBerry Code For Debugging

Generate BlackBerry code and launch a BlackBerry project, where you can debug the generated code.

**Prerequisites**

Install the BlackBerry Java Plug-in for Eclipse v1.1.

**Task**

1. Click the Verify icon on the toolbar to verify the device application has no errors.
2. Click the code generation icon on the toolbar.
3. In the Generate Device Application wizard, in Device Platform, select **BlackBerry**, and, optionally, select:

| Option | Description |
| --- | --- |
| Server domain | The domain to use for the connection profile. The profiles used in the design appear in the Profile column. The initial value of "default" appears under the domain. Select the Domain column to choose a different domain. |
| | You can enter any value you want or select one that is available. The list of available domains are returned from the profile. If you have previously connected to any domains, it caches the list of last known domains. If you have never connected to a domain, "default" is returned. |
| Locale | Expand this section to see a list of available locales from which you can select. |

| Option | Description |
|--------|-------------|
| Advanced | Expand this section to see advanced options: <br> • Check Mobile Business Object on Sybase Unwired Platform Server – verify that the mobile business objects that are used in the device application exist on the corresponding Unwired Server. <br> • Mobile Business Object Group – the mobile business object group that contains the mobile business objects you want to verify. Click **Generate Code** to launch the Generate Code wizard. |

4. Click **Next**.
5. (Optional) In the Generate Device Application wizard, in Favorite Configurations, select a saved configuration from the drop-down list.
6. Select **Debug the BlackBerry Application with BlackBerry Java Plug-in** and click **Browse** to specify the client and options project to use to debug the device application. Click **New** to create a new BlackBerry project.

   Two projects are created for the generated code.
   • Client project – contains the application with an icon that you can view from the home screen or from the download folder. This application includes the user interface that you see in the Device Application Designer in the Flow Design and Screen Design pages.

   > **Note:** If more than one application will reside on the same device, they must each have their own unique client package name unless the object API code are exactly the same for those applications.

   • Options project – contains settings screens and the object API code and can be accessed from BlackBerry options.
7. (Optional) Select **Launch the BlackBerry Project** to launch the BlackBerry project with a launch configuration after the code generation wizard finishes, then from Launch Configuration, select the launch configuration you want to use.
8. (Optional) To create a new launch configuration, or edit an existing launch configuration, click **Configure**.
9. (Optional) In the Advanced section, select **Remove the Custom Folder** to remove any previously generated custom code placed in the Custom folder.
10. Click **Finish**.

### Creating a BlackBerry Project

Create new client and library options projects in which to debug the generated device application code.

1. In the Device Application Designer Code Generation wizard, next to either Options Project or Client Project, click **New**.
2. In the New BlackBerry Project wizard, enter a name for the project.
3. In Contents, select either:
   - Create new project in workspace – to create a new project with the specified name in the workspace.
   - Create project from existing source – to specify the location from which the New Java Project wizard retrieves an existing Java project.
4. Select **Use a project specific JRE** to specify a specific JRE, or select **Use default JRE** to specify the default JRE in the workspace.
5. (Optional) Click **Configure JREs** to add, delete, or edit JRE definitions.
6. (Optional) Select **Add project to working sets**, then click **Select** to add the BlackBerry project to an existing Eclipse working set, or to create a new working set.
7. (Optional) Click **Next** to define the Java build settings.

   See the *Java Development User Guide > Reference > Wizards and Dialogs > Create New Elements > New Java Project Wizard* for information about defining Java build settings.
8. Click **Finish**.

### Creating a Launch Configuration

Create and configure a new launch configuration for the BlackBerry project.

1. In the Configuration Settings section of the code generation wizard, select **Launch the BlackBerry Project**.
2. Click **Configure**.
3. Right-click **BlackBerry Simulator** and select **New**.
4. On the **Projects** tab, select the project for which to create a configuration.
5. Click the **JRE** tab, and select the JRE to use.
6. Click the **Simulator** tab, and configure the settings for the BlackBerry Simulator.
7. Click **Apply**.
8. Click **Close**.

### Cleaning the BlackBerry Simulator

After running the `clean.bat` file in the BlackBerry Java Plug-in component pack simulator directory to clean the BlackBerry simulator, or if you manually delete the required Device Application Designer framework `cod/alx/debug` files, you must copy some files before

you can run or debug the Sybase Unwired Platform BlackBerry project without using the Device Application Designer code generation wizard again.

Go to the BlackBerry RIM plugin installation location, and copy these files into the corresponding BlackBerry Java Plug-in component pack simulator path:

| Files | File location |
|-------|---------------|
| • `UltraLiteJ.alx`<br>• `UltraLiteJ.cod` | `%Sybase%\Unwired_WorkSpace`<br>`\Eclipse\sybase_workspace\mo-`<br>`bile\eclipse\plugins\com.syb-`<br>`ase.uep.bob.rim\generate`<br>`\blackberry` |
| • `BOBFramework.alx`<br>• `BOBFramework.cod`<br>• All `BOBFramework-x.debug` files<br>• `SUPPushListener.cod`<br>• `SUPPushListener.alx` | `%Sybase%\Unwired_WorkSpace`<br>`\Eclipse\sybase_workspace\mo-`<br>`bile\eclipse\plugins\com.syb-`<br>`ase.uep.bob.rim\generate`<br>`\blackberry\build-`$x.x.x$, where<br>$x.x.x$ is the version of the JDE you are using. |

### *Generating a Device Application to Push to a BlackBerry Device*

Use the Generate Device Application wizard to generate a device application to push to a device.

**Prerequisites**

You must have Research in Motion (RIM) server and client software installed and synchronized with a BlackBerry mobile device.

**Task**

1. Right-click on any page (except the Introduction page) of the Device Application Designer and select **Generate Device Application**. Or, click on the Generate Device Application icon 🖥️ on the toolbar.
2. In the Generate Device Application wizard, select **Copy to desktop location** and click **Browse** to find the location to which you want to copy the generated COD files.
3. Click **Finish**.
4. Connect your device to the machine where you copied the generated COD files, and sync them to your device.

### *Generating Custom Code Subclasses*

Generating custom subclasses allows you to modify the generated code so you can further customize screens and controls.

**1.** In the Screen Design page, click the **Platform-specific** tab.

**2.** In the **Generate a custom coding options subclass option**, click **false** to change it to **true**.
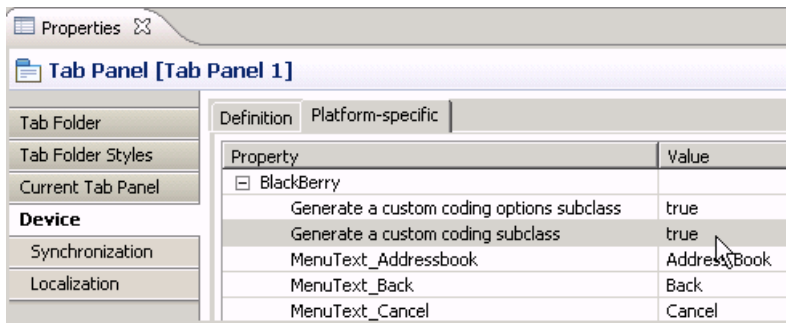


You can set the custom coding options on:

- Device – subclasses for the BOBCUIDefinition and BOBCOptionsDefinition will be generated if set to true.
- Display – a subclass for the screen is generated if set to true.
- Tab panel – a subclass for the tab panel is generated if set to true.
- Tab folder – a subclass for the tab folder is generated if set to true.

**3.** (Optional) To change all of the property values to true at once (Device, Display, Tab folder, and Tab panel):

a) Right-click in the Screen Design page and select **Preferences**.

b) In Preferences, go to **Sybase, Inc. > Mobile Development > Device Application Designer > Devices > BlackBerry**.

c) In Platform Property, select **Generate custom subclasses for all**.

d) Restart Eclipse.

**4.** Click the Verify icon  on the toolbar to verify the device application has no errors.

**5.** Click the code generation icon  on the toolbar and proceed through the Generate Device application wizard to generate the custom code.

### Generating Code For a Windows Mobile Device Application

Use the Generate Device Application wizard to generate code for a Windows Mobile device application.

**1.** Click the Verify icon  on the toolbar to verify the device application has no errors.

**2.** Click the code generation icon  on the toolbar.

**3.** In the Generate Device Application wizard, select **Windows Mobile** and, optionally, select:

| Option | Description |
|--------|-------------|
| Locale | Expand this section to see a list of available locales from which you can select. |
| Advanced | Expand this section for advanced options: <br>• Check mobile business object on Sybase Unwired Platform Server – select to verify that the mobile business objects that are used in the device application exist on the corresponding Unwired Server. <br>• Mobile Business Object Group – the mobile business object group that contains the mobile business objects you want to verify. <br>To generate code for the Mobile Application Project, click **Generate Code**. Use the wizard to generate the metadata classes for the selected mobile business objects. |

4. Click **Next**.
5. Enter the information for the device application code generation options:

| Option | Description |
|--------|-------------|
| Favorite Configurations | Select a configuration. |
| Device | • Target device – select the device. |
| Code Generation | • Visual Studio solutions folder – accept the default or click **Browse** to enter the location for the Visual Studio Solutions folder. <br><br>**Note:** A .cab file is generated and placed in the Visual Studio solutions folder. <br><br>• Solution name – enter the name of the Visual Studio solution. <br>• Delete solution folder prior to generation – remove existing source folders before regenerating the Visual Studio solution. <br>• Generate replication based application – select if the application has replication based synchronization. <br>• Generate messaging based application – select if the application has message based synchronization. |

| Option | Description |
|---|---|
| Advanced | When the device application code is generated, two projects are created—the client project, which contains the user interface screens, and the mobile application project, which contains the mobile business objects that are used to access and update the data.<br>• Client project name – enter the name of the project that contains the user interface.<br>• Client project namespace – enter the namespace to use for the generated UI classes.<br>• Client project assembly name – the name of the generated `.exe` file for the project. This is the name that appears on the mobile device.<br>• Mobile application project name – the name of the Mobile Application Project that contains the mobile business objects used in the device application.<br>• Mobile application project assembly name – accept the default or enter the name for the .dll of the mobile application project.<br>• Client project icon – click **Browse** to select an icon with which to associate the generated `.exe` file. This is the icon that appears on the mobile device.<br>• Generate source codes for Sybase Windows frameworks – select to generate sources for Sybase Unwired Platform Windows frameworks, including stock screens and actions.<br>• Deploy to an ActiveSync connected device or emulator – select this option to deploy the generated code to a Windows Mobile device or emulator. ActiveSync enables the transferring and installation of the application on the mobile device.<br><br>**Note:** ActiveSync is for Windows XP. Windows Vista, Windows 7, and Windows 2008 uses Windows Mobile Device Center.<br><br>• Create a shortcut in Programs – select to create a shortcut in the Programs folder on the device.<br>• Shortcut name – enter a name for the shortcut. |

| Option | Description |
|--------|-------------|
| | • Deployment timeout (minutes) – the maximum time to wait for deployment to the device.<br>• Perform silent install – use this option only when you are deploying to the emulator (not the device itself). This enables deployment to proceed with no user input. |

6. (Optional) Click **Start Device Emulator Manager** if the Device Emulator Manager is not already running and you want to deploy the application to an emulator.

7. Click **Finish**.

*Deploying the Device Application to the Windows Mobile Emulator*
Deploy generated code to a connected Windows Mobile device or emulator.

1. Click the Verify icon 🖳 on the toolbar to verify the device application has no errors.

2. Click the code generation icon 🖳 on the toolbar.

3. In the Generate Device Application wizard, select **Windows Mobile** and, optionally, select:

| Option | Description |
|--------|-------------|
| Locale | Expand this section to see a list of available locales from which you can select. |
| Advanced | Expand this section for advanced options:<br>• Check Mobile Business Object on Sybase Unwired Platform Server – verify that the mobile business objects that are used in the device application exist on the corresponding Unwired Server.<br>• Mobile Business Object Group – the mobile business object group that contains the mobile business objects you want to verify.<br><br>To generate code for the mobile application project, click **Generate Code**.Use the wizard to generate the metadata classes for the selected mobile business objects. |

4. Click **Next**.

5. Enter the information for the device application code generation options:

| Option | Description |
|--------|-------------|
| Favorite Configurations | Select a configuration. |

| Option | Description |
|---|---|
| Device | • Target device – select the device.<br>• Library version – choose the Microsoft .NET version of the library used to compile the generated code.<br><br>**Note:** Microsoft .NET Compact Framework 2.0 requires SP 2.0. |
| Code Generation | • Visual Studio solutions folder – accept the default or click **Browse** to enter the location for the Visual Studio Solutions folder.<br><br>**Note:** A .cab file is generated and placed in the Visual Studio solutions folder.<br>• Solution name – enter the name of the Visual Studio solution.<br>• Delete solution folder prior to generation – remove existing source folders before re-generating the Visual Studio solution.<br>• Generate replication-based application – select if the application has replication based synchronization.<br>• Generate messaging-based application – select if the application has message based synchronization. |

| Option | Description |
|--------|-------------|
| Advanced | When the device application code is generated, two projects are created—the client project, which contains the user interface screens, and the mobile application project, which contains the mobile business objects that access and update the data.<br>• Client project name – enter the name of the project that contains the user interface.<br>• Client project namespace – enter the namespace to use for the generated UI classes.<br>• Client project assembly name – the name of the generated `.exe` file for the project. This is the name that appears on the mobile device.<br>• Mobile application project name – the name of the mobile application project that contains the mobile business objects used in the device application.<br>• Mobile application project assembly name – accept the default or enter the name for the .dll of the mobile application project.<br>• Client project icon – click **Browse** to select an icon with which to associate the generated `.exe` file. This is the icon that appears on the mobile device.<br>• Generate source codes for Sybase Windows frameworks – generate sources for Sybase Unwired Platform Windows frameworks, including stock screens and actions.<br>• Deploy to an ActiveSync connected device or emulator – deploy the generated code to a Windows Mobile device or emulator. ActiveSync enables the transferring and installation of the application on the mobile device.<br><br>**Note:** ActiveSync is for Windows XP. Windows Vista, Windows 7, and Windows 2008 uses Windows Mobile Device Center.<br><br>• Create a shortcut in Programs – create a shortcut in the Programs folder on the device.<br>• Shortcut name – enter a name for the shortcut.<br>• Deployment timeout (minutes) – the maximum time to wait for deployment to the device. |

| Option | Description |
|---|---|
|  | • Perform silent install – use this option only when you are deploying to the emulator (not the device itself). This enables deployment to proceed with no user input.<br>• Generate metadata access classes – generate additional metadata classes that describe the attributes and parameters of the generated MBO classes. |

6. Click **Start Device Emulator Manager** if the Device Emulator Manager is not already running.

7. In the Device Emulator Manager, right-click the emulator to which to deploy the application, and select **Connect**.

8. When the selected emulator is connected, right-click the same emulator, and select **Cradle**.
   The Microsoft ActiveSync Synchronization Setup wizard launches.

9. In the Synchronization Setup wizard, click **Cancel**.

   ActiveSync connects to the device emulator.

10. In the device application code generation wizard, click **Finish**.
   The code generation wizard verifies the code, checks for a connected device, and compiles the code in the specified location.

### Rebuilding the Generated Solution in Visual Studio
After generating code for a Mobile Windows device, you can modify the code in Visual Studio.

### Prerequisites
Visual Studio must be installed.

### Task

When you generate the code for a Windows Mobile device using the Device Application Designer, the Visual Studio solution is saved to the folder you designated in the Code Generation section of the Generate Mobile Windows Application wizard.

1. In Visual Studio, select **File > Open > Project/Solution**.

2. Browse to the solution file (.sln) you want to open and double-click the file.

3. In Solution Explorer, right-click the solution and select **Rebuild Solution**.

4. Select **File > Save**.
   You can now open the form for which you want to modify the code.

# Develop a Mobile Workflow Package

Sybase Mobile Workflow Forms Editor lets you create or modify mobile business objects on different device platforms without custom device development.

Mobile Workflows support the occasionally connected user and addresses the replication and synchronization issues those users present for the back-end system. The user can update, delete, or create data even when not connected to the network. When the user connects, data is automatically sent and received.

A Mobile Workflow application requires an integration module on the server side, which is implemented by a static set of logic that processes Mobile Workflow-specific metadata to map keys to and from mobile business object attributes, personalization keys, and parameters. This integration module processes the notifications identified by matching rules configured for the server-initiated starting point and also processes the responses sent to the server from the device.

The Mobile Workflow Forms Editor allows you to create the forms that can be assigned to the devices that have the Sybase Mobile Workflow application installed and that are registered in Sybase Control Center.

When you use the Mobile Workflow Forms Editor, the data source for the application is supplied by mobile business objects that are designated as message-based synchronization mobile business objects when deployed to the server. The Mobile Workflow Forms Editor allows you to develop mobile workflow screens that can call on the create, update, and delete operations, as well as object queries of a mobile business object.

You can generate a Mobile Workflow form that works on these platforms:
* Apple iOS
* Windows Mobile

See the *Sybase Unwired Platform Installation Guide* for supported version levels.

When designing the user interface for Mobile Workflow forms for mobile devices, keep it simple. The Mobile Workflow form should deliver clear, concise content with minimal navigation menus. The Mobile Workflow should be compatible with a wide variety of devices, and support both horizontal and vertical screen layouts. Consider the limitations of mobile devices when designing a Mobile Workflow form.

## The Mobile Workflow Forms Editor

Use the Mobile Workflow Forms Editor to develop mobile workflow forms.

The Mobile Workflow Forms Editor contains these pages:

**Introduction**

Create a new mobile workflow or double-click an existing one, to open the Introduction page of the Mobile Workflow Forms Editor. This page is the starting point for designing your mobile workflow form. You can access help from this page.

**Flow Design**

The Flow Design page displays an empty canvas if you did not choose any of starting points in the New Mobile Workflow Forms Editor. If you selected a starting point when creating the new mobile workflow form, the selected starting point appears.

Use this page to specify how starting points and screens link together. The Palette view on the right shows available starting points, screens, and connections. You can drag and drop items from the Palette to the canvas. Screens and starting points that you drag to the Flow Design canvas appear as icons. Double-click a screen to open the Screen Design page.

Right-click anywhere in the Flow Design page to open a context menu.

**Screen Design**

Use this page to design the screen of your mobile workflow form. The Palette view shows the menu items and controls you can use to design a new screen. Drag controls to design the user interface.

**See also**

## Creating a New Mobile Workflow Form

Access the Sybase Mobile Workflow Forms Editor.

**Prerequisites**

You must have an existing project in which to create the mobile workflow form.

**Task**

1. In the Mobile Development perspective, select **File > New > Mobile Workflow Forms Editor**.
2. Follow the instructions in the New Mobile Workflow Forms Editor wizard:

| Field | Description |
|---|---|
| Enter or select the parent folder | Select the mobile application project in which to create the mobile workflow form. |
| File name | Enter a name for the mobile workflow form. The extension for mobile workflow forms is .xbw. |
| Advanced | Link the mobile workflow form to an existing file in the file system. |
| Link to file in the file system | Click **Browse** to locate the file to which to link the mobile workflow form. Linked resources are files or folders that are stored in the file system outside of the project's location. If you link a resource to an editor, when you select the editor, the resource is selected in the Work-Space Navigator. Conversely, when you select the resource in the WorkSpace Navigator, the editor is selected. |

3. In the Starting Points page, select the ways the mobile workflow will start:
   - **Can be started on demand from the client** – allows the user to start the mobile workflow on the device and submit data to the server.
   - **Responds to server-driven e-mail notifications** – when a notification that matches the defined matching rules arrives in the Notification Mailbox, it is then sent as a notification to the devices that match those matching rules. When you select this option, you will set up the server-initiated starting point in the next screens of the wizard. Follow the procedure for *Creating a Server-initiated Starting Point*.
   - **Has setup options that need to be set when the client application is first activated** – the first time the workflow is launched, the screen pointed to by the Activation screen is shown.
   - **Credentials (authentication) may be requested dynamically from the client application** – select to create a credential request starting point with pre-defined username and password keys.
4. (Optional) If you selected the **Responds to server-driven e-mail notifications** option in the previous step, follow the procedure in *Creating a Server-initiated Starting Point*.
5. Click **Finish**.

6. In the Mobile Workflow Forms Editor, click **Flow Design** to start designing the mobile workflow form.

**See also**
- *Flow Design Page Properties* on page 256
- *Flow Design Palette Options* on page 259
- *Screen Design Page Properties* on page 261
- *Screen Design Palette Options* on page 262
- *Mobile Workflow Forms Editor Context Menus* on page 265
- *Sending a Test Notification* on page 266

## Flow Design Page Properties

These are the Flow Design page properties shown in the Properties view when no objects in the Flow Design page are selected.

*General*

| Property | Description |
|---|---|
| Module name | The name of the module. |
| Module version | The version of the module. |
| Module description | Enter a description for the module. |
| Module display name | Enter the name that appears on the workflow application when it is viewed on the client. |
| Delete message after processing | If this option is selected, server-initiated workflow messages are removed from the device Inbox after they have been processed. A message is considered to have been processed if the corresponding workflow application is closed as a result of an asynchronous submit menu item being activated. |
| Mark message as processed after processing | If this option is selected, server-initiated workflow messages are marked as processed in the device Inbox (indicated by a green check mark over the original icon) after they have been processed. A message is considered to have been processed if the corresponding workflow application is closed as a result of an asynchronous submit menu item being activated. |
| Client icon | Select the icon that appears on the client. |

| Property | Description |
|---|---|
| Credentials cache key | Specify a key to use for the credential starting point. If you do not enter anything, each workflow will have a different credentials key. If there is more than one workflow with the same credentials key, then all of those workflows can share the same credentials so that authentication for each workflow does not have to be separately defined. |
| Requires activation key | Specify a key to use for activation. If you do not enter anything, each workflow will have a different activation key, which means that each workflow will have to be activated separately. If you specify an activation key, then when you activate one workflow, all the other workflows sharing the specified activation key are automatically activated. |

*Rulers and grid*

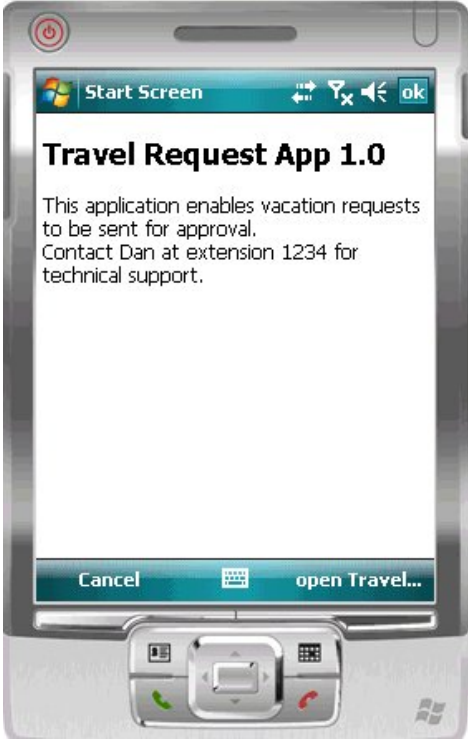| Property | Description |
|---|---|
| Display | • Show ruler<br>• Show grid<br>• Grid in front |
| Measurement | • Ruler units<br>• Grid spacing |
| Grid line | • Color<br>• Style |
| Snap to grid | Automatically align objects to the grid lines on the canvas. |
| Snap to shapes | Position objects to grid lines that go through the vertical and horizontal edges of other objects. |

*Localization*

| Property | Description |
|----------|-------------|
| New | Launch the New Locale wizard and create a new locale file for the mobile workflow application. |
| Update | Update the currently loaded locale file. |
| Load | Load a locale properties file to the Mobile Workflow Designer. |
| Validate | Select a locale properties file and click to validate. |
| Edit | Select the locale properties file, then click to edit the locale properties file. |
| Remove | Select the locale properties file to remove, then click to remove. |

*Matching rules*
Any existing matching rules appear in the matching rules Properties view. Matching rules determine how e-mail messages are redirected at runtime.

| Property | Description |
|----------|-------------|
| New | Create a new matching rule. |
| Delete | Select a matching rule and click to delete. |
| Edit | Select a matching rule and click to edit. |

*Authentication*

| Property | Description |
|----------|-------------|
| Use static credentials | Select when the authentication field is the same for all users accessing the resource. |
| Use SUP server connection profile authentication | Use properties from the connection profile to supply credential data for authentication. This is similar to hard-coded credentials but the credentials are taken from the specified server connection profile instead of being specified in the Mobile Workflow Forms Editor. |
| Use hard-coded credentials | Specify a user name/password combination for authentication. |

| Property | Description |
|---|---|
| User name | This field is editable only if you selected the hard-coded credentials option. Enter the user name to use for authentication. |
| Password | This field is editable only if you selected the hard-coded credentials option. Enter the password to use for authentication. |
| Unwired Server domain | Optionally specify the Sybase Unwired Platform domain to use. If specified, a context variable is created in the application for which the initial value is set to this domain. At runtime, when determining authentication, $@<domain>$ is added to the user name, thus allowing the user access to that domain. |

**See also**

## Flow Design Palette Options

Use the items in the Mobile Workflow Forms Editor Flow Design palette to configure the screens and connections for the mobile workflow application.

*Connections*

| Connection | Description |
|---|---|
| Go to | Specifies which screen the user goes to upon completing an action on the current screen, such as clicking OK or Next. |
| Operation success | Specifies which screen the user automatically goes to upon successful completion of an operation. |

| Connection | Description |
|---|---|
| Operation error | Specifies the screen that the user goes to automatically when an error occurs while attempting to complete an operation. |
| Listview details | Specifies which screen to open when a row in a listview is clicked. Once on the details screen, you can add, update, or delete rows from the listview. |
| Listview add | Specifies a screen that can be directly navigated to in order to add a new row to the listview. |

*Screen*

Select the screen ▣ and click the Flow Design page to create a new empty screen.

*Starting points*

| Starting point | Description |
|---|---|
| Activate | The Activate starting point indicates a screen that is displayed to the user the only the first time they open the application. It is not displayed the second and subsequent times the application is opened. For example, you might want to use an Activate starting pint to show usage instructions for the application to the user the first time they open the application. |
| Credential request | Provide the credentials for authentication. During execution, authentication credentials are evaluated in this order:<br><br>**1.** User-defined user name and password keys<br>**2.** Credential cache<br>**3.** Specified generic credential pair |
| Server-initiated | Use this starting point to define the matching rules for the workflow form and to specify how to extract values from notifications and map them to mobile business object read and search parameters. |
| Client-initiated | The mobile business object referencing starting point. |

**See also**
- *Creating a New Mobile Workflow Form* on page 254
- *Flow Design Page Properties* on page 256
- *Screen Design Page Properties* on page 261
- *Screen Design Palette Options* on page 262
- *Mobile Workflow Forms Editor Context Menus* on page 265
- *Sending a Test Notification* on page 266

## Screen Design Page Properties

These are the Screen Design page properties shown in the Properties view when no objects in the Screen Design page are selected.

*General*
These are the general properties for the screen.

| Property | Description |
|---|---|
| Name | The name of the screen. |
| Key | The key associated with the screen. |
| Background color | Click **Color picker** to choose a background color for the screen. |
| | You can also enter the value manually. The possible values are black, blue, brown, gray, green, orange, pink, purple, red, white, yellow, any string with the format #rrggbb, where rr, gg and bb are the hexadecimal values, in the range 0 to ff(255), for the red, green and blue. |
| | **Note:** The background color for screens can be set only for Windows Mobile devices. It is not supported for iOS devices. |
| OK action | Select the action that will be performed when the user clicks OK on the screen. |

*Rulers and grid*
The ruler and grid properties are applied to the Screen Design page, or canvas.

| Property | Description |
|---|---|
| Display | <ul><li>Show ruler</li><li>Show grid</li><li>Grid in front</li></ul> |
| Measurement | <ul><li>Ruler units</li><li>Grid spacing</li></ul> |
| Grid line | <ul><li>Color</li><li>Style</li></ul> |
| Snap to grid | Automatically align objects to the grid lines on the canvas. |
| Snap to shapes | Position objects to grid lines that go through the vertical and horizontal edges of other objects. |

**See also**
- *Creating a New Mobile Workflow Form* on page 254
- *Flow Design Page Properties* on page 256
- *Flow Design Palette Options* on page 259
- *Screen Design Palette Options* on page 262
- *Mobile Workflow Forms Editor Context Menus* on page 265
- *Sending a Test Notification* on page 266

## Screen Design Palette Options

Use the items on the palette to design the screen for the mobile workflow application.

| Option | Description |
|---|---|
| Menu item | Use to provide additional options for the top-level menu on the screen. |
| Checkbox | Displays a true/false (or yes/no) option on the screen. |
| Editbox | Select the logical type for the edit box, then enter text and assign values to strings, dates, decimals, integers, and so on. |
| Choice | Display a collection of items on the screen as a drop-down list from which the user can choose. |

| Option | Description |
|---|---|
| Slider | Represent numeric values within a range. |
| Signature | Add a signature box to the screen. The value is a list of points that, combined, make up the signature that gets or is drawn to the control, for example, -1x-1,5x7,5x8,-1x-1, where -1x-1 is interpreted as an interruption of a line drawn to the screen. |

| Option | Description |
|---|---|
| HtmlView | Allows text with HTML markup to display on the screen. For example, this code: |
|  | <pre>&lt;H2&gt;Travel Request App 1.0&lt;/H2&gt;<br>This application enables vaca-<br>tion requests to be<br>sent for approval.&lt;br&gt;Contact<br>Dan at extension 1234<br>for technical support.</pre> |
|  | displays like this on the device: |
|  |  |
|  | **Note:** The HtmlView control fills the entire screen of a workflow form, so no other controls can be placed on a screen that hosts an HtmlView control. |

| Option | Description |
|---|---|
| ListView | Displays lines that can be clicked on to show further details of the selected line. This facilitates the navigation for a one-to-many mobile business object relationship. For example, the listview could contain the items of a sales order. When a line in the sales order is clicked on, a details screen for that sales order appears. |
| | **Note:** The ListView control fills the entire screen of a workflow form, so no other controls can be placed on a screen that hosts a ListView control. |

**See also**
- *Creating a New Mobile Workflow Form* on page 254
- *Flow Design Page Properties* on page 256
- *Flow Design Palette Options* on page 259
- *Screen Design Page Properties* on page 261
- *Mobile Workflow Forms Editor Context Menus* on page 265
- *Sending a Test Notification* on page 266

## Mobile Workflow Forms Editor Context Menus

Right-click in any of the pages (except the Introduction page) of the Mobile Workflow Forms Editor to display a context menu and perform these actions.

| Option | Description |
|---|---|
| Edit | Edit the selected object. |
| Delete | Delete the selected object. |
| Mobile Business Object References | Shows current mobile business object references and allows you to create new mobile business object references and edit or delete existing mobile business object references. |
| | **Note:** This menu item appears only on the Flow Design page. |
| Generate Mobile Workflow Package | Generate the mobile workflow form code. |

| Option | Description |
|---|---|
| Send a notification | Send an e-mail from the Mobile Workflow Forms Editor to a device user. This can be used to quickly test e-mail triggered applications. The test e-mail bypasses the Notification Mailbox. |
| | **Note:** The Cc and Bcc fields are used only for matching rules. An e-mail is not sent to the users entered in these fields. |
| Select | Select: <br><br>• All – selects all objects on the page. <br>• All shapes – selects all shapes on the page. <br>• All connections – selects all connections on the page. |
| Arrange all | Arranges the shapes on the page. |
| View | View: <br><br>• Grid <br>• Rulers <br>• Page breaks <br>• Recalculate page breaks <br>• Snap to grid |
| Zoom | Select from the zoom options. |
| Show properties view | Shows the Properties view for the selected object. |

**See also**

## Sending a Test Notification

Send a test notification from the Mobile Workflow Forms Editor to a device user to test server-initiated Mobile Workflow forms.

1. Right-click in the Flow Design or Screen Design page of the Mobile Workflow Forms
   Editor and select **Send a notification**.
2. Fill out the notification form and click **Send**.

   **Note:** The Cc and Bcc fields are used only for matching rules. A notification is not sent to
   the users entered in these fields.

   The notification bypasses the Notification Mailbox.
3. (Optional) Click **Get Device Users** to bring up a list of registered device users to whom
   you can send the notification.

**See also**

# Starting Points

Starting points specify which screen on the Mobile Workflow a user sees when they perform a
certain action.

Starting points include:

- Client-initiated – this starting point is activated when the user opens the Mobile Workflow
  on the device.
- Activate – this is the starting point that is activated when the user first opens the workflow
  application. It does not activate on the second and subsequent times the workflow
  application is started. Multiple workflow applications can share the same Activate starting
  point. Different workflows can specify an activation key, and workflows with the same
  activation key share their activation status. For example, if Workflow A and Workflow B
  both specify an activation key of AB (via the key attribute on the RequiresActivation tag),
  when Workflow A gets activated, it also activates Workflow B such that when Workflow B
  is invoked for the very first time, its activation screen will not be displayed.
- Credential request – this starting point is activated when the workflow application requires
  a user name and password and the program does not have one in its cache, or the values in
  the cache are no longer valid.
- Server-initiated – this starting point is activated when a notification message is sent to the
  device that matches the matching rules specified in the Mobile Workflow application.

Each starting point has a list of keys with which it is associated. View the keys associated with
the starting point in the Properties view for the starting point.

**See also**

- *The Mobile Workflow Forms Editor* on page 253
- *Configuring a Mobile Workflow Package for Online Mobile Business Object Data Access* on page 278
- *Mobile Business Object Relationships* on page 282
- *Creating Screens Automatically by Dragging and Dropping a Data Source* on page 282
- *Adding Controls to the Screen* on page 283
- *Configuring Control Properties* on page 294
- *Adding Connections to Screens* on page 306
- *Activating the Workflow* on page 309
- *Localization* on page 309
- *Generating the Files For a Mobile Workflow Package* on page 315

## Adding a Starting Point Manually

Every screen must be part of a flow that connects to a starting point.

**Note:** If you manually create a server-initiated starting point that invokes an object query, the list key will not have any children. To create the children for the list key, you must use the New Mobile Workflow Forms creation wizard to create the server-initiated starting point, or you can drag and drop the mobile business object onto the Flow Design page. Either of these two methods will create the associated children key for a server-initiated starting point that invokes an object query. Otherwise, you will have to manually set up the list key's children.

1. From the palette on the Flow Design page, select a starting point, then click on the Flow Design canvas.
2. Add a screen to the starting point by either:
   - Dragging and dropping a data source, for example, a mobile business object or mobile business object operation, onto the Flow Design canvas, or
   - Selecting an empty screen from the Flow Design palette, then clicking on the Flow Design page.
3. Add a connection between the starting point and the screen.
4. Select **File > Save**.

   You can also automatically create starting points based on selections you make in the New Mobile Workflow Form wizard when you create a new Mobile Workflow form.

## Creating a Server-initiated Starting Point

Create a server-initiated starting point.

**Note:** Each Mobile Workflow Forms Editor can have only one server-initiated starting point.

1.  In the Mobile Development perspective, select **File > New > Mobile Workflow Forms Editor**.

2.  Follow the instructions in the New Mobile Workflow Forms Editor wizard:

| Field | Description |
|---|---|
| Enter or select the parent folder | Select the mobile application project in which to create the mobile workflow form. |
| File name | Enter a name for the mobile workflow form. The extension for mobile workflow forms is .xbw. |
| Advanced | Link the mobile workflow form to an existing file in the file system. |
| Link to file in the file system | Click **Browse** to locate the file to which to link the mobile workflow form. Linked resources are files or folders that are stored in the file system outside of the project's location. If you link a resource to an editor, when you select the editor, the resource is selected in the Work-Space Navigator. Conversely, when you select the resource in the WorkSpace Navigator, the editor is selected. |

3.  In the Starting Points page, select **Responds to server-driven e-mail notifications**.

4.  In the E-mail Processing wizard, click **Search** to locate the mobile business object with which to associate the server-initiated starting point.

5.  In the Search for Mobile Business Object wizard, click **Search**.

6.  Select the mobile business object from the list, and click **OK**.

7.  (Optional) Select an **Object query** and click **Next**.

    **Note:** Object queries that return result sets are not supported.

8.  Enter the contents for a sample notification message, and click **Next**.

9.  Select the text that will serve as the matching rule by right-clicking and selecting **Select as Matching Rule**; then click **Next**.

10. From **Parameters**, select each parameter for which there is a matching rule, and specify from where the parameter receives data in the Parameter Properties.

    **Note:** The supported syntax for the regular expression is that used by the C# Match() methods(s) of the class System.Text.RegularExpressions.Regex.

| Property | Description |
|---|---|
| Field | Select the field from the notification message from which the parameter receives data, for example, Subject. Selecting Subject indicates that the parameter receives data from the subject line of the notification message. |
| Start tag | Enter or select the text for the start tag regular expression, for example, `"Approval Request\("`. |
| End tag | Enter or select the text for the end tag regular expression, for example, `"\)was sent"`. |
| Format | Supports locale-specific parameter extraction. For example, you can indicate that a given date parameter will be in the form of YYYY-MM-DD in the e-mail, and another date parameter will be in the form of YYYY-DD-MM. This applies to the DateTime type. |
| Sample e-mail field contents | The value of the field previously chosen as the matching rule in the Specify Matching Rules page of the wizard. |
| Value extracted from sample e-mail field contents | The value the start tag and end tag matching rules would arrive at given the sample field value. |

**11.** Click **Finish**.

### Server-initiated Starting Point Properties

View and configure the server-initiated starting point properties.

Select the **Server-initiated** starting point in the Flow Design page to view and configure its properties.

#### Table 10. General properties for the server-initiated starting point

| Property | Description |
|---|---|
| Mobile business object | The mobile business object with which the server-initiated starting point is associated. |

| Property | Description |
|---|---|
| Object query | When you define an object query in the mobile business object associated with the Mobile Workflow form, the object query returns the specified instances of the mobile business object. For example, if a travel request to Hawaii is submitted, the object query returns an instance of that mobile business object so the approver can review the travel request details then modify the status and comment fields for that mobile business object. Object queries that return multiple instances are supported, as well as object queries that return only one instance. |
| Error screen | Select the screen to go to if the operation fails. |
| Jumpstart | Launch the Notification Processing wizard, where you can compose a sample notification message from which to specify parameter value extraction rules and matching rules. |
| Extraction rules | Shows the notification extraction rules for the server-initiated starting point. Select a rule and click **Edit** to change the transformation rule. |

Click **Keys** in the left pane of the Properties view to see the keys that are associated with the server-initiated starting point.

**Table 11. Keys**

| Property | Description |
|---|---|
| Key name | The name of the associated key. |
| Type | The type of the associated key, for example, string. |

| Property | Description |
|---|---|
| Data binding | Shows the data source to which the key is bound:<br><br>• Mobile business object attribute<br>• Mobile business object relationship<br>• MBO object query results<br>   If you select an object query that returns multiple MBO instances, you can bind the result set to a Listview control using a key that is of the list type.<br>• User-defined<br>• Extraction rule<br>   If you create a server-initiated starting point that invokes an object query with a parameter, this value is extracted from the email message automatically.<br><br>Click **New** to create a new key.<br><br>Select any key and click **Edit** to edit the information for an existing key.<br><br>Select a key and click **Remove** to remove an existing key. |

### Notification Extraction Rules

Notification extractions rules are used to extract parameter values to keys, so you can then map the parameters to objects, such as keys.

This means that you can extract values from the e-mail without needing to use them right away as parameter values. You can instead bind them to controls and display them to the user, or use them later on in the workflow to submit a create/read/update/delete (CRUD) operation.

Notification extraction rules include a regular expression that determines from where in the notification message the parameter value starts, and another regular expression that determines from where in the notification message the parameter value ends. Define the notification extraction rules when you create a server initiated starting point.

### Editing Notification Extraction Rules

Edit notification extraction rules in the server-initiated starting point General properties.

1. On the Flow Design page, select the Server-initiated starting point.
2. In the General Properties view, select the notification extraction rule to edit and click **Edit**.

**3.** In the Edit Notification Extraction Rule dialog:

| Property | Description |
|---|---|
| MBO search parameter | Enter the search parameters for the mobile business object (MBO). |
| Field type | Select the field in the notification from which the parameter value is extracted, for example, Subject, which indicates the parameter value is extracted from the subject line of the notification message. |
| Tag before parameter | Enter the tag to indicate what to extract from the beginning of the field's contents, for example, `"Approval Request\("`. |
| Tag after parameter | Enter the tag that indicates what to extract from the end of the field's contents, for example, `"\)was sent"`. |

**Note:** The supported syntax for the regular expression is that used by the C# Match() methods(s) of the class System.Text.RegularExpressions.Regex.

**4.** Click **OK**.

### Editing the Server-initiated Starting Point

Edit the Server-initiated starting point.

**1.** In the Flow Design page, right-click the **Server-initiated starting point** to edit and select **Edit Server-initiated starting point**.

**2.** Make your changes in the Notification Processing wizard and click **Finish**.

**3.** From the main menu, select **File > Save**.

### Adding Matching Rules

Matching rules determine how notification messages are redirected at runtime.

To see the Matching Rules section of the Properties view, verify there are no objects selected on the Flow Design page.

**1.** In the Properties view, click **Matching Rules**.

**2.** Click **Add**.

**3.** In the matching rules dialog, specify the type and regular expression for the matching rule, and click **OK**.

## Credentials

You can use either dynamic or static credentials in a mobile workflow form.

The user name and password values are required when the mobile workflow application invokes a mobile business object operation. These authentication values can be provided statically (at design time), or dynamically (by the user at runtime).

Dynamic credentials mean that the user sets the user name and password on a screen that is pointed to by the credential request starting point. The text fields must have the corresponding Credential Cache User Name and Password checkbox checked to indicate the value is to be used to provide the user name and password on the client. When the user logs in, the credentials are authenticated using the stored credentials.

**Note:** If an e-mail triggered workflow form has dynamic cached credentials, the cached credentials are not cached between invocations of the workflow form through an email trigger.

Static credentials mean that everyone who has access to the resource uses the same user name and password. By default, static credentials are used. The static credential user name and password for the mobile workflow application can be extracted from the selected Sybase Unwired Platform profile user name and password when the mobile workflow application is generated, or they can be hard-coded using the Properties view. After deployment, you can change static credentials in the Sybase Control Center.

The application can also have a credential screen (Credential Request) that appears if the mobile workflow application detects that the cached credentials are empty or incorrect.

### Setting Up Static Authentication

With static authentication, everyone who has access to the resource uses the same user name and password.

Set up static credentials in the Authentication section of the Properties tab. To see the Properties page, verify there are no objects selected on the Flow Design page.

1. In the Properties view, click **Authentication**.
2. Select **Use static credentials**.
3. Select from these options:
   - Use SUP Server connection profile authentication – selected by default. When the code is generated for the mobile workflow application, the user name and password associated with the SUP connection profile are used.
   - Use hard-coded credentials – sets the user name and password. When you select this option, the User name and Password fields are activated.
4. (Optional) If you selected the **Use hard-coded credentials** option in the previous step, enter the User name and Password that are to be used for authentication.
5. Select **File > Save**.

## Setting Up Dynamic Authentication

Use dynamic authentication when you want the user to set the name and password on the client.

You can create the Credential Request starting point with a Credential screen automatically when you initially create a new mobile workflow, or you can create the Credential Request starting point and associated screen manually. This procedure shows how to create the Credential Request starting point automatically when you create a new mobile workflow.

1. In the Mobile Development perspective, select **File > New > Mobile Workflow Forms Editor**.

2. Follow the instructions in the New Mobile Workflow Forms Editor wizard:

| Field | Description |
|---|---|
| Enter or select the parent folder | Select the mobile application project in which to create the mobile workflow form. |
| File name | Enter a name for the mobile workflow form. The extension for mobile workflow forms is .xbw. |
| Advanced | Link the mobile workflow form to an existing file in the file system. |
| Link to file in the file system | Click **Browse** to locate the file to which to link the mobile workflow form. Linked resources are files or folders that are stored in the file system outside of the project's location. If you link a resource to an editor, when you select the editor, the resource is selected in the Work-Space Navigator. Conversely, when you select the resource in the WorkSpace Navigator, the editor is selected. |

3. In the Starting Points page, select **Credentials (authentication) may be requested dynamically from the client application**.

4. Continue with the New Mobile Workflow wizard as appropriate to create the type of mobile workflow application you want to create. Click **Finish**.

5. When the Mobile Workflow Forms Editor opens, click **Flow Design**.

   The Credential Request starting point and its associated Credential Request screen appear on the Flow Design page.

   Select the Credential Request starting point. You will see the two pre-defined keys (cc_username and cc_password) in the Properties view.

6. Double-click the **Credential Request** screen to go to the Screen Design page.

Two editbox controls, which are bound to the pre-defined cc_username and cc_password keys appear on the screen.

**7.** Select the **Username** editbox, then click **Advanced** on the left side of the Properties view.

The Username editbox has the **Credential cache username** checkbox selected. Select the Password editbox and note that is has the **Credential cache password checkbox** checked.

If you create a Credential Request starting point and screen manually, you must add the editbox controls, create the keys for the username and password, and check the corresponding Credential cache username or password box.

**8.** Select **File > Save**.

The first time the mobile workflow is started following deployment, the credential screen is shown. The username and password values are then cached in the credential cache.

**Note:** If an e-mail triggered workflow form has dynamic cached credentials, the cached credentials are not cached between invocations of the workflow form through an email trigger.

## Creating Keys

Create new keys that can be bound to controls.

**1.** On the Screen Design page, select the control to which you want to bind the new key.

**2.** In General Properties, in the Input Data Binding section, click **New key**.

**3.** In the Key dialog, enter:

| Property | Description |
|---|---|
| Name | Enter a name for the key. |
| Type | Select the key type. |
| Sent by server | Select this checkbox to input data binding for the key. |
| Input Data Binding | |
| MBO attribute | Select to bind the key to a mobile business object (MBO) attribute:<br>• Name<br>• Convert to UTC |
| MBO relationship | Bind the key to an MBO relationship. |
| MBO object query results | Bind the key to MBO object query results. |
| User defined | Bind the key to user-defined data. |

**4.** Click **OK**.

**Keys**

Keys are unique identifiers within the scope of a workflow application, used when setting or retrieving a value.

Keys can be bound to the mobile business object (MBO), whether it is to an MBO attribute, an MBO parameter, an MBO operation parameter, a subscription parameter, an object query, or an MBO relationship (list type keys only).

Keys can be of the following types:

- string
- int
- double
- decimal
- bool
- DateTime
- list – can be bound to relationships, or nothing. Keys of other types cannot be bound to relationships. List type keys can also be bound to MBO object query results, which returns multiple objects. List type keys have one or more child keys to which they are bound.

This example of an .xml file shows how keys are used in a workflow application. In this example, the keys associated with the editbox control are bound to the parameters of the mobile business object's (TravelRequest) create operation.

```
<screens default="TravelRequestcreate">
    <screen key="TravelRequestcreate" text="TravelRequest_create" >
      <actions>
        <action key="Submit" type="submit" text="Create" />
        <action key="Cancel" type="cancel" text="Cancel" />
      </actions>
      <controls>
        <datepicker  key="TravelRequest_create_trvl_date_paramKey"
value="" label="Trvl date:" labelpos="LEFT" maxlength="32767" />
        <editbox  key="TravelRequest_create_trvl_location_paramKey"
value="" label="Trvl location:" labelpos="LEFT" maxlength="20" />
        <editbox  key="TravelRequest_create_est_cost_paramKey"
value="" label="Est cost:" labelpos="LEFT" numeric="true"
maxlength="32767" />
        <editbox  key="TravelRequest_create_purpose_paramKey"
value="" label="Purpose:" labelpos="LEFT" maxlength="200" />
      </controls>
    </screen>
</screens>
```

**Binding Controls to Keys**

Use the Properties view to bind keys to controls.

**1.** On the Screen Design page, select the control to which you want to bind the key.

**2.** In General Properties, in the Input Data Binding section, select the key with which to bind the control from the drop-down list.

**3.** (Optional) In Default Value, enter a default value for the key.

**4.** Select **File > Save**.

### Specifying Values for Session Personalization Keys

If you have a mobile business object with load parameters mapped to session personalization keys, you can specify values for those personalization keys when you invoke an operation or object query from the Mobile Workflow package.

This is useful for loading only specific data into the consolidated database (CDB) from large data sources.

**1.** Create a mobile business object that has load parameters. (Set the load parameters in the MBO Properties view, from the Load Parameters tab.)

**2.** Create a corresponding session personalization key for each load parameter.

**3.** Map each load parameter to the corresponding personalization key.

**4.** In the Mobile Workflow Forms Editor, map the personalization keys to the Mobile Workflow keys for the menu item (in the same way you map parameters to Mobile Workflow keys).

You can also fill the personalization key values from values extracted from the e-mail message, depending on from where you are invoking the object query.

# Configuring a Mobile Workflow Package for Online Mobile Business Object Data Access

Bind Personalization key values or the findByParameter object query to mobile workflow keys to control caching of mobile business object (MBO) data.

**See also**
- *The Mobile Workflow Forms Editor* on page 253
- *Starting Points* on page 267
- *Mobile Business Object Relationships* on page 282
- *Creating Screens Automatically by Dragging and Dropping a Data Source* on page 282
- *Adding Controls to the Screen* on page 283
- *Configuring Control Properties* on page 294
- *Adding Connections to Screens* on page 306
- *Activating the Workflow* on page 309
- *Localization* on page 309
- *Generating the Files For a Mobile Workflow Package* on page 315

## Binding the findByParameter Object Query to a Menu Action

For Synchronous, real-time data access, define a Submit menu action and bind it to the findByParameter object query.

### Prerequisites

You must have propagate-to attributes mapped to MBO load parameters, and the deployed MBO must use an Online cache group policy. Unwired Platform services must be running.

### Task

1. From Unwired WorkSpace, launch the Mobile Workflow Forms Editor. You can use either server-initiated or client-initiated starting points.
2. From the Flow Design screen Double-click the screen for which you are defining a mapping to open it in the Screen Design tab.

   For example, you could have a client-initiated starting point with a Start screen that connects to the Online Data screen.
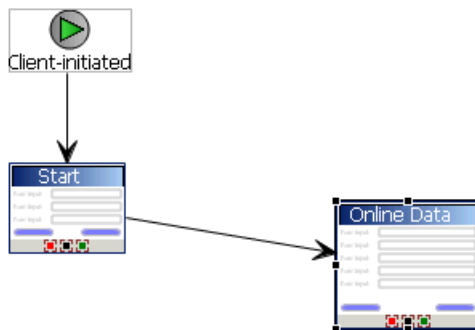


3. Highlight the menu item you want to map, or create a new menu item.
4. Define a Submit action that invokes the findByParameter object query:
   a) From the Communication Style tab, select **Synchronous**.
   b) From the General tab, enter **Submit** as the Type.
   c) In the Details section, select **Search** to locate the MBO that contains the findByParameter object query.
   d) Click the **General** tab, select **Invoke object query** and select **findByParameter**.

If you select the Parameter Mappings tab, you see all the load parameters defined for the MBO and used to generate the findByParameter object query. In addition to Key, you can map parameters to BackEndPassword, BackEndUser, DeviceId, DeviceName, DeviceType, UserName, MessageId, ModuleName, ModuleVersion, and QueueId.

Unmapped parameters can get their value from the default value, if specified, or from the personalization key value they are mapped to, if that is specified. If the key is unmapped, and the parameter has no default value and is not mapped to a personalization key value, the parameter value is empty (NULL for string, 0 for numeric, and so on).

## Binding Transient Personalization Keys to Mobile Workflow Keys

Use transient personalization key values to determine the data to be cached.

**Prerequisites**

You must have transient personalization keys mapped to Mobile Business Object load parameters.

**Task**

1. Create a new mobile workflow package and launch the Mobile Workflow Forms Editor from Unwired WorkSpace:

   a) Select **File > New > Mobile Workflow Forms Editor**.
   b) Select the parent folder that contains the MBO with a load parameter mapped to a transient personalization key. Name the file and click **Next**.

    c) Select **Responds to server-driven email notifications** from the Starting Points screen and click **Next**.

    d) Select the MBO that contains the load parameter to transient key mapping in the Search for MBO screen and click **OK**, then click **Next**.

    e) Specify sample e-mail contents and click **Next**.

    f) Specify the matching rules used to trigger a screen flow by highlighting the text, right-clicking it, and selecting **Select as matching rule**.

    g) Click **Finish**.

**2.** In the Mobile Workflow Forms Editor, map the personalization keys to the Mobile Workflow keys for the menu item:

    a) From the Flow Design screen select the operation for which you are defining a mapping.

    b) Select the Screen Design tab, and highlight the menu item you want to map.

    c) Select **Personalization Key Mappings**, click **Add**, and select a personalization key from the drop-down list and the key to which it maps.
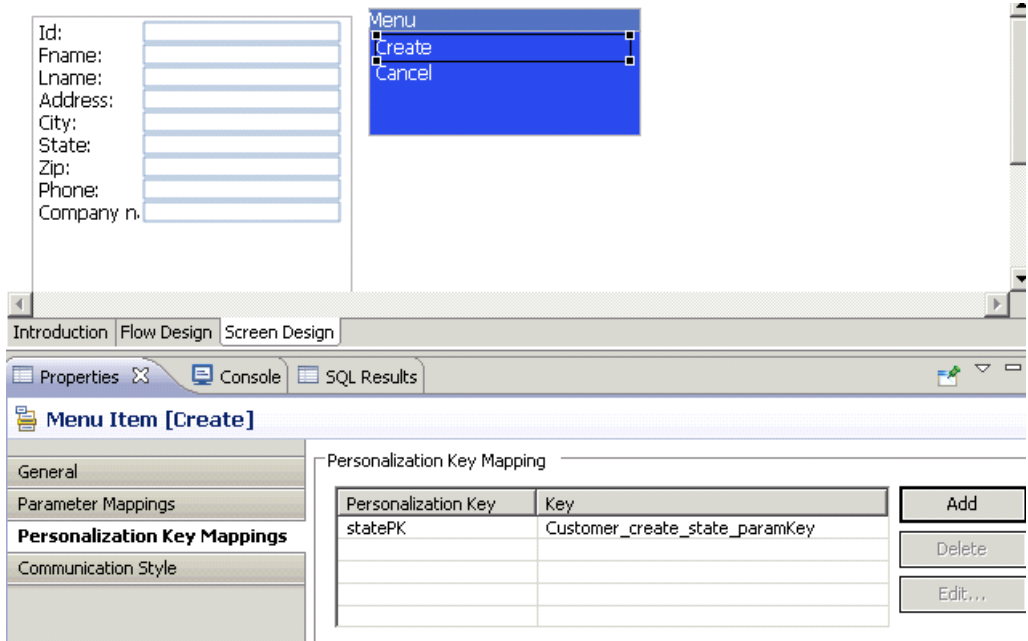


You can also fill the personalization key values from values extracted from the e-mail, depending on from where you are invoking the object query.

When the application runs, the values are sent from the client which are used to fill the load parameter values, and determine what data is cached in the Unwired Server cache (CDB) and returned to the client.

## Mobile Business Object Relationships

When you are using two mobile business objects with a one to many or one to one composite relationship defined, use the listview control on the mobile workflow application to show the list of related objects.

The listview control allows you to navigate the one to many or one to one relationship between the two mobile business objects. The one to one cascade relationships use regular controls or keys that bind to the related mobile business object's attributes.

When you create the relationship between the two mobile business objects, you must select the One to many or One to one and Composite options.

**Note:** Structured mobile business objects and complex parameters are not supported in mobile workflow forms.

**See also**
* *The Mobile Workflow Forms Editor* on page 253
* *Starting Points* on page 267
* *Configuring a Mobile Workflow Package for Online Mobile Business Object Data Access* on page 278
* *Creating Screens Automatically by Dragging and Dropping a Data Source* on page 282
* *Adding Controls to the Screen* on page 283
* *Configuring Control Properties* on page 294
* *Adding Connections to Screens* on page 306
* *Activating the Workflow* on page 309
* *Localization* on page 309
* *Generating the Files For a Mobile Workflow Package* on page 315

## Creating Screens Automatically by Dragging and Dropping a Data Source

Drag and drop a data source, for example a mobile business object (MBO) or operation onto the Flow Design page to create screens, controls, keys, menus, and navigations automatically.

**Note:** Mobile business objects used as the data source must be contained in a deployed mobile application package that is identified as "messaging."

When you use an MBO operation, a screen is created with a menu item that invokes that operation. Controls bound to keys that are bound to that operation's parameters are also created.

When you use an MBO as the data source, screens and keys for the operations are created, along with a details screen for the MBO itself, and also screens for any related MBOs and their operations.

1. From the Flow Design page palette, select a starting point and click on the Flow Design page.
2. From WorkSpace Navigator, expand the project that contains the data source you want to use.
3. Select the data source, for example, a mobile business object, and drag and drop it onto the Flow Design page.

   If the mobile business object you are using as the data source has relationships with other mobile business objects defined, all of the screens and connections between screens for the initial mobile business object and its relationships are automatically created. You can change the order of the screens and create your own connections manually.

   **Note:** Structured mobile business objects and complex parameters are not supported in mobile workflow forms.

**See also**
- *The Mobile Workflow Forms Editor* on page 253
- *Starting Points* on page 267
- *Configuring a Mobile Workflow Package for Online Mobile Business Object Data Access* on page 278
- *Mobile Business Object Relationships* on page 282
- *Adding Controls to the Screen* on page 283
- *Configuring Control Properties* on page 294
- *Adding Connections to Screens* on page 306
- *Activating the Workflow* on page 309
- *Localization* on page 309
- *Generating the Files For a Mobile Workflow Package* on page 315

# Adding Controls to the Screen

Use the Screen Design palette to add controls to the mobile workflow screen.

1. In the Mobile Workflow Forms Editor, select **Screen Design > Palette > Controls**, then select the control you want to add to the screen and click on the screen.
2. Click the control to select it, then use the Properties page to configure it.

**See also**
- *The Mobile Workflow Forms Editor* on page 253

## Mobile Workflow Controls

Controls are objects that users can interact with to view, enter, or manipulate data

Supported controls are determined by the selected device. The set of supported controls depends on the target device platform.

| Control | Supported Platform |
|---------|--------------------|
| Checkbox | <ul><li>Apple iOS</li><li>Windows Mobile Professional</li></ul> |
| Editbox | <ul><li>Apple iOS</li><li>Windows Mobile Professional</li></ul> |
| Choice | <ul><li>Apple iOS</li><li>Windows Mobile Professional</li></ul> |
| Slider | <ul><li>Apple iOS</li><li>Windows Mobile Professional</li></ul> |
| Signature | <ul><li>Windows Mobile Professional</li></ul> |
| HtmlView | <ul><li>Apple iOS</li><li>Windows Mobile Professional</li></ul> |
| Listview | <ul><li>Apple iOS</li><li>Windows Mobile Professional</li></ul> |

## Adding a Menu Item

Add a menu item to the menu.

If you manually create a menu item that invokes an object query, it will automatically create an output key, which is a list and which has keys that are bound to the attributes of that mobile business object.

1. On the Screen Design page, from Palette, select **Menuitem**, then click the Menu box.
2. Select the menu item to configure the properties in the Properties view.

### Menu Items

Menu items function as actions and are associated with mobile business object operations.

Menu items can be one of these types:

- Open – must be associated with another screen. The open menu item is created automatically when a Go To connection is created on the Flow Design page.
- Save – allows the user to close the current open screen and save the input made to the collection of response values to be sent to the server using the submit action. Using the save menu item, a user can navigate back one single screen at a time.
- Cancel – allows the user to close the current open screen without performing any data validation. The cancel action type immediately navigates the user back to the underlying screen.
- Submit (synchronous) – the submit menu item on a screen must be associated with a mobile business object operation or an object query. If the user has multiple screens open during the submit execution, and one of the underlying screens fails validation, the user is prompted with the configured warning message and the submit process stops.
- Submit (asynchronous) – the submit menu item on a screen must be associated with a mobile business object operation.
- Close – allows the user to leave the client-side component no matter which screen the client-side component currently shows.
- Listview – includes AddListView Row, Delete ListView Row, UpdateListView Row, which allows the user to add, delete, or update a listview row. The AddListView Row menu type allows you to navigate directly from the listview details screen and perform an add from the subsequent screen.
- Add – available for ListView controls. The Add menu item type allows you to navigate directly from the listview itself and perform an add from the subsequent screen.

**Menu Item Properties**

Select the menu item on the Screen Design page to view and configure its properties using the Properties page.

*General*

| Property | Description |
|----------|-------------|
| Name | Enter a valid name for the menu item. |
| Key | Select the key with which to bind the menu item. |
| Default | Select to display the menu item prominently on the screen. |

| Property | Description |
| --- | --- |
| Type | Select the type of action the menu item will perform when selected:<br><br>• Save – allows the user to close the current open screen and save the input made to the collection of response values to be sent to the server.<br>• Close – allows the user to leave the client-side component no matter which screen the client-side component currently shows.<br>• Cancel – allows the user to close the current open screen without performing any data validation. The cancel action type immediately navigates the user back to the underlying screen.<br>• Open – indicates the screen to be opened with a GoTo connection. The Open type menu item must be associated with a screen on the Flow Design.<br>• Submit – Closes the client application and invokes a mobile business object method using the values of the current flow.<br>• The Listview actions can be invoked only within the context of a Listview, for example, on a screen that is arrived at after navigating from a listview. It does not immediately execute the MBO create operation in question, but rather stores it so that later, when the parent MBO is submitted, this, and other, MBO operations will be invoked.<br>  • Add Listview Row<br>  • Update Listview Row<br>  • Delete Listview Row<br>• Add – available for ListView controls. The Add menu item type allows you to navigate directly from the listview itself and perform an add from the subsequent screen. |

| Property | Description |
|---|---|
| Details | You can configure menu item actions in the Details section. The properties available depend on the action type. See the topics for configuring the different menu item types for information about configuring these properties. |

*Parameter Mappings*

Use the Parameter Mapping section to bind the parameters (if any) of an operation or object query to keys or, context data, such as DeviceName, BackEndPassword, BackEndUser, and so on.
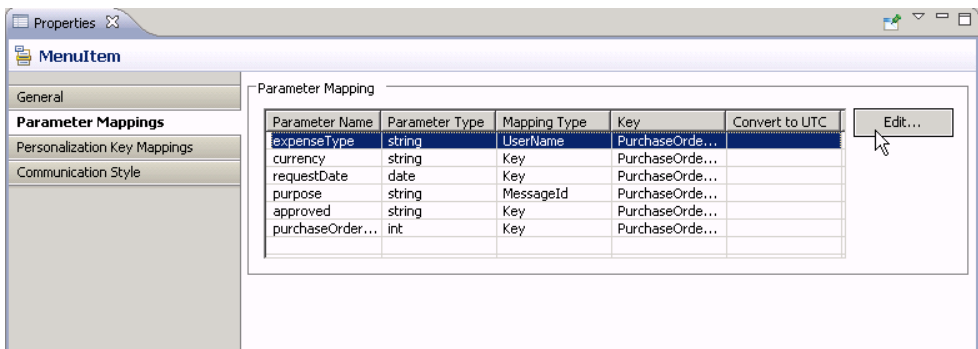


*Communication Style*

| Property | Description |
|---|---|
| Asynchronous | Validates the data input on all open screens, closes the application, and sends the data to the server. Enter the confirmation message to submit to the server. |

| Property | Description |
|---|---|
| Synchronous | Validates the data input on all open screens, saves it, send it to the server and waits for a response from the server before continuing. Enter: <br><br> • Submit error message – enter the message the user sees if there is an error. <br> • Timeout – enter the allowed amount of time to wait for the server to respond before timing out and giving an error message. <br> • Success screen – select the screen to go to upon successful communication with the server. <br> • Error screen – select the screen to go to if there is an error when submitting the data to the server. |

### Creating a Save, Open, Close, or Cancel Menu Item

Create a menu item that performs a save, open, close, or cancel action on the device application screens.

**Note:** The first screen in each mobile workflow form should have a Close or Cancel menu action assigned to it so that the user can exit from the mobile workflow form on the device. On a Windows Mobile device, the user can exit by clicking the OK button, but not all devices have that function.

1. On the Screen Design page, from Palette, click **Menuitem**, then click the Menu box.
2. Select the menu item to configure the properties in the Properties view.
3. In the Properties view:

| Property | Description |
|---|---|
| Name | Enter a valid name for the menu item. |
| Key | Select the key with which to bind the menu item. |
| Default | Select to display the menu item prominently on the screen. |

| Property | Description |
|---|---|
| Type | Select the type of action the menu item will perform when selected:<br>• Save – allows the user to close the current open screen and save the input made to the collection of response values to be sent to the server.<br>• Close – allows the user to leave the client-side component no matter which screen the client-side component currently shows.<br>• Cancel – allows the user to close the current open screen without performing any data validation. The cancel action type immediately navigates the user back to the underlying screen.<br>• Open – indicates the screen to be opened with a GoTo connection. The Open type menu item must be associated with a screen on the Flow Design. |
| Screen | This field appears in the Details section only if the menu item type is Open. Select the screen that will open from the GoTo connection. |

**4.** Select **File > Save**.

### Creating a Submit Menu Item
Create a "submit" menu item.

The submit menu item of the first screen in a mobile workflow form must be associated with a mobile business object operation or an object query.

If the user has multiple screens open at the time of the submit execution, and one of the underlying screens fails validation, the user is prompted with the configured warning message and the submit process stops.

**1.** From the palette on the Screen Design page, click **Menuitem**, then click the Menu box.

**2.** Select the menu item to configure the properties in the Properties view.

**3.** In the Properties view, configure the properties for the submit menu item:

| Property | Description |
|---|---|
| Name | Enter a valid name for the menu item. |
| Key | Select the key with which to bind the submit menu item. |

| Property | Description |
|---|---|
| Default | Select to display the menu item prominently on the screen. |
| Type | Select **Submit** from the drop-down list. |

**4.** If the menu item is an operation associated with a mobile business object, you can configure these properties in Details:

| Property | Description |
|---|---|
| Mobile business object | The mobile business object with which the menu item is associated. Click **Search** to locate the mobile business object with which to associate the menu item. |
| Invoke parent update | No changes are made to the parent MBO, but the appropriate operations (create, update, delete) are performed on its descendants. |
| Invoke operation | Invoke the selected mobile business object operation with which the menu item is associated. Select the operation from the drop-down list. The list is populated when you select the mobile business object. |
| Invoke object query | Invoke the object query from the menu item. **Note:** The menu item must be the Submit type menu item and the communication style must be synchronous. |

Configure these properties in Advanced:

| Property | Description |
|---|---|
| Flow-specific username key | User name with which to connect when invoking this operation. |
| Flow-specific password key | User password with which to connect when invoking this operation. |

**5.** Select **File > Save**.

*Configuring Communication Style*
Choose whether to make the submit action for the menu item synchronous or asynchronous.

**1.** In the Properties view for the menu item, select the **Communication Style** tab.
**2.** Select from the options:

| Property | Description |
|---|---|
| Asynchronous | Validates the data input on all open screens, closes the application, and sends the data to the server. Enter the confirmation message to submit to the server. |
| Synchronous | Validates the data input on all open screens, saves it, sends it to the server and waits for a response from the server before continuing. |
| | **Note:** When executing a synchronous Submit menu item, then subsequently closing the application, it is not treated as if it was processed. It is only treated as having been processed if the application was closed through an asynchronous Submit menuitem. |
| | Enter: |
| | • Submit error message – enter the message the user sees if there is an error. When an operation is invoked, the user can receive any error messages that may occur on the client application. The error messages are a single key of the list type with a single key of the string type as its child. Each of the child instances corresponds to an error message. |
| | • Timeout – enter the allowed amount of time to wait for the server to respond before timing out and giving an error message. |
| | • Success screen – select the screen to go to upon successful communication with the server. |
| | • Error screen – select the screen to go to if there is an error when submitting the data to the server. |
| | Click **Generate Error Screen** to generate error screens automatically. |

**3.** Select **File > Save**.

### Creating a ListView Menu Item
Create and configure a Listview menu item.

**1.** In the Screen Design page, from the palette, click **Menuitem**, then click the Menu box.
**2.** Click the menu item to select it and configure the properties in the Properties view.

**3.** In the Properties view:

| Property | Description |
|----------|-------------|
| Name | Enter a valid name for the menu item. |
| Key | Select the key with which to bind the Listview menu item. |
| Default | Select to display the menu item prominently on the screen. |
| Type | Select one of these types:<br>• Add Listview Row<br>• Update Listview Row<br>• Delete Listview Row |

**4.** If the menu item is an operation associated with a mobile business object, configure these properties in Details.

| Property | Description |
|----------|-------------|
| Mobile business object | The mobile business object with which the menu item is associated. Click **Search** to locate the mobile business object with which to associate the menu item. |
| Operation | The mobile business object operation with which the menu item is associated. Select the operation from the list, which is populated when you select the mobile business object. |
| Parameter mapping | Parameter mapping for the selected operation is shown. Click **Edit** to edit the parameter mapping for the operation. |
| ListView | Select the Listview from the drop-down list. |

**5.** Select **File > Save**.

## Configuring Parameter Mappings

Configure mapping between parameters and keys or context data.

Parameter mappings for the selected operation or object query are shown in the Properties view.

**1.** In the Properties view for the menu item, select the **Parameters Mapping** tab.

**2.** Click **Add** to map a key or context data, such as BackEndUser, BackEndPassword, DeviceName, UserName, MessageID, and so on to a parameter.

**3.** Select a key, and click **Edit** to edit mapping between a parameter and a key or context data.

**4.** Select a key and click **Delete** to delete existing mapping between a key or context data and a parameter.

# Configuring Control Properties

Use the Properties page of any control to configure control properties.

**1.** On the Screen Design page, select the control for which to configure the properties. The configurable properties for the selected control appear in the Properties view.

**2.** Configure the properties for the control and select **File > Save**.

**See also**
- *The Mobile Workflow Forms Editor* on page 253
- *Starting Points* on page 267
- *Configuring a Mobile Workflow Package for Online Mobile Business Object Data Access* on page 278
- *Mobile Business Object Relationships* on page 282
- *Creating Screens Automatically by Dragging and Dropping a Data Source* on page 282
- *Adding Controls to the Screen* on page 283
- *Adding Connections to Screens* on page 306
- *Activating the Workflow* on page 309
- *Localization* on page 309
- *Generating the Files For a Mobile Workflow Package* on page 315

## Checkbox Properties

Select the checkbox on the Screen Design page to view and configure its properties.

*General control properties*
These are general properties for most controls.

| Property | Description |
|----------|-------------|
| Label | This property applies to all controls except HtmlView and Listview.<br><br>• Label – enter the text that describes the control on the user interface.<br>• Label position – the position of the label in relation to the control it describes. |

| Property | Description |
|---|---|
| Input data binding | • Key – the unique identifier within the scope of the widget, used when setting or retrieving a value.<br>Click **New key** to create a new key for the control.<br>• Default value – default value for the control. |
| Read only | This property applies to all controls except HtmlView and Listview. Select to make the control read-only. |

*Advanced*

These are the advanced validation properties for the checkbox control.

| Property | Description |
|---|---|
| Required | Specifies that submission is possible only when the checkbox is selected. A submission must have a value. |
| Validation message | Specifies the error message that appears to the user in case the input validation criteria do not match. |

## Editbox Properties

Select Editbox on the Screen Design page to view and configure its properties.

*General control properties*

These are general properties for most controls.

| Property | Description |
|---|---|
| Label | This property applies to all controls except HtmlView and Listview.<br><br>• Label – enter the text that describes the control on the user interface.<br>• Label position – the position of the label in relation to the control it describes. |

| Property | Description |
|---|---|
| Input data binding | <ul><li>Key – the unique identifier within the scope of the widget, used when setting or retrieving a value.<br>Click **New key** to create a new key for the control.</li><li>Default value – default value for the control.</li></ul> |
| Read only | This property applies to all controls except HtmlView and Listview. Select to make the control read-only. |

*Editbox general properties*

These are the general properties that apply to the editbox control.

| Property | Description |
|---|---|
| Password | If selected (set to true), text entered in the editbox is replaced with asterisks. The default is false. |

| Property | Description |
|---|---|
| Logical type | Select the logical type for the editbox:<br><br>• TEXT<br>• NUMERIC<br>• DATE – uses absolute values or values relative to the current day using simple calculations in days, for example, "2008-05-24" "today" "today+1" "today-1." All values are expected to be in UTC.<br>• DATETIME – uses absolute values or values relative to the current time or day using simple calculations in seconds, for example, "2008-05-24" or "2008-05-24T14:30:15" "today" "today +86400" or "today-86400." In "today+nnn," the nnn means seconds and "today" is interpreted as the current date time 12:00PM.<br>• TIME – uses only absolute values, for example, "14:30" or "14:30:15." Valid values are 00:00:00 to 23.59:59.<br><br>**Note:** The properties that you can configure depend on the logical type set for the editbox control. Some properties apply only to the numeric type, some apply only to the text type, and so on. |

*Advanced*

These are the advanced properties that apply to the editbox.

| Property | Description |
|---|---|
| Content size | • Maximum length – specify the maximum number of characters allowed for the control.<br>• Number of lines – specify the number of lines for the control.<br>• Number of decimals – specify the number of decimals (the number of digits after the decimal point—valid range is 0 – 28) for a number. Valid only in conjunction with numeric attributes. The default is zero. |

| Property | Description |
|---|---|
| Validation | • Minimum value – specify the minimal value for numeric input. Valid only in conjunction with numeric attributes.<br>• Maximum value – specify the maximum value for numeric input. Valid only in conjunction with numeric attributes.<br>• Required – specify if user input is required. A value is required for submission.<br>• Validation regular expression – if the user enters text in the control, the input has to comply to the provided regular expression.<br>• Validation message – enter the error message the user sees in case the input validation criteria do not match. |
| Time precision | Specifies the precision of the time portion displayed to the user and calculations.<br><br>**Note:** This property is used only if the editbox logical type is datetime or time. |
| Credential cache username | If the widget supports cached credentials, select to specify that the editbox be treated as a user name. |
| Credential cache password | If the widget supports cached credentials, select to specify that the editbox be treated as a password. |

## Choice Properties

Select the choice control on the Screen Design page to view and configure its properties.

*General control properties*
These are general properties for most controls.

| Property | Description |
|---|---|
| Label | This property applies to all controls except HtmlView and Listview.<br><br>• Label – enter the text that describes the control on the user interface.<br>• Label position – the position of the label in relation to the control it describes. |
| Input data binding | • Key – the unique identifier within the scope of the widget, used when setting or retrieving a value.<br>Click **New key** to create a new key for the control.<br>• Default value – default value for the control. |
| Read only | This property applies to all controls except HtmlView and Listview. Select to make the control read-only. |

*Choice general properties*
These are the general properties that apply to the choice control.

| Property | Description |
|---|---|
| Logical type | Select the logical type for the edit box from the drop-down list. Logical types include:<br><br>• TEXT<br>• NUMERIC<br><br>**Note:** The properties that you can configure depend on the logical type set for the control. Some properties only apply to the numeric type and some only apply to the text type. |
| Options | • Static – the choice contents are static. Click **Add** to assign a display name and value pair to the choice item.<br>• Dynamic – the choice contents are dynamic. |

*Advanced*
These are the advanced properties that apply to the choice control.

| Property | Description |
|---|---|
| Content size | Number of decimals – specify the number of decimals (the number of digits after the decimal point—valid range is 0 to 28) for a number. Only valid in conjunction with the numeric attributes. The default is zero. |
|  | **Note:** This property is used only if the logical type for the choice control is set to numeric. |
| Validation required | • Required – specify if user input is required. A submission without a value is not possible. <br> • Validation message – enter the error message displayed to the user in case the input validation criteria do not match. |

### Adding Choice Items to the Choice Control

Add choice items to the choice control.

1. In the General Properties view for the choice control, click **Add**.
2. In the Choice Option Item dialog, enter the Display Name and Value for the choice item, and click **OK**.

| Property | Description |
|---|---|
| Display Name | Enter the name for the choice item as it will appear on the user interface. |
| Value | Enter the value for the choice item. |

### Editing Choice Items

Edit choice items for the choice control.

1. In the General Properties view for the choice control, select the choice control item to edit and click **Edit**.
2. In the Choice Option Item dialog, make the changes and click **OK**.

| Property | Description |
|---|---|
| Display Name | Enter the name for the choice item as it will appear on the user interface. |
| Value | Enter the value for the choice item. |

**Deleting Choice Items**

Delete choice items from the choice control.

1. In the General Properties view for the choice control, select the choice control item to delete.
2. Click **Delete**.

**Creating Dynamic Contents for the Choice Control**

Specify contents of the choice control dynamically.

1. In the Screen Design page, select the Choice item from **Palette > Controls** and drag it onto the screen.
2. In the General Properties view for the choice control, from the Options section, select **Dynamic**.
3. In **Value key**, enter the value for the key.
4. In **Display name key**, enter the value that the user will see on the device interface.

**Note:** The display name key and the value name key can be different or the same. The two keys must be children of a key that is a list type.

## Slider Properties

Select the slider control on the Screen Design page to view and configure its properties.

*General control properties*

These are general properties for most controls.

| Property | Description |
| --- | --- |
| Label | This property applies to all controls except HtmlView and Listview.<br><br>• Label – enter the text that describes the control on the user interface.<br>• Label position – the position of the label in relation to the control it describes. |

| Property | Description |
|---|---|
| Input data binding | • Key – the unique identifier within the scope of the widget, used when setting or retrieving a value.<br>Click **New key** to create a new key for the control.<br>• Default value – default value for the control. |
| Read only | This property applies to all controls except HtmlView and Listview. Select to make the control read-only. |

*Slider control general properties*
These are the general properties that apply to the slider control.

**Table 12. Value constraints**

| Property | Description |
|---|---|
| Minimum value | Specify the minimum value. |
| Maximum value | Specify the maximum value. |

*Advanced*
These are the advanced properties that apply to the slider control.

| Property | Description |
|---|---|
| Validation message | Specifies the error message displayed to the user in case the input validation criteria do not match. |

## Signature Properties

Select the signature control on the Screen Design page to view and configure its properties.

**Note:** The signature control is not supported on Apple devices.

*General control properties*
These are general properties for most controls.

| Property | Description |
|---|---|
| Label | This property applies to all controls except HtmlView and Listview.<br><br>• Label – enter the text that describes the control on the user interface.<br>• Label position – the position of the label in relation to the control it describes. |
| Input data binding | • Key – the unique identifier within the scope of the widget, used when setting or retrieving a value.<br>Click **New key** to create a new key for the control.<br>• Default value – default value for the control. |
| Read only | This property applies to all controls except HtmlView and Listview. Select to make the control read-only. |

*Advanced*

These are the advanced properties that apply to the signature control.

| Property | Description |
|---|---|
| Validation Required | • Required – specify if user input is required. A submission without a value is not possible.<br>• Validation message – enter the error message displayed to the user in case the input validation criteria do not match. |

## HtmlView Properties

Select the HtmlView control on the Screen Design page to view and configure its properties.

The HtmlView control fills the entire screen of a workflow form, so no other controls can be placed on a screen that hosts an HtmlView control.

**Table 13. HtmlView control properties**

| Property | Description |
|---|---|
| Input data binding | • Key – the unique identifier within the scope of the workflow form, used when setting or retrieving a value. Click **New key** to create a new key for the control. <br>• Default value – default value for the control. |

## Listview Properties

Select Listview on the Screen Design page to view and configure its properties.

*General Listview properties*

| Property | Description |
|---|---|
| Input data binding | Select the key with which to bind the Listview control from the drop-down, or click **New key** to create a new key. |
| Alternate row color | Click **Color Picker** to select the color for the alternate rows in the list. <br><br>You can also manually enter the value. The possible values are black, blue, brown, gray, green, orange, pink, purple, red, white, yellow, any string with the format #rrggbb, where rr, gg and bb are the hexadecimal values, in the range 0 to ff(255), for the red, green and blue. <br><br>**Note:** The alternate row color attribute is supported on Windows Mobile devices only. |
| On empty list | The text that appears on the screen when the Listview is empty. |
| Listview details screen | Select the screen that will be opened when the user clicks a row. |

*Cells*

| Property | Description |
|----------|-------------|
| Cell lines | The lines that appear in each row of the control. Click **Add** to add a new line. Select an existing line and click **Delete** to remove a line from the row. |
| Fields for | Click **Add** to create a new field. Click **Edit** to edit an existing field. Click **Delete** to delete an existing field. |

### Adding a Listview Field to a Line

Use the Properties view to add fields to Listview lines.

1. Select the **Listview** control to see the Properties view.
2. In Properties, in the left pane, click **Lines**.
3. In **Listview lines**, select the line for which to add the field.
4. In **Fields for** click **Add**.
5. In the Listview Field dialog:

| Option | Description |
|--------|-------------|
| Key | Select the key with which to associate the field. |
| Field width | Enter the field width. This value is a percentage, so the values entered should be between 1 – 100 and add up to 100. |
| Font options | Select the appearance of the font:<br>• Bold<br>• Italics |
| Field format | Enter the field format. A standard format string contains a format specifier, which is an alphabetic character, for example, G, that defines the string that represents the object to which it is applied.<br><br>**Note:** This attribute is supported only on Windows Mobile Platforms. |

6. Click **OK**.

<u>**Editing a ListView Field**</u>

Edit ListView control fields in the Properties view.

1. Select the ListView control to see the Properties view.
2. In Properties, in the left pane, click **Lines**.
3. In Fields for ListView Line, select the line for which to edit the field and click **Edit**.
4. In the ListView Field dialog, make your changes and click **OK**.

# Adding Connections to Screens

The Start icon  appears on the Flow Design page, and its connection is attached to the first screen you add to the Flow Design. You can change the target to any screen in the Flow Design.

1. From the Flow Design Palette, select the connection you want to use:

   **Note:** Some connections cannot be explicitly created; they can only be implicitly created when creating or editing menu item properties.

   | Connection | Description |
   |---|---|
   | Go to | Specifies which screen the user goes to upon completing an action on the current screen, such as clicking OK or Next. |
   | Operation success | Specifies which screen the user goes to upon successful completion of an operation. |
   | Operation error | Specifies the screen that the user goes to automatically when an operation fails to execute. |
   | Listview details | Specifies which screen to open when a row in a listview is clicked. Once on the details screen you can add, update, or delete rows from the listview. |
   | Listview add | Specifies a screen that can be directly navigated to in order to add a new row to the listview. |

2. In the Flow Design canvas, click the source screen and drag to connect it to the target screen.
   The connection line appears between the two screens.

**See also**
- *The Mobile Workflow Forms Editor* on page 253

## Connections

Use connections to connect screens with actions in the Flow Design.

Connection actions describe the navigation direction the screen should go after the user performs an action, cancels an action, or uses a control. Alert actions use actions defined in the Flow Design.

**Note:** You can use only one connection of each type from a given source screen, except for the GoTo and Operation Success connection types. You can have as many Operation Success connections on a source screen as you have synchronous operations on the source screen. You can have as many GoTo connection types as you want to on a source screen.

## Connection Properties

Use the Properties view for connections to configure the connection properties.

Select the connection on the Flow Design page to view its properties.

| Property | Description |
|---|---|
| Routing | When a new connection is created, it uses the default style defined in the preferences settings. You can change the style of the connection line individually in the Properties view. |
| | • Styles: |
| |     • Oblique – connection lines are not aligned with the horizontal or vertical axis, and you can create breakpoints anywhere on the connection line. If you drag the connection line, a new breakpoint is created at the drag point and two lines are created. |
| |     • Rectilinear – the connection line always remains either horizontal or vertical. |
| |     • Tree – arranges the connection lines on the Flow Design in a tree structure. |
| | • Avoid obstructions – connection lines go around objects. |
| | • Closest distance – connection lines pass through objects to shorten the distance of the connection line. |
| Lines and arrows | |
| Smoothness | • None<br>• Less<br>• Normal<br>• More |
| Jump links | • Status:<br>    • None<br>    • Below<br>    • All<br>    • Above<br>• Type:<br>    • Semicircle<br>    • Chamfered<br>    • Square<br>• Reverse jump links |

## Activating the Workflow

The last screen must have an asynchronous submit action.

The submit action on a Workflow screen may be either synchronous or asynchronous.

To complete the workflow activation process, the last screen must have an asynchronous submit action. This is necessary in order for the device and sever side to activate the workflow for the device.

### See also
- *The Mobile Workflow Forms Editor* on page 253
- *Starting Points* on page 267
- *Configuring a Mobile Workflow Package for Online Mobile Business Object Data Access* on page 278
- *Mobile Business Object Relationships* on page 282
- *Creating Screens Automatically by Dragging and Dropping a Data Source* on page 282
- *Adding Controls to the Screen* on page 283
- *Configuring Control Properties* on page 294
- *Adding Connections to Screens* on page 306
- *Localization* on page 309
- *Generating the Files For a Mobile Workflow Package* on page 315

## Localization

You can localize different objects in the Mobile Workflow Forms Editor, such as the names of screen controls, screens, and mobile business objects.

You can localize the mobile workflow by creating locale properties files. You can then load, update, and generate localized device applications.

You can create or update resource bundles for the Mobile Workflow Forms Editor using the resource keys in the locale properties file. The values for the resource keys are defined by the corresponding elements in the current Mobile Workflow Forms Editor. You can also upload specific locale resource bundles into the Mobile Workflow Forms Editor.

Some restrictions for the locale properties files apply:

- When you specify a country for the language, the basic language locale must also be available. For example, if you create a locale and specify English as the language and the United States as the country, then a locale for English (the basic language) must also be available.

- If you create a locale that specifies language, country, and variant, the locale for the basic language and the locale for the basic language and the country must be available. For example, if you create a locale and specify English as the language, United States as the country, and WIN as the variant, then English (United States) and English locales must also be available.
- The language code must be a 2-letter code, and the country code can be either a 2-letter or 3-letter code.
- If you specify a variant, the country code must be a 2-letter code.

All locale properties files are saved in the same directory as the device application.

When localizing the Mobile Workflow Forms Editor:

- Mobile workflow applications that have names that begin with numbers or special characters cannot be localized; you will receive an error when you generate the code. Make sure that any mobile workflow you want to localize does not have a file name that begins with a number or special character.
- If you manually localize the locale properties file using an external editor, you must make sure the file is encoded in ASCII, so that the content can be correctly read and converted to Unicode.

**See also**
- *The Mobile Workflow Forms Editor* on page 253
- *Starting Points* on page 267
- *Configuring a Mobile Workflow Package for Online Mobile Business Object Data Access* on page 278
- *Mobile Business Object Relationships* on page 282
- *Creating Screens Automatically by Dragging and Dropping a Data Source* on page 282
- *Adding Controls to the Screen* on page 283
- *Configuring Control Properties* on page 294
- *Adding Connections to Screens* on page 306
- *Activating the Workflow* on page 309
- *Generating the Files For a Mobile Workflow Package* on page 315

## Changing the Encoding Type

**Goal**: Change the encoding type in Preferences.

The localization file is encoded in standard ISO-8859-1. All non-ASCII character values are converted to escaped Unicode hexadecimal values before they are written to the properties files. Before translating the localization file, select the correct file encoding option, for example UTF-8.

**1.** In Sybase Unwired Platform, select **Window > Preferences**.

2. Expand **General > Content Types**.

3. In the right pane, select, **Text > Java Properties File**.

4. In the **File Associations** list, select `*.properties(locked)`.

5. In the Default encoding field, change ISO-8859-1 to **UTF-8**, and click **Update**.
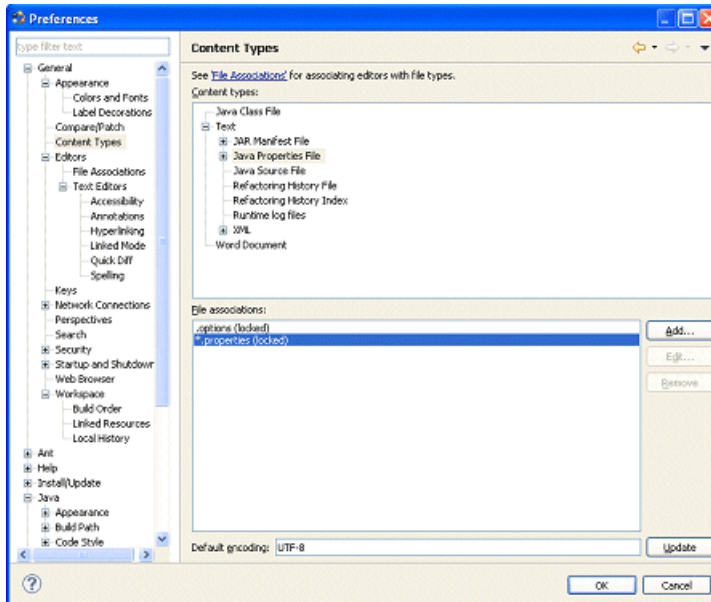


**See also**
- *Creating a New Locale* on page 311
- *Editing a Locale Properties File* on page 313
- *Updating the Current Locale* on page 313
- *Removing a Locale* on page 314
- *Loading a Locale Properties File* on page 314
- *Validating the Locale Properties Files* on page 314

## Creating a New Locale

Create a new locale with resource keys and values extracted from localizable strings in the current device application file.

When you create a new locale, keep in mind:

- When you specify a country for the language, the basic language locale must also be available. For example, if you create a locale and specify English as the language, then there must also be a locale for English (the basic language).

- If you create a locale that specifies language, country, and variant, the locale for the basic language and the locale for the basic language and the country must be available. For example, if you create a locale and specify English as the language, United States as the country, and WIN as the variant, then English (United States) and English locales must also be available.

1. In the Properties view, click **Localization** in the left pane.
2. In the Localization page, click **New**.
3. In the New Locale wizard:

| Option | Description |
|---|---|
| Language | Select the language. |
| Country | Select the country. |
| Variant | Enter the variant, which is the vendor or browser-specific code. For example, enter "WIN" for Windows, "MAC" for Macintosh and "POSIX" for POSIX. If there are two variants, separate them with an underscore, and put the most important one first. For example, a Traditional Spanish collation might construct a locale with parameters for language, country, and variant as: "es", "ES", "Traditional_WIN". |
| Overwrite existing file | Overwrite an existing localization file. |
| Automatically create default locale | Automatically create the default locale properties file. For example, if you specify the language as "English" and the country as the "United States" for a device application called `test`, then both `test_en_uS.properties` and `test.properties` files are created. |

4. Click **Finish**.
   The properties file is created in the same directory as the device application file.

**See also**
- *Changing the Encoding Type* on page 310
- *Editing a Locale Properties File* on page 313
- *Updating the Current Locale* on page 313
- *Removing a Locale* on page 314
- *Loading a Locale Properties File* on page 314
- *Validating the Locale Properties Files* on page 314

## Editing a Locale Properties File

Use the Properties File Editor to edit the localization properties file.

1. In the Screen Design page Properties view, click **Localization**.

2. Select the locale to edit and click **Edit**.

   **Note:** If you use an external editor to manually edit the locale properties file, make sure the file is encoded in ASCII so that the content can be correctly read and converted to Unicode.

3. Make the changes to the file.

4. Select **File > Save**.

### See also
- *Changing the Encoding Type* on page 310
- *Creating a New Locale* on page 311
- *Updating the Current Locale* on page 313
- *Removing a Locale* on page 314
- *Loading a Locale Properties File* on page 314
- *Validating the Locale Properties Files* on page 314

## Updating the Current Locale

Update the currently loaded locale properties file with the resource keys from the current Mobile Workflow Designer.

If the locale properties file does not already exist, it is created. If the current locale is not defined in the mobile workflow application file, the updated locale is used as the default, and the file name is *{device_application}*.properties. Otherwise, the locale defined in the mobile workflow application file is updated.

1. In the Screen Design page Properties view, click **Localization**.

2. Click **Update**.

### See also
- *Changing the Encoding Type* on page 310
- *Creating a New Locale* on page 311
- *Editing a Locale Properties File* on page 313
- *Removing a Locale* on page 314
- *Loading a Locale Properties File* on page 314
- *Validating the Locale Properties Files* on page 314

## Removing a Locale

Remove locale properties files.

1. In the Screen Design page Properties view, click **Localization**.
2. Select the locale to remove and click **Remove**.
3. Click **Yes** to confirm the deletion.

**See also**
- *Changing the Encoding Type* on page 310
- *Creating a New Locale* on page 311
- *Editing a Locale Properties File* on page 313
- *Updating the Current Locale* on page 313
- *Loading a Locale Properties File* on page 314
- *Validating the Locale Properties Files* on page 314

## Loading a Locale Properties File

Load a locale properties file.

1. In the Screen Design page Properties view, click **Localization**.
2. Select the locale properties to load and click **Load**.
   You see a message if there are missing resource keys.
3. (Optional) Click **Yes** to automatically fix the properties file.
   If you choose to not fix the locale properties file, loading does not continue.

   Loading the locale property file updates the current locale of the device application file.

**See also**
- *Changing the Encoding Type* on page 310
- *Creating a New Locale* on page 311
- *Editing a Locale Properties File* on page 313
- *Updating the Current Locale* on page 313
- *Removing a Locale* on page 314
- *Validating the Locale Properties Files* on page 314

## Validating the Locale Properties Files

Validate locale properties files.

Validation verifies that all the resource keys for the current mobile workflow application file exist in the locale properties file.

1. In the Flow Design Properties view, click **Localization**.
2. Select the locale for which to validate the properties file, and click **Validate**.
   You see a message if there are missing resource keys.
3. (Optional) Click **Yes** to automatically fix the properties file.

**See also**
- *Changing the Encoding Type* on page 310
- *Creating a New Locale* on page 311
- *Editing a Locale Properties File* on page 313
- *Updating the Current Locale* on page 313
- *Removing a Locale* on page 314
- *Loading a Locale Properties File* on page 314

# Generating the Files For a Mobile Workflow Package

Use the Mobile Workflow Package Generation wizard to generate the files for the mobile workflow form, optionally deploy the generated workflow package files to the server, and assign it to one or more users' devices.

1. Right-click in either the Flow Design or Screen Design page of the Mobile Workflow Forms Editor and select **Generate Mobile Workflow Package**, or click the code generation icon 📇 on the toolbar.
2. In the Mobile Workflow Package Generation wizard, enter or select:

| Option | Description |
|---|---|
| Favorite configurations | (Optional) Select a configuration. |
| Code Generation and Deployment | |
| Generation folder | Select the folder in which to save the `.zip` file containing the application and its generated files. |
| Deploy to an Unwired Server | Select to deploy the mobile workflow application to an Unwired Server. |
| Unwired Server Profile | Select the Unwired Server profile with which to associate the mobile workflow application and extract the username and password credentials if you are using static authentication. |

| Option | Description |
|---|---|
| Assign workflow to user(s) | The mobile workflow must be assigned to a device user before the mobile workflow is visible on the user's device. It can be assigned to multiple users. Separate multiple users with a comma. Device users must be registered in Sybase Control Center. |
| | Click **Get Users** to select device users from the list. |
| Deploy the Sybase messaging client to a connected Microsoft Windows Mobile device or emulator | Select to deploy the mobile workflow application to a Windows Mobile connected device or emulator. |
| | **Note:** This option requires Microsoft Active-Sync 4.5 (if you are using Windows XP) or Windows Mobile Device Center (Windows Vista, Windows 7, Windows 2008) to be installed and connected. |
| Windows Mobile device type | Select the Mobile Windows device type from the drop-down list. <br> • Windows Mobile Professional |
| Perform silent install | Select to perform the deployment on the device without requiring user input. |
| Deployment timeout (minutes) | Enter the number of minutes allowed before the deployment times out. |
| Start Device Emulator Manager | Start the Device Emulator Manager. The Device Emulator Manager is used to connect and cradle the selected device. |

**3.** (Optional) Click **Start Device Emulator Manager** to launch the Emulator Manager so you can connect to different Windows Mobile device types.

**4.** Click **Finish**.

A .zip file containing the application and its generated files is created and placed in the specified location.

**See also**

- *Adding Controls to the Screen* on page 283
- *Configuring Control Properties* on page 294
- *Adding Connections to Screens* on page 306
- *Activating the Workflow* on page 309
- *Localization* on page 309

# Preparing Devices

To enable deploying Workflow packages to a device, you must download, install, and configure a Workflow container on the device.

Refer to the appropriate procedure for your device type.

## Preparing iPhone Devices for the Workflow Package

Install the Mobile Workflow package on the device using the App Store, or use the sample Workflow container to deploy to the iPhone simulator from the Xcode project.

**See also**
* *Preparing Windows Mobile Devices for the Mobile Workflow Package* on page 322

## Installing the Mobile Workflow Package on the Device Using the Mobile Workflow Container Downloaded From App Store

Install the Mobile Workflow container from the iPhone App Store.

1. On the iPhone home page, tap **App Store**.
2. Search for **Sybase**.
3. When the Sybase Mobile Sales & Workflow application appears, tap **Free**.
4. Tap **Free** again on the Mobile Sales information page.
5. Tap **Install** to download the application.
6. In **Settings > Sybase**, for Connection Info, enter:

    a) In Connection Info, enter:
       * Server Name – your unwiredserver.company.com or your relayserver.company.com
       * Server Port – 5001 your Unwired Server port or relay server port
       * Farm ID – company ID, or Farm ID registered in the relay server
       * User name – the user name you entered for the device in Sybase Control Center
       * Activation Code – the activation code for the device

    b) In Sybase Features, turn the Mobile Workflow option to **On** and the Mobile Sales option to **Off**.
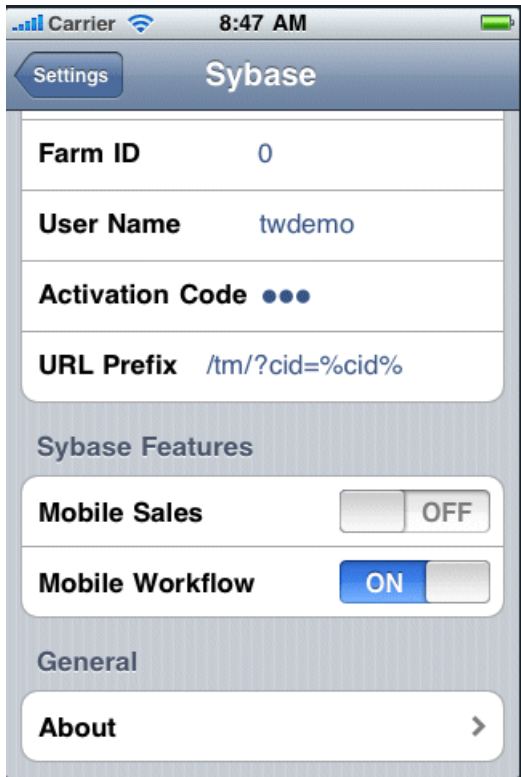
7. Scroll to the page that contains the **Sybase** icon, then tap to launch.
   The **Workflows** page appears.

8. To view the inbox e-mails, click the **Messages** Tab bar, then tap **Messages** to view the Workflows.

## Deploying the Mobile Workflow Package to the iPhone Simulator

Deploy the Mobile Workflow package to an iPhone simulator for testing.

### Prerequisites

- You must have a registered device user in Sybase Control Center to which to assign the mobile workflow. See the Sybase Control Center online help topic *Configure > Configuring Unwired Platform > Device Users > Devices > Messaging Devices*.
- You must have a Mac with iPhone SDK 4.0.x installed.
- Xcode 3.2.3 or higher.
- Generate the Mobile Workflow package.

_____

**Task**

The Workflow Container referenced in this procedure is a sample container. You can use Xcode to build your own customized user interface and configure other resources.

1. Log into Sybase Control Center and assign the mobile workflow package to registered device users.

   See the Sybase Control Center online help topic *Configure > Configuring Unwired Platform > Mobile Workflows > Configuring a Mobile Workflow Package*.

2. Copy these directories from your Sybase Unwired Platform installation `%Sybase%\UnwiredPlatform\Servers\UnwiredServer\ClientAPI\ObjectiveC` to a location on your Mac:

   - `includes`
   - `libs`
   - libs.iPad
   - `images`
   - `strings`
   - `WorkflowContainer`

   You can copy these files into whatever location you choose, but they should all be at the same directory level.

3. In the `WorkflowContainer` folder, unzip `WorkflowContainer-iPhone.zip` into the same directory level in which you copied the other folders.

   This creates a `SUPWorkflows` folder.

4. In the `images` folder, unzip the `WorkflowImages.zip` file in its current location.

   It unzips as a `workflowimages` folder.

5. In the SUPWorkflowContainer folder, double-click **SUPWorkFlows.xcodeproj** so that it opens in the XCode IDE.

6. (Optional) If you are building a project for the iPad simulator, go into the Target settings for the project, and change the **Library Search Paths** to replace `libs` with `libs.iPad`.

7. In XCode, select **Build > Build**.

   The project builds and the simulator starts.

8. In the Settings for the simulator, choose **SUPWorkflows** and verify the connection settings.

   The ServerNameSetting should be the name of the machine on which the Mobile Workflow package is deployed to Unwired Server.
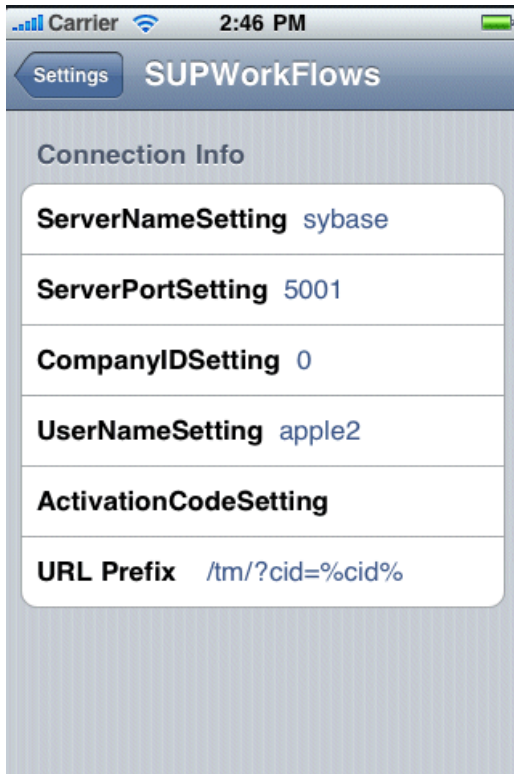
9. In the applications screen, open the SUPWorkFlows applications.
   The deployed Mobile Workflow package is shown in the SUPWorkFlows menu.

# Preparing Windows Mobile Devices for the Mobile Workflow Package

Install and configure the container required to prepare a Windows Mobile device to run Mobile Workflow packages.

1. Navigate to `<UnwiredPlatform_InstallDir>\UnwiredPlatform \Servers\UnwiredServer\ClientAPI\MoMessaging\wm`.
2. Copy the Windows Mobile Professional device file, `SUPMessaging_Pro.cab`, to the device's **My Documents** folder.
3. Cradle the Windows Mobile device.
4. Connect a USB cable between the PC and device, and transfer the .cab file.
5. Open the .cab file from the Windows Mobile device. This installs the container.

6. In the Connection screen, enter the connection settings. These settings should match the values you used when you registered the device in Sybase Control Center.
   - Name/IP Address – your unwiredserver.company.com or your relayserver.company.com
   - Host Port – 5001 your Unwired Server port or relay server port
   - CompanyID –company ID, or Farm ID registered in the relay server
   - User name – the user name you entered for the device in Sybase Control Center
   - Activation Code – the activation code for the device

   **Note:** Select the right arrow icon ( ⭬ ) to view the container log. This is useful for checking the connection, or retrieving other debugging information.

7. Access your Microsoft Outlook inbox on the device. For example:
   - On Windows Mobile Touch Screen devices, tap **Start** > **Messaging**, and then scroll to and tap Outlook Inbox.
   - On Windows Mobile Non-touch Screen devices, click **Start** > **Messaging** on the Home screen. Scroll to a Outlook Inbox and press **ENTER**.

**8.** Select **Menu** > **Workflows**.

**9.** Select a workflow package from the list of available packages.

**See also**
*   *Preparing iPhone Devices for the Workflow Package* on page 319

# Launching a Mobile Workflow on the Device

Launch the deployed Mobile Workflow from the device inbox.

When you select the Workflow menu item in the device inbox, only the latest version of the Workflow appears. If you click the Workflow notification itself, the Workflow version that is associated with the e-mail notification is launched whether it is the latest version or not.

*Example*
You develop a Mobile Workflow application that has both client-initiated and server-initiated starting points. You deploy the initial version, which is called version 1, and a Mobile Workflow e-mail notification is sent.

Next, make some changes and deploy a second version, version 2. Again, a Mobile Workflow e-mail notification is sent.

There are now three ways that this Mobile Workflow application can be launched, and the way that it is launched determines which version of the Workflow is launched:

• If you launch the application from the Workflows menu, the last deployed version of the Workflow, in this case, version 2, is launched. Although version 1 of the Mobile Workflow application still exists somewhere on the device it is never used as long as you launch the Workflow from the Workflows menu.
• If you launch the Workflow by opening the initial e-mail notification, the version that corresponds with the latest version that existed at the time the notification was sent, is used. In this case, that is version 1; it does not matter that a later version (version 2) exists.
• If you launch the Workflow by opening the second notification, the version that corresponds with the latest version that existed at the time the notification was sent is used. In this case, that is version 2.

# Testing Mobile Workflow Packages

To test a Mobile Workflow package:

1. Launch and/or connect to the mobile device or emulator.
2. Deploy the Mobile Workflow package to the device.
3. Establish the connection to the server on the device.
4. For user-initiated Mobile Workflow packages, go to the Workflows menu of the e-mail inbox and click on the appropriate Workflow package.
5. For e-mail subscription Mobile Workflow packages, send the e-mail to the device, either automatically, for example, database trigger, or manually, through the Send E-mail dialog; then open that e-mail on the device.
6. Enter data and execute menu items appropriately.
7. Verify that the backend is updated correctly.
8. Check the logs.

## Testing Server Initiated Mobile Workflow Packages

Test a server-initiated Mobile Workflow package.

1. In the Mobile Workflow Forms editor, open the Mobile Workflow form, `<workflow_form>.xbw`.
2. Click **Flow Design**.
3. Right-click in the editor, and select **Send a notification**.
4. In the Send a Notification window:
   a) Select the Unwired Server profile and click **Get Device Users**.
   b) Choose the desired user and fill in the fields according to the matching rules specified when creating the Mobile Workflow form.
5. Click **Send**.
6. On the client, from the applications screen, open SUPWorkflows.
7. In the client application, click **WorkFlow Messages**. This contains the server-initiated Mobile Workflow form.

# Troubleshoot

Use troubleshooting tips to isolate and resolve common issues.

See *Troubleshooting Sybase Unwired Platform* for information about troubleshooting issues with the Eclipse - based user interface or other Sybase Unwired Platform components.

Troubleshoot

# API Documentation

You can use Sybase Unwired Platform APIs to develop and customize mobile applications, data handling, error handling, and system functionality programmatically.

- **Client Object API** – generated business object classes that represent the mobile business object model built and designed in the Unwired WorkSpace development environment. Use the API to synchronize and retrieve data and invoke mobile business object operations. See the developer references, and the Javadocs included in the installation directory:
  - *Developer Reference for BlackBerry*
  - *Developer Reference for iOS*
  - *Developer Reference for Windows and Windows Mobile*
  - *Developer Reference for Mobile Workflow Packages*
- **Unwired Server API** – custom Java classes used to implement advanced data handling features in Unwired Server. See *Reference: Custom Development for Unwired Server* and the Javadocs included in the installation directory.
- **Administration API** – custom Java classes used to integrate Sybase Unwired Platform system management tools with your enterprise system. See *Reference: Administration APIs* and the Javadocs included in the installation directory.

Check the Sybase Product Documentation Web site regularly for updates: *http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.pubs.docset-SUP-1.5.5/doc/html/title.html*.

**See also**
- *Product Task Flow* on page 1
- *Generating Code For a BlackBerry Device Application* on page 237

# Glossary: Sybase Unwired Platform

Defines terms for all Sybase Unwired Platform components.

**administration perspective** – Or administration console. The Unwired Platform administrative perspective is the Flash-based Web application for managing Unwired Server. *See* Sybase Control Center.

**administrators** – Unwired Platform users to which an administration role has been assigned. A user with the "SUP Administrator" role is called a "platform administrator" and a user with the "SUP Domain Administrator" role is called a "domain administrator". These administration roles must also be assigned SCC administration roles to avoid having to authenticate to Sybase Control Center in addition to Unwired Server:

- A domain administrator only requires the "sccUserRole" role.
- A platform administrator requires both the "sccAdminRole" and "sccUserRole" roles.

**Adobe Flash Player** – Adobe Flash Player is required to run Sybase Control Center. Because of this player, you are required to run Sybase Control Center in a 32-bit browser. Adobe does not support 64-bit browsers.

**Advantage Database Server®** – A relational database management system that provides the messaging database for Sybase Unwired Platform. *See* messaging database.

**Afaria** – An enterprise-grade, highly scalable device management solution with advanced capabilities to ensure that mobile data and devices are up-to-date, reliable, and secure. Afaria includes a server (Afaria Server), a database (Afaria Database), an administration tool (Afaria Administrator), and other runtime components.

**APNS** – Apple Push Notification Service.

**artifacts** – Artifacts can be client-side or automatically generated files; for example: `.xml`, `.cs`, `.java`, `.cab` files.

**BAPI** – Business Application Programming Interface. A BAPI is a set of interfaces to object-oriented programming methods that enable a programmer to integrate third-party software into the proprietary R/3 product from SAP. For specific business tasks such as uploading transactional data, BAPIs are implemented and stored in the R/3 system as remote function call (RFC) modules.

**BLOB** – Binary Large Object. A BLOB is a collection of binary data stored as a single entity in a database management system. A BLOB may be text, images, audio, or video.

**cache** – The virtual tables in the consolidated database that store synchronization data.*See* CDB.

**cache group** – Defined in Unwired WorkSpace, MBOs are grouped and the same cache refresh policy is applied to their virtual tables (cache) in the CDB.

**cache partitions** – Partitioning the cache divides it into segments that can be refreshed individually, which gives better system performance than refreshing the entire cache. Define cache partitions in Unwired WorkSpace by defining a partition key, which is a load parameter used by the operation to load data into the cache from the enterprise information system (EIS).

**CDB** – Consolidated database. The CDB stores runtime metadata (for Unwired Platform components) and cache data (for MBOs). *See also* data tier.

**CLI** – Command line interface. CLI is the standard term for a command line tool or utility.

**client application** – *See* mobile application.

**client object API** – The client object API is described in the *Developer Reference for BlackBerry*, *Developer Reference for iOS*, and *Developer Reference for Windows Mobile*.

**cluster** – Also known as a server farm. Typically clusters are setup as either runtime server clusters or database clusters (also known as a data tier). Clustering is a method of setting up redundant Unwired Platform components on your network in order to design a highly scalable and available system architecture.

**cluster database** – A data tier component that holds information pertaining to all Unwired Platform server nodes. Other databases in the Unwired Platform data tier includes the consolidated, messaging, and monitoring databases, and the database for Afaria.

**connection** – Includes the configuration details and credentials required to connect to a database, Web service, or other EIS.

**connection pool** – A connection pool is a cache of Enterprise Information System (EIS) connections maintained by Unwired Server, so that the connections can be reused when Unwired Server receives future requests for data.

**connection profile** – In Unwired WorkSpace, a connection profile includes the configuration details and credentials required to connect to an EIS.

**context variable** – In Unwired WorkSpace, these variables are automatically created when a developer adds reference(s) to an MBO in a mobile application. One table context variable is created for each MBO attribute. These variables allow mobile application developers to specify form fields or operation parameters to use the dynamic value of a selected record of an MBO during runtime.

**data change notification (DCN)** – Data change notification (DCN) allows an Enterprise Information System (EIS) to synchronize its data with the consolidated database through a push event.

**data refresh** – A data refresh synchronizes data between the consolidated database and a back-end EIS so that data in the cache is updated. *See also* scheduled data refresh.

**data source** – In Unwired WorkSpace, a data source is the persistent-storage location for the data that a mobile business object can access.

**data tier –** The data tier includes Unwired Server data such as cache, cluster information, and monitoring. The data tier includes the consolidated database (CDB), cluster, monitoring, and messaging databases.

**deploy –** (Unwired Server) Uploading a deployment archive or deployment unit to an Unwired Server instance. Unwired Server can then make these units accessible to users via a client application that is installed on a mobile device.

There is a one-to-one mapping between an Unwired WorkSpace project and a server package. Therefore, all MBOs that you deploy from one project to the same server are deployed to the same server package.

**deployment archive –** In Unwired WorkSpace, a deployment archive is created when a developer creates a package profile and executes the **build** operation. Building creates an archive that contains both a deployment unit and a corresponding descriptor file. A deployment archive can be delivered to an administrator for deployment to a production version of Unwired Server.

**deployment descriptor –** A deployment descriptor is an XML file that describes how a deployment unit should be deployed to Unwired Server. A deployment descriptor contains role-mapping and domain-connection information. You can deliver a deployment descriptor and a deployment unit—jointly called a deployment archive—to an administrator for deployment to a production version of Unwired Server.

**deployment mode –** You can set the mode in which a mobile application project or mobile deployment package is deployed to the target Unwired Server.

**deployment profile –** A deployment profile is a named instance of predefined server connections and role mappings that allows developers to automate deployment of multiple packages from Sybase Unwired WorkSpace to Unwired Server. Role mappings and connection mappings are transferred from the deployment profile to the deployment unit and the deployment descriptor.

**deployment unit –** The Unwired WorkSpace build process generates a deployment unit. It enables a mobile application to be effectively installed and used in either a preproduction or production environment. Once generated, a deployment unit allows anyone to deploy all required objects, logical roles, personalization keys, and server connection information together, without requiring access to the whole development project. You can deliver a deployment unit and a deployment descriptor—jointly called a deployment archive—to an administrator for deployment to a production version of Unwired Server.

**development package –** A collection of MBOs that you create in Unwired WorkSpace. You can deploy the contents of a development package on an instance of Unwired Server.

**device application –** *See also* mobile application. A device application is a software application that runs on a mobile device.

**Device Application Designer –** In Unwired WorkSpace, this is the Eclipse plug-in that you can use to create and edit custom device applications for device clients graphically.

**device notification –** Replication-based synchronization (RBS) clients receive device notifications when a data change is detected for any of the MBOs in the synchronization group to which they are subscribed. Both the change detection interval of the synchronization group and the notification threshold of the subscription determine how often RBS clients receive device notifications. Administrators can use subscription templates to specify the notification threshold for a particular synchronization group.

**device user –** The user identity tied to a device.

**DML –** Data manipulation language. DML is a group of computer languages used to retrieve, insert, delete, and update data in a database.

**DMZ –** Demilitarized zone; also known as a perimeter network. The DMZ adds a layer of security to the local area network (LAN), where computers run behind a firewall. Hosts running in the DMZ cannot send requests directly to hosts running in the LAN.

**domain administrator –** A user to which the platform administrator assigns domain administration privileges for one or more domain partitions. The domain administrator has a restricted view in Sybase Control Center, and only features and domains they can manage are visible.

**domains –** Domains provide a logical partitioning of a hosting organization's environment, so that the organization achieves increased flexibility and granularity of control in multitenant environments. By default, the Unwired Platform installer creates a single domain named "default". However the platform administrator can also add more domains as required.

**EIS –** Enterprise Information System. EIS is a back-end system, such as a database.

**Enterprise Explorer –** In Unwired WorkSpace and Device Application Designer, Enterprise Explorer allows you to define data source and view their metadata (schema objects in case of database, BAPIs for SAP, and so on).

**export –** The Unwired Platform administrator can export the mobile objects, then import them to another server on the network. That server should meet the requriement needed by the exported MBO.

**hostability –** *See* multitenancy.

**IDE –** Integrated Development Environment.

**JDE –** BlackBerry Java Development Environment.

**key performance indicator (KPI) –** Used by Unwired Platform monitoring. KPIs are monitoring metrics that are made up for an object, using counters, activities, and time which jointly for the parameters that show the health of the system. KPIs can use current data or historical data.

**keystore –** The location in which encryption keys, digital certificates, and other credentials in either encrypted or unencrypted keystore file types are stored for Unwired Server runtime components. *See also* truststore.

**LDAP –** Lightweight Directory Access Protocol.

**local business object –** Defined in Unwired WorkSpace, local business objects are not bound to EIS data sources, so cannot be synchronized. Instead, they are objects that are used as local data store on device.

**logical role –** Logical roles are defined in mobile business objects, and mapped to physical roles when the deployment unit that contain the mobile business objects are deployed to Unwired Server.

**matching rules –** A rule that triggers a mobile workflow application. Matching rules are used by the mobile workflow email listener to identify e-mails that match the rules specified by the administrator. When emails match the rule, Unwired Server sends the e-mail as a mobile workflow to the device that matches the rule. A matching rule is configured by the administrator in Sybase Control Center.

**MBO –** Mobile business object. The fundamental unit of data exchange in Sybase Unwired Platform. An MBO roughly corresponds to a data set from a back-end data source. The data can come from a database query, a Web service operation (including Remedy), or SAP. An MBO contains both concrete implementation-level details and abstract interface-level details. At the implementation-level, an MBO contains read-only result fields that contain metadata about the data in the implementation, and parameters that are passed to the back-end data source. At the interface-level, an MBO contains attributes that map to result fields, which correspond to client properties. An MBO may have operations, which can also contain parameters that map to arguments, and which determines how the client passes information to the enterprise information system (EIS).

You can define relationships between MBOs, and link attributes and parameters in one MBO to attributes and parameters in another MBO.

**MBO attribute –** An MBO attribute is a field that can hold data. You can map an MBO attribute to a result field in a back-end data source; for example, a result field in a database table.

**MBO binding –** An MBO binding links MBO attributes and operations to a physical data source through a connection profile.

**MBO operation –** An MBO operation can be invoked from a client application to perform a task; for example, create, delete, or update data in the EIS.

**MBO relationship –** MBO relationships are analogous to links created by foreign keys in a relational database. For example, the account MBO has a field called *owner_ID* that maps to the *ID* field in the owner MBO.

Define MBO relationships to facilitate:
- Data synchronization
- EIS data-refresh policy

**messaging based synchronization (MBS) –** A synchronization method where data is delivered asynchronously using a secure, reliable messaging protocol. MBS provides fine-

grained synchronization (synchronization is provided at the data level—each process communicates only with the process it depends on), and it is therefore assumed that the device is always connected and available. *See also* replication based synchronization.

**messaging database** – The messaging database allows in-flight messages to be stored until they can be delivered. This database is used in a messaging based synchronization environment. The messaging database is part of the Unwired Platform data tier, along with the consolidated, cluster, and monitoring databases.

**mobile application** – A Sybase Unwired Platform mobile application is an end-to-end application, which includes the MBO definition (back-end data connection, attributes, operations, and relationships), the generated server-side code, and the client-side application code.

**Mobile Application Diagram** – The Mobile Application Diagram is the graphical interface to create and edit MBOs. By dragging and dropping a data source onto the Mobile Application Diagram, you can create a mobile business object and generate its attribute mappings automatically.

**Mobile Application Project** – A collection of MBOs and client-side, design-time artifacts that make up a mobile application.

**mobile workflow packages** – Mobile workflow packages use the message-based synchronization model. The mobile workflow packages are deployed to Unwired Server, and can be deployed to mobile devices, via the Unwired Platform administrative perspective in Sybase Control Center.

**monitoring** – Monitoring is an Unwired Platform feature available in Sybase Control Center that allows administrators to identify key areas of weakness or periods of high activity in the particular area they are monitoring. It can be used for system diagnostic or for troubleshooting. Monitored operations include replication-based synchronization, messaging-based synchronization, messaging queue, data change notification, device notification, package, user, and cache activity.

**monitoring database** – A database that exclusively stores data related to replication and messaging synchronization, queues status, users, data change notifications, and device notifications activities. By default, the monitoring database runs in the same data tier as the consolidated database, messaging database and cluster database.

**monitoring profiles** – Monitoring profiles specify a monitoring schedule for a particular group of packages. These profiles let administrators collect granular data on which to base domain maintenance and configuration decisions.

**multitenancy** – The ability to host multiple tenants in one Unwired Cluster. Also known as hostability. *See also* domains.

**node** – A host or server computer upon which one or more runtime components have been installed.

**object query –** Defined in Unwired WorkSpace for an MBO and used to filter data that is downloaded to the device.

**openDS –** The default LDAP server that is installed in Developer Edition and is suitable for authentication and authorization in a development environment.

**operation –** *See* MBO operation.

**package –** A package is a named container for one or more MBOs. On Unwired Server a package contains MBOs that have been deployed to this instance of the server.

**palette –** In Unwired WorkSpace, the palette is the graphical interface view from which you can add MBOs, local business objects, structures, relationships, attributes, and operations to the Mobile Application Diagram.

**parameter –** A parameter is a value that is passed to an operation/method. The operation uses the value to determine the output. When you create an MBO, you can map MBO parameters to data-source arguments. For example, if a data source looks up population based on a state abbreviation, the MBO gets the state from the user, then passes it (as a parameter) to the data source to retrieve the information. Parameters can be:

- Synchronization parameters – synchronize a device application based on the value of the parameter.
- Load parameters – perform a data refresh based on the value of the parameter.
- Operation parameters – MBO operations contain parameters that map to data source arguments. Operation parameters determine how the client passes information to the enterprise information system (EIS).

**personalization key –** A personalization key allows a mobile device user to specify attribute values that are used as parameters for selecting data from a data source. Personalization keys are also used as operation parameters. Personalization keys are set at the package level. There are three type of personalization keys: Session, client, server.

They are most useful when they are used in multiple places within a mobile application, or in multiple mobile applications on the same server. Personalization keys may include attributes such as name, address, zip code, currency, location, customer list, and so forth.

**physical role –** A security provider group or role that is used to control access to Unwired Server resources.

**Problems view –** In Eclipse, the Problems view displays errors or warnings for the Mobile Application Project.

**provisioning –** The process of setting up a mobile device with required runtimes and device applications. Depending on the synchronization model used and depending on whether or not the device is also an Afaria client, the files and data required to provision the device varies.

**pull synchronization –** Pull synchronization is initiated by a remote client to synchronize the local database with the CDB. On Windows Mobile, pull synchronization is supported only in RBS applications.

**push synchronization** – Push is the server-initiated process of downloading data from Unwired Server to a remote client, at defined intervals, or based upon the occurrence of an event.

**queue** – In-flight messages for a messaging application are saved in a queue. A queue is a list of pending activities. The server then sends messages to specific destinations in the order that they appear in the queue. The depth of the queue indicates how many messages are waiting to be delivered.

**relationship** – *See* MBO relationship.

**relay server** – *See also* Sybase Hosted Relay Service.

**replication based synchronization (RBS)** – A synchronization method where data is delivered synchronously using an upload/download pattern. For push-enabled clients, RBS uses a "poke-pull" synchronization model, where a notification is pushed to the device (poke), and the device fetches the content (pull), and is assumed that the device is not always connected to the network and can operate in a disconnected mode and still be productive. For clients that are not push-enabled, the default synchronization model is pull.*See also* messaging based synchronization.

**REST web services** – Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web.

**RFC** – Remote Function Call. You can use the RFC interface to write applications that communicate with SAP R/3 applications and databases. An RFC is a standalone function. Developers use SAP tools to write the Advanced Business Application Programming (ABAP) code that implements the logic of a function, and then mark it as "remotely callable," which turns an ABAP function into an RFC.

**role** – Roles control access to Sybase Unwired Platform resources. *See also* logical role and physical role.

**role mapping** – Maps a physical (server role) to a logical (Unwired Platform role). Role mappings can be defined by developers, when they deploy an MBO package to a development Unwired Server, or by platform or domain administrators when they assign a security configuration to a domain or deploy a package to a production Unwired Server (and thereby override the domain-wide settings in the security configuration).

**RSOE** – Relay Server Outbound Enabler. An RSOE is an application that manages communication between a back-end server—Unwired Server or Afaria—and a relay server.

**runtime server** – An instance of Unwired Server that is running. Typically, a reference to the runtime server implies a connection to it.

**SAP** – SAP is one of the EIS types that Unwired Platform supports.

**SCC** – Sybase Control Center. A Web-based interface that allows you to administer your installed Sybase products.

**scheduled data refresh** – Data is updated in the consolidated database from a back-end EIS, based on a scheduled data refresh. Typically, data is retrieved from an EIS (for example, SAP) when a device user synchronizes. However, if an administrator wants the data to be preloaded for a mobile business object, a data refresh can be scheduled so that data is saved locally in a cache. By preloading data with a scheduled refresh, the data is available in the information server when a user synchronizes data from a device. Scheduled data refresh requires that an administrator define a cache group as "scheduled" (as opposed to "on-demand").

**security configuration** – Part of the application user and administration user security. A security configuration determines the scope of user identity, authentication and authorization checks, and can be assigned to one or more domains by the platform administrator in Sybase Control Center. A security configuration contains:

- A set of configured security providers (for example LDAP) to which authentication, authorization, attribution is delegated.
- Role mappings (which can be specified at the domain or package level)

**security provider** – A security provider and it's repository holds information about the users, security roles, security policies, and credentials used by some to provide security services to Unwired Platform. A security provider is part of a security configuration.

**security profile** – Part of the Unwired Server runtime component security. A security profile includes encryption metadata to capture certificate alias and the type of authentication used by server components. By using a security profile, the administrator creates a secured port over which components communicate.

**server connection** – The connection between Unwired WorkSpace and a back-end EIS is called a server connection.

**server farm** – *See also* cluster. Is the relay server designation for a cluster.

**server-initiated synchronization** – *See* push synchronization.

**SOAP** – Simple Object Access Protocol. SOAP is an XML-based protocol that enables applications to exchange information over HTTP. SOAP is used when Unwired Server communicates with a Web service.

**solution** – In Visual Studio, a solution is the high-level local workspace that contains the projects users create.

**Solution Explorer** – In Visual Studio, the Solution Explorer pane displays the active projects in a tree view.

**statistics** – In Unwired Platform, the information collected by the monitoring database to determine if your system is running as efficiently as possible. Statistics can be current or historical. Current or historical data can be used to determine system availability or performance. Performance statistics are known as key performance indicators (KPI).

**Start Page** – In Visual Studio, the Start Page is the first page that displays when you launch the application.

**structured data** – Structured data can be displayed in a table with columns and labels.

**structure object** – Defined in Unwired WorkSpace, structures hold complex datatypes, for example, a table input to a SAP operation.

**subscription** – A subscription defines how data is transferred between a user's mobile device and Unwired Server. Subscriptions are used to notify a device user of data changes, then these updates are pushed to the user's mobile device.

**Sybase Control Center** – Sybase Control Center is the Flash-based Web application that includes a management framework for multiple Sybase server products, including Unwired Platform. Using the Unwired Platform administration perspective in Sybase Control Center, you can register clusters to manage Unwired Server, manage domains security configurations, users, devices, connections and monitor the environment. You can also deploy MBO packages and manage deployed MBO packages in order to design the synchronization behavior for those packages. Only use the features and documentation for Unwired Platform. Default features and documentation in Sybase Control Center do not always apply to the Unwired Platform use case.

**Sybase Hosted Relay Service** – The Sybase Hosted Relay Service is a Web-hosted relay server that enables you to test your Unwired Platform development system.

**Sybase Messaging Service** – The synchronization service that facilitates communication with device client applications.

**Sybase Unified Agent** – Provides runtime services to manage, monitor, and control distributed Sybase resources. The agent must be running for Sybase Control Center to run.

**Sybase Unwired Platform** – Sybase Unwired Platform is a development and administrative platform that enables you to mobilize your enterprise. With Unwired Platform, you can develop mobile business objects in the Unwired WorkSpace development environment, connect to structured and unstructured data sources, develop mobile applications, deploy mobile business objects and applications to Unwired Server, which manages messaging and data services between your data sources and your mobile devices.

**Sybase Unwired WorkSpace** – Sybase Unwired Platform includes Unwired WorkSpace, which is a development tool for creating mobile business objects and mobile applications.

**synchronization group** – Defined in Unwired WorkSpace, a synchronization group is a collection of MBOs that are synchronized at the same time.

**synchronization parameter** – A synchronization parameter is an MBO attribute used to filter and synchronize data between a mobile device and Unwired Server.

**synchronization phase** – For replication based synchronization packages, the phase can be an upload event (from device to the consolidated database) or download event (from the consolidated database to the device).

**synchronize –** *See also* data refresh. Synchronization is the process by which data consistency and population is achieved between remote disconnected clients and Unwired Server.

**truststore –** The location in which certificate authority (CA) signing certificates are stored. *See also* keystore.

**undeploy –** Running **undeploy** removes a domain package from an Unwired Server.

**Unwired Server –** The application server included with the Sybase Unwired Platform product that manages mobile applications, back-end EIS synchronization, communication, security, transactions, and scheduling.

**user –** Sybase Control Center displays the mobile-device users who are registered with the server.

**Visual SQL –** A graphical user interface tool that you can use to build SQL queries.

**Visual Studio –** Microsoft Visual Studio is an integrated development environment product that you can use to develop device applications from generated Unwired WorkSpace code.

**Welcome page –** In Eclipse, the first set of pages that display when you launch the application.

**workspace –** In Eclipse, a workspace is the directory on your local machine where Eclipse stores the projects that you create.

**WorkSpace Navigator –** In Eclipse, the tree view that displays your mobile application projects.

**WSDL file –** Web Service Definition Language file. The file that describes the Web service interface that allows clients to communicate with the Web service. When you create a Web service connection for a mobile business object, you enter the location of a WSDL file in the URL.

# Index