



**Sybase Unwired WorkSpace - Mobile  
Workflow Package Development**

---

**Sybase Unwired Platform 2.1**

**ESD #3**

DOCUMENT ID: DC01284-01-0213-01

LAST REVISED: March 2012

Copyright © 2012 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase trademarks can be viewed at the Sybase trademarks page at <http://www.sybase.com/detail?id=1011207>. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.

Java and all Java-based marks are trademarks or registered trademarks of Oracle and/or its affiliates in the U.S. and other countries.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

# Contents

<b>Develop a Mobile Workflow Package .....</b>	<b>1</b>
<b>The Mobile Workflow Forms Editor .....</b>	<b>3</b>
Setting Mobile Workflow Forms Editor Preferences .....	3
Mobile Workflow Forms Editor Preferences .....	4
Creating a New Mobile Workflow Form .....	6
Creating Screens Automatically by Dragging and Dropping a Data Source .....	7
Adding Images to the Project .....	8
Mobile Workflow Forms Editor Context Menus .....	9
Starting Points .....	10
Adding a Starting Point Manually .....	11
Creating a Server-Initiated Starting Point .....	12
Implementing a Conditional Start Screen .....	18
Credentials .....	20
Creating Keys .....	24
Flow Design Page Properties .....	25
Flow Design Palette Options .....	29
Adding GoTo Connections to Screens .....	32
Screen Design Page Properties .....	34
Screen Design Palette Options .....	36
Adding Palette Objects to the Screen Design .....	39
Mobile Workflow Controls .....	39
<b>Configuring Control Properties .....</b>	<b>43</b>
Checkbox Properties .....	43
Editbox Properties .....	44
Choice Properties .....	47
Adding Choice Items to the Choice Control .....	49
Editing Choice Items .....	50
Deleting Choice Items .....	50
Creating Dynamic Contents for the Choice Control .....	50

- Configuring a Choice Control to Use Data from an MBO .....50
- Slider Properties .....52
- Signature Properties .....53
- HtmlView Properties .....55
- Listview Properties .....55
  - Adding or Editing a Listview Field .....57
  - Generated Keys .....58
- AttachmentViewer Properties .....59
  - Adding and Configuring the AttachmentViewer ..61
- Button Properties .....64
- Label Properties .....65
- Link Properties .....65
- Image Properties .....66
- Configuring a Menu Item or Custom Action .....69**
  - Menu Item and Custom Action Properties .....69
  - Creating a Save, Open, Close, or Cancel Menu Item ...71
  - Creating a Submit Workflow Menu Item .....72
  - Creating an Online Request Menu Item .....74
  - Creating a Key Collection Menu Item .....77
  - Configuring Parameter Mappings .....78
  - Defining a Custom Menu Item or Custom Action .....79
- Localization and Internationalization .....81**
  - Localization Limitations .....81
  - Localizing a Mobile Workflow Package .....82
    - Changing the Encoding Type .....83
    - Creating and Validating a New Locale Properties File .....84
    - Editing the Locale Properties File .....87
    - Deploying the Mobile Application Project .....87
    - Configuring the Device for Localization .....88
    - Removing a Locale .....90
    - Updating the Current Locale .....91
  - Mobile Workflow Package Internationalization .....91
  - Internationalization on the Device .....93

<b>Sending a Test Notification .....</b>	<b>95</b>
<b>Generating the Files for a Mobile Workflow Package .....</b>	<b>97</b>
Deployment Modes .....	99
Generated Mobile Workflow Files .....	99
Generated HTML Files .....	101
Deploying the Mobile Workflow Package to Devices ..	106
Activating the Workflow .....	106
<b>Adding Custom Code .....</b>	<b>107</b>
<b>Troubleshoot .....</b>	<b>109</b>
<b>API Documentation .....</b>	<b>111</b>
<b>Glossary: Sybase Unwired Platform .....</b>	<b>113</b>
<b>Index .....</b>	<b>125</b>

# Contents

# Develop a Mobile Workflow Package

Sybase® Mobile Workflow Forms Editor lets you create or modify Mobile Workflow packages on different device platforms without native device development.

Mobile Workflows support the occasionally connected user and addresses the replication and synchronization issues those users present for the back-end system.

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 container installed and that are registered with Unwired Server through 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 the create, update, delete, and other operations, as well as object queries of a mobile business object.

You can generate a Mobile Workflow application that works on these platforms:

- Apple iOS
- Android
- BlackBerry
- Windows Mobile Professional

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.

## Develop a Mobile Workflow Package



# The Mobile Workflow Forms Editor

The Mobile Workflow Forms Editor allows you to create the forms that can be assigned to the devices that have the Hybrid Web Container installed and that are registered in Sybase Control Center.

The Mobile Workflow Forms Editor contains these pages:

- **Introduction**

By default, the Introduction page opens when you create a new mobile workflow or double-click an existing mobile workflow. To hide this page, click **Window > Preferences**, navigate to **Sybase, Inc > Mobile Development > Mobile Workflow Forms Editor**, and unselect the option, **Show the Introduction page when opening the designer**.

- **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 application. The Palette view shows the menu items and controls you can use to design a new screen. Drag controls to design the user interface.

## Setting Mobile Workflow Forms Editor Preferences

---

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

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

### See also

- *Creating a New Mobile Workflow Form* on page 6
- *Mobile Workflow Forms Editor Context Menus* on page 9
- *Starting Points* on page 10
- *Flow Design Page Properties* on page 25
- *Flow Design Palette Options* on page 29
- *Screen Design Page Properties* on page 34
- *Screen Design Palette Options* on page 36

## Mobile Workflow Forms Editor Preferences

Use the Preferences dialog to set the default behavior of the Mobile Workflow Forms editor.

Preferences affect all Mobile Workflow Forms editor instances; alternatively, you can change the style and behavior of a particular object in a form that you are editing, by modifying its properties in the Properties view.

### *Mobile Workflow Forms Editor Preferences*

You can set the following preferences in the Mobile Workflow Forms Editor page:

- **Optimize for performance** provides a more basic look and feel. Raw HTML controls are used to the extent possible (for example, there is no signature or slider control in HTML), with little to no CSS stylization or JavaScript enhancements to customize the look and feel. This option is always used for Windows Mobile Professional platforms.
- **Optimize for appearance** provides a more sophisticated, or native, look and feel, but may affect performance. This option is not available on Windows Mobile platforms.
- **Show the Introduction page when opening the designer** controls whether the Introduction page is visible or hidden. If this option is enabled, the Forms Editor opens to the Introduction page, which provides basic information and help. .

### *Screen Design Page Preferences*

This page and its subpages control global settings for the Flow Design and Screen Design pages of the Mobile Workflow Forms Editor. The page includes these options:

- Show connector handles
- Show pop-up bars
- Enable animated layout
- Enable animated zoom
- Enable anti-aliasing
- Show status line

### *Appearance Page Preferences*

Modify the colors and fonts for various Mobile Workflow objects by clicking the color controls, or by selecting **Change**, and setting the desired color and font.

### *Connections Page Preferences*

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.

Select the default line style for new connections:

- Oblique (default) – the 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.

### *Printing Page Preferences*

Set the general printing preferences:

Section	Description
Orientation	Either portrait or landscape.
Units	Either inches or millimeters.
Size	Choose one of the available print size. Optionally, override the default width and height values.
Margins	Enter a value, in inches, for the margins.

### *Rulers and Grid Page Preferences*

Set the preferences for the ruler and grids:

Section	Description
Rulers options	To show ruling lines, select <b>Show rulers for new diagram</b> , then select ruler units.
Grid options	To modify grid settings, select any of these options:  <p style="text-align: center;"><b>Show grid for new diagrams</b>  <b>Snap to grid for new diagrams</b>  <b>Snap to shapes for new diagrams</b></p> Then, enter the grid spacing.

## 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 <b>Browse</b> 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 WorkSpace Navigator. Conversely, when you select the resource in the WorkSpace Navigator, the editor is selected.  Click <b>Variables</b> to define a new path variable. Path variables specify locations on the file system.

3. Click **Next**.
4. In the Starting Points page, select the ways the mobile workflow starts:
  - **Can be started, on demand, from the client** – creates a Client-initiated starting point. This allows the user to start the mobile workflow on the device and submit data to the server.

- **Responds to server-driven notifications** – creates a Server-initiated starting point. 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** – creates an Activate starting point so that the first time the application is launched, the screen pointed to by the Activation screen is shown.
  - **Credentials (authentication) may be requested dynamically from the client application** – creates a Credential Request starting point with pre-defined username and password keys.
5. (Optional) If you selected the **Responds to server-driven notifications** option in the previous step, follow the procedure in *Creating a Server-initiated Starting Point*.
  6. Click **Finish**.
  7. In the Mobile Workflow Forms Editor, click **Flow Design** to start designing the mobile workflow.

### See also

- *Setting Mobile Workflow Forms Editor Preferences* on page 3
- *Mobile Workflow Forms Editor Context Menus* on page 9
- *Starting Points* on page 10
- *Flow Design Page Properties* on page 25
- *Flow Design Palette Options* on page 29
- *Screen Design Page Properties* on page 34
- *Screen Design Palette Options* on page 36
- *Creating a Server-Initiated Starting Point* on page 12
- *Server-initiated Starting Point Properties* on page 14

## 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.

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.

## The Mobile Workflow Forms Editor

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 workflows.

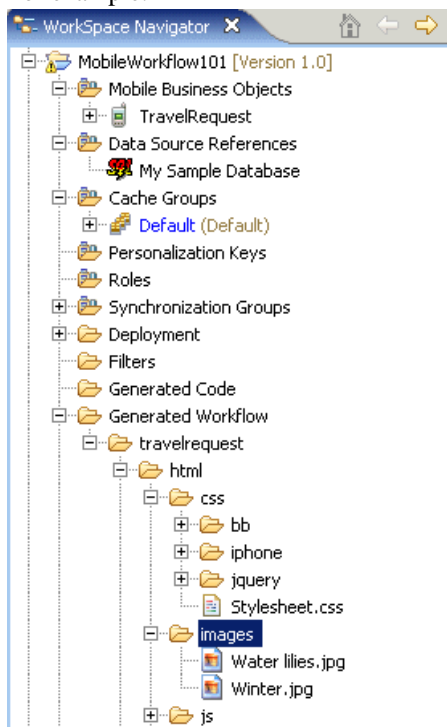
---

## Adding Images to the Project

To use images in the mobile workflow application, you must first add them to the project.

1. In WorkSpace Navigator, navigate to **<MBO\_project> > Generated Workflow > <Mobile\_Workflow> > html** and create a subfolder for images.

For example:



2. Copy the images you want to use into the folder you created in Step 1.
3. Save your project.

### See also

- *Screen Design Page Properties* on page 34

## 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 edit or delete existing mobile business object references.  <b>Note:</b> This menu item appears only on the Flow Design page.
Generate Mobile Workflow Package	Generate the mobile workflow form code.
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.  <b>Note:</b> 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: <ul style="list-style-type: none"> <li>• All – selects all objects on the page.</li> <li>• All shapes – selects all shapes on the page.</li> <li>• All connections – selects all connections on the page.</li> </ul>
Arrange all	Arranges the shapes on the page.

Option	Description
View	View: <ul style="list-style-type: none"> <li>• Grid</li> <li>• Rulers</li> <li>• Page breaks</li> <li>• Recalculate page breaks</li> <li>• Snap to grid</li> </ul>
Zoom	Select from the zoom options.
Show properties view	Shows the Properties view for the selected object.

**See also**

- *Setting Mobile Workflow Forms Editor Preferences* on page 3
- *Creating a New Mobile Workflow Form* on page 6
- *Starting Points* on page 10
- *Flow Design Page Properties* on page 25
- *Flow Design Palette Options* on page 29
- *Screen Design Page Properties* on page 34
- *Screen Design Palette Options* on page 36

## Starting Points

---

Starting points specify which screen the 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 mobile workflow. It does not activate on the second and subsequent times the 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.

---

**Note:** A mobile workflow application with an Activate starting point is considered to be successfully activated only if it is closed with a Submit Workflow menu item.

---



- Credential request – this starting point is activated when the mobile 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. The developer can add conditions, based on the notification message, to determine which application screen opens first.

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

- *Setting Mobile Workflow Forms Editor Preferences* on page 3
- *Creating a New Mobile Workflow Form* on page 6
- *Mobile Workflow Forms Editor Context Menus* on page 9
- *Flow Design Page Properties* on page 25
- *Flow Design Palette Options* on page 29
- *Screen Design Page Properties* on page 34
- *Screen Design Palette Options* on page 36

## 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 can 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. Depending on the type of starting point you select:
  - If you are adding an Activate, Credential Request, or Client-initiated starting points, go to the next step.
  - If you are adding a Server-initiated starting point, the Notification Processing wizard opens. In the wizard, complete the steps to configure the object query and notification message to be used, and click **Finish** to create the starting point.
3. 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

## The Mobile Workflow Forms Editor

- Selecting an empty screen from the Flow Design palette, then clicking on the Flow Design page.
4. Add a connection between the starting point and the screen.
  5. 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 Form package 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 <b>Browse</b> 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 WorkSpace Navigator. Conversely, when you select the resource in the WorkSpace Navigator, the editor is selected.  Click <b>Variables</b> to define a new path variable. Path variables specify locations on the file system.

3. Click **Next**.
4. In the Starting Points page, select **Responds to server-driven notifications** and click **Next**.
5. In the Select a Mobile Business Object and Object Query page, click **Search**.
6. In the Search for Mobile Business Object page: click **Search**.

- a) Click **Search**.
  - b) Accept the current project or choose a different one.
  - c) Click **Search**.
  - d) Select a Mobile Application Project from the list and click **OK**.
7. (Optional) In the Select a Mobile Business Object and Object Query page, select an object query and click **Next**.

---

**Note:** Object queries that return result sets are not supported.

---

8. In the Specify Sample Notification page, enter contents in the fields that you want to appear in the sample notification message, then click **Next**.  
For example enter `Order (2001) created` as the Subject.
9. In the Specify Matching Rules page:
- a) Select the text that will serve as the matching rule.
  - b) Right-click and select **Select as Matching Rule** to create the matching rule.
  - c) Click **Next**.
10. In the Identify Parameter Values page, select an extraction rule and specify its properties.

---

**Note:** The supported syntax for the regular expression is that used by the C# `Match()` methods 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	Specify a format that 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 notification field contents	Enter the value of the field previously specified on the Specify Sample Notification page of the Notification Processing wizard.

Property	Description
Value extracted from sample notification field contents	Specify the value the start tag and end tag matching rules, given the sample field value.

**11. Click **Finish**** to create default screens and starting points.

When the notification is sent to the client, the sample value is replaced with the specified key, and the client's form is populated with values returned by the object query

**See also**

- *Creating a New Mobile Workflow Form* on page 6
- *Server-initiated Starting Point Properties* on page 14

**Server-initiated Starting Point Properties**

View and configure the server-initiated starting point properties.

*General*

Select the **Server-initiated** starting point in the Flow Design page to view and configure its properties.

Property	Description
Mobile business object	The mobile business object with which the server-initiated starting point is associated.
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.

Property	Description
Generate old value keys	By default, this is selected, which means when data is sent, there are two copies sent—one with the key names you select at design time, and one with the same names prefixed with "_old." The keys you selected at design time may get updated by the application, but the old values will not unless this option is selected. Those "_old" value keys may be critical when executing an update or parent update operation at a later time. If you unselect this option, old value keys are not generated.
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 <b>Edit</b> to change the transformation rule.
Error screen	The screen to which the user navigates in case an asynchronous operation invocation fails.
Subject	If an error occurs, a notification is sent to the device with the appropriate error log keys and specified Subject. The value you specify here appears in the Subject field of the notification.
From	When an error occurs and the notification appears in the inbox, what you specify here will appear in the From field of the notification.

### Keys

Keys associated with the controls on the screen are shown. You can sort the table on any property by clicking its column heading.

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. Click <b>New</b> to create a new key. Select any key and click <b>Edit</b> to edit the information for an existing key. Select a key and click <b>Remove</b> to remove an existing key.

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.

For information about these parameters and how they map to MBO operations and queries, please see *WorkflowClient.xml File in Sybase Unwired Workspace - Mobile Workflow Package Development*.

### Personalization Key Mappings

If you have a mobile business object with load parameters mapped to transient personalization keys, you can specify values for those personalization keys when you invoke an operation or object query from the Mobile Workflow package. This can, for example, be useful for loading only specific data into the consolidated database from large data sources.

### Start Screen(s)

This page contains a table of condition names with the matching Start screen. If all of the conditions are evaluated as false (or if they are absent), the default navigation is executed.

Property	Description
Default screen	Select the default start screen. This is the start screen that is used if all of the conditions are false or absent.
Condition name	Click <b>Add</b> to add a condition. If this condition is determined to be true, the screen that matches the condition is used as the start screen.
Conditional success screen name	Select the screen that will be associated with the condition.

### See also

- *Creating a New Mobile Workflow Form* on page 6
- *Creating a Server-Initiated Starting Point* on page 12

### **Notification Extraction Rules**

Notification extraction 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 notification 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 or object query.

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.

Some e-mail clients allow you to specify the To: and From: fields without quotes (for example, John Doe <john.doe@sybase.com>), the actual value received by the Unwired server may have quotes in it (for example, "John Doe" <john.doe@sybase.com>), which will need to be accounted for in any extraction rules that act on those fields.

### **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:

<b>Property</b>	<b>Description</b>
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".

Property	Description
Format	Format uses the C# ParseExact syntax, and applies to DateTime only. It 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.
Key	The key that the value from the notification is extracted to.

---

**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**.

## **Implementing a Conditional Start Screen**

Add conditions that determine which start screen the user sees based on the conditions.

Like the conditional success navigation feature, there is a table of condition names with the matching Start screen. If all of the conditions are evaluated as false (or if they are absent), the default navigation is executed.

1. In the Flow Design page, select the server-initiated starting point to see the Properties.



2. In the Properties view, click Start Screen(s).
3. Click **Add** to add a condition.
4. In the dialog, enter the condition name, select the target screen with which to associate the condition, and click **OK**.

This means that if the defined condition is found to be true, the screen you choose here will be the start screen. Condition names can include:

- Letters A-Z and a-z
- Numbers 0-9
- Embedded spaces (beginning and ending spaces are trimmed off)
- Special characters in the set \$.\_-+

In the Flow Design page, you can see the flow line for the conditional start is a shade of gray to differentiate it from the default GoTo line.

Condition Name	Conditional Success Screen Name
Wilma_first_ss1	Screen Start One
Fred_second_screen	Screen Start Two

5. Add you custom code to the Custom.js file. For example:

```
function customConditionalNavigation( currentScreenKey,
actionName,
defaultNextScreen, conditionName,
workflowMessage ) {
    if((currentScreenKey === SERVERINITIATEDFLAG) && (actionName
=== '')) {
        // conditional start screen uses this magic screen key and
the empty action name.
        if( conditionName === 'Wilma_first_ss1') {
            // custom logic

```

```
        return true;
    }
    else if(conditionName === 'Fred_second_screen'){
        // custom logic
        // return true or false
        return false;
    }
}
// default case is to NOT change the flow
return false;
}
```

### 6. Regenerate the Mobile Workflow package.

When you regenerate the Mobile Workflow package, the `Workflow.js` file is regenerated. The conditional start screen method is shown in the `Workflow.js` file similar to this:

```
function customNavigationEntry() {
    this.condition;
    this.screen;
}
function customNavigationEntry( a_condition, a_screen ) {
    this.condition = a_condition;
    this.screen = a_screen;
}

/**
 * For the specific pair - screen named 'currentScreenKey' and the
 * action 'actionName', return
 * the list of custom navigation condition-names and their
 * destination screens.
 */
function getCustomNavigations( currentScreenKey, actionName ) {
    var customNavigations = new Array();
    if((currentScreenKey === SERVERINITIATEDFLAG) && (actionName
=== '')) {
        customNavigations[0] = new
customNavigationEntry( 'Wilma_first_ssl',
'Screen_Start_One' );
        customNavigations[1] = new
customNavigationEntry( 'Fred_second_screen',
'Screen_Start_Two' );
        return customNavigations;
    }
    return customNavigations;
}
```

## Credentials

You can use either dynamic or static credentials in a mobile workflow form.

See the *Developer Guide for Mobile Workflow Packages* and *System Administration* for more detailed information about implementing security and certificates.

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.
  - Use certificate-based credentials – when you select this option, you can use a certificate to generate authentication credentials.
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 Static Authentication Using a Certificate

Set up static authentication credentials generated from a certificate.

1. In the Properties view, click **Authentication**.
2. Select **Use static credentials** and Use certificate-based credentials.
3. Click **Generate from Certificate** to select a certificate file from which to generate authentication.
4. In the Certificate Picker, click **Browse** to locate the certificate to use.
5. Enter a password and select an alias, then click **OK**.

The information from the certificate is shown in the Properties view.

- Issuer – the issuer of the certificate
- Subject – the value of the subject field in the metadata of the certificate as defined in the X.509 standard
- Valid from – the date the certificate is valid from
- Valid until – the date after which the certificate expires

6. 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.

Field	Description
Link to file in the file system	<p>Click <b>Browse</b> 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 WorkSpace Navigator. Conversely, when you select the resource in the WorkSpace Navigator, the editor is selected.</p> <p>Click <b>Variables</b> to define a new path variable. Path variables specify locations on the file system.</p>

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 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 it 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. (Optional) To use certificate-based authentication instead of the user name and password:
  - a) Add a **MenuItem** to the Menu box.
  - b) Select the MenuItem to see the Properties.
  - c) In the Properties view, from Type, choose **Select Certificate**.  
When the user selects the menu item on the device, a dialog is opened that allows the user to select a certificate to use for credentials.
9. 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 to bind 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. The key name must be unique across the scope of the workflow application.
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: <ul style="list-style-type: none"> <li>• Name</li> <li>• Convert to UTC</li> </ul>
MBO relationship	Bind the key to an MBO relationship.
MBO object query results	Bind the key to MBO object query results.
Hard coded value	When the message is sent by the server to the client, rather than getting the value of the key from a source like an MBO attribute, it uses the fixed value you input. <hr/> <b>Note:</b> The return of hard coded key values is supported from server-initiated starting points, only, not online requests.
User defined	Bind the key to user-defined data.
Extraction rule	This is determined by the extraction rules you set up.

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 – NaN (not a number) values are not supported for keys of type 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.

**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**.

**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.

Property	Description
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.
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.
Shared storage key	Specify a storage key to share storage between Mobile Workflow packages. This storage key is passed to the manifest.xml file used by the container.



*Rulers and grid*

Property	Description
Display	<ul style="list-style-type: none"> <li>• Show ruler</li> <li>• Show grid</li> <li>• Grid in front</li> </ul>
Measurement	<ul style="list-style-type: none"> <li>• Ruler units</li> <li>• Grid spacing</li> </ul>
Grid line	<ul style="list-style-type: none"> <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.

*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 notifications 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.
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.
Use certificate-based credentials	<p>Click <b>Generate from Certificate</b> to select a certificate file from which to generate authentication. Once you choose the certificate to use, these fields are filled in automatically.</p> <ul style="list-style-type: none"> <li>• Issuer – the issuer of the certificate</li> <li>• Subject – the value of the subject field in the metadata of the certificate as defined in the X.509 standard</li> <li>• Valid from – the date the certificate is valid from</li> <li>• Valid to – the date the certificate is valid to</li> </ul>

Property	Description
Unwired Server domain	<p>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, @&lt;domain&gt; is added to the user name, thus allowing the user access to that domain.</p> <hr/> <p><b>Note:</b> When you create a new security configuration that includes the SAPSSOTokenLoginModule, and deploy it to a new domain, if the mobile workflow application uses the MBOs associated with the new security configuration, you must specify an Unwired Server domain that corresponds to the domain using the security configuration. See the Sybase Unwired Platform <i>Security</i> guide for more information about security configurations.</p>

### See also

- *Setting Mobile Workflow Forms Editor Preferences* on page 3
- *Creating a New Mobile Workflow Form* on page 6
- *Mobile Workflow Forms Editor Context Menus* on page 9
- *Starting Points* on page 10
- *Flow Design Palette Options* on page 29
- *Screen Design Page Properties* on page 34
- *Screen Design Palette Options* on page 36

## Flow Design Palette Options

---

The Flow Design Palette lists connection types, screens, and starting points that can be used in a mobile workflow application.

### *Connections*

From the Flow Design Palette, you can add only a GoTo connection directly to the canvas. All the other connections are listed only for reference, and must be defined in the properties view.

Connection	Description
GoTo	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.
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.
Conditional Operation Success	This connection implies a code generation callout to the customConditionalNavigation function, which returns a boolean. When a user navigates through the application, each piece of custom logic is evaluated individually, in order. Upon finding the first true condition, the matching screen is shown on the user's device.
Async Request Error	Indicates that, if an asynchronous operation fails, a new notification should be sent to the device with the appropriate error log keys, along with user-specified email field values (such as Subject and From).

*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: <ol style="list-style-type: none"> <li>1. User-defined user name and password keys</li> <li>2. Credential cache</li> <li>3. Specified generic credential pair</li> </ol>
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**

- *Setting Mobile Workflow Forms Editor Preferences* on page 3
- *Creating a New Mobile Workflow Form* on page 6
- *Mobile Workflow Forms Editor Context Menus* on page 9
- *Starting Points* on page 10
- *Flow Design Page Properties* on page 25
- *Screen Design Page Properties* on page 34
- *Screen Design Palette Options* on page 36

## **Adding GoTo 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. In the Flow Design Palette, click **GoTo**.
2. In the Flow Design canvas, click the source screen and drag to connect it to the target screen.

The GoTo connection line appears between the two screens.

---

**Note:** You cannot use the Flow Design Palette to add any of the other connection types. For example, to create an Async Request Error connection, set the Error Screen property for a submit workflow menu item or a server-initiated starting point.

---

### **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	<p>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.</p> <ul style="list-style-type: none"> <li>• Styles: <ul style="list-style-type: none"> <li>• 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.</li> <li>• Rectilinear – the connection line always remains either horizontal or vertical.</li> <li>• Tree – arranges the connection lines on the Flow Design in a tree structure.</li> </ul> </li> <li>• Avoid obstructions – connection lines go around objects.</li> <li>• Closest distance – connection lines pass through objects to shorten the distance of the connection line.</li> </ul>
Lines and arrows	
Smoothness	<ul style="list-style-type: none"> <li>• None</li> <li>• Less</li> <li>• Normal</li> <li>• More</li> </ul>
Jump links	<ul style="list-style-type: none"> <li>• Status: <ul style="list-style-type: none"> <li>• None</li> <li>• Below</li> <li>• All</li> <li>• Above</li> </ul> </li> <li>• Type: <ul style="list-style-type: none"> <li>• Semicircle</li> <li>• Chamfered</li> <li>• Square</li> </ul> </li> <li>• Reverse jump links</li> </ul>

## 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	<p>Click <b>Color picker</b> to choose a background color for the screen.</p> <p>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.</p>
Background image	<p>Click <b>Browse</b> to choose an image to set as the background image for the screen.</p> <p><b>Note:</b> The image must be located in the project.</p>
Background image repeat	<p>Select from these options:</p> <ul style="list-style-type: none"> <li>Repeat – the background image will repeat on all screens in the mobile workflow.</li> <li>No repeat – the image does not repeat on each screen</li> <li>Inherit – child screen inherit the background image from the parent screen</li> <li>Repeat-X – the background image repeats horizontally only</li> <li>Repeat-Y – the background image repeats vertically only</li> </ul>
OK action	Select the action to perform when the user clicks <b>OK</b> on the screen.



### Rulers and grid

The ruler and grid properties are applied to the Screen Design page, or canvas.

Property	Description
Display	<ul style="list-style-type: none"> <li>• Show ruler</li> <li>• Show grid</li> <li>• Grid in front</li> </ul>
Measurement	<ul style="list-style-type: none"> <li>• Ruler units</li> <li>• Grid spacing</li> </ul>
Grid line	<ul style="list-style-type: none"> <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.

### Keys

Keys associated with the controls on the screen are shown. You can sort the table on any property by clicking its column heading.

Property	Description
Key name	The name of the associated key.
Type	The type of the associated key, for example, string.
Data binding	<p>Shows the data source to which the key is bound.</p> <p>Click <b>New</b> to create a new key.</p> <p>Select any key and click <b>Edit</b> to edit the information for an existing key.</p> <p>Select a key and click <b>Remove</b> to remove an existing key.</p>

### See also

- *Setting Mobile Workflow Forms Editor Preferences* on page 3
- *Creating a New Mobile Workflow Form* on page 6
- *Mobile Workflow Forms Editor Context Menus* on page 9

- *Starting Points* on page 10
- *Flow Design Page Properties* on page 25
- *Flow Design Palette Options* on page 29
- *Screen Design Palette Options* on page 36
- *Adding Images to the Project* on page 8

## Screen Design Palette Options

---

Use the palette to add menu items, custom actions, and controls to the mobile workflow application screen design.

These options are available in the Palette:

**MenuItem** – Adds an entry to the top-level menu on the screen.

**Custom Action** – Provides an action that you can invoke from your custom code. The Custom Actions box and its contents are not visible in the generated application form.

**Control** – Adds a visible screen object that users can interact with to view, enter, or manipulate data. The set of supported controls depends on the target device platform.

Control	Description	Supported Platforms
Checkbox	Displays a true/false (or yes/no) option on the screen.	<ul style="list-style-type: none"><li>• Apple iOS</li><li>• BlackBerry</li><li>• Android</li><li>• Windows Mobile Professional</li></ul>
Editbox	Select the logical type for the edit box, then enter text and assign values to strings, dates, decimals, integers, and so on.	<ul style="list-style-type: none"><li>• Apple iOS</li><li>• BlackBerry</li><li>• Android</li><li>• Windows Mobile Professional</li></ul>
Choice	Display a collection of items on the screen as a drop-down list from which the user can choose.	<ul style="list-style-type: none"><li>• Apple iOS</li><li>• BlackBerry</li><li>• Android</li><li>• Windows Mobile Professional</li></ul>

Control	Description	Supported Platforms
Slider	Represent numeric values within a range.	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> </ul> <hr/> <p><b>Note:</b> When the <b>Optimize for performance</b> option is set in preferences, the Slider control is supported only on BlackBerry.</p>
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.	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry 6.0</li> <li>• Android</li> </ul> <hr/> <p><b>Note:</b> When the Optimize for Performance option is set in preferences, the Signature control is not supported.</p>
HtmlView	Allows text with HTML markup to display on the screen.	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul>
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.	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul> <hr/> <p><b>Note:</b> You can have only one Listview control per screen in the mobile workflow.</p>
AttachmentViewer	Allows the user to download and view attachments on the device. A file viewer must be available on the device for the associated file type, for example, .pdf, doc., xls, and so on.	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul>

Control	Description	Supported Platforms
Label	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul>	
Link	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul>	
Button	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul> <p><b>Note:</b> On Windows Mobile, only text is supported on the Button control, not images.</p>	
Image	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul>	

**See also**

- *Setting Mobile Workflow Forms Editor Preferences* on page 3
- *Creating a New Mobile Workflow Form* on page 6
- *Mobile Workflow Forms Editor Context Menus* on page 9
- *Starting Points* on page 10
- *Flow Design Page Properties* on page 25
- *Flow Design Palette Options* on page 29
- *Screen Design Page Properties* on page 34
- *Configuring Control Properties* on page 43

## **Adding Palette Objects to the Screen Design**

Use the Screen Design palette to add menu items, custom actions, and controls to the mobile workflow screen.

1. In the Mobile Workflow Forms Editor, select **Screen Design > Palette > Controls**.
2. In the Palette, select the object you want to add to the screen design.
3. Click the appropriate area in the Screen Design page to add the selected item:
  - Add a menu item to the **Menu box**.
  - Add a custom action to the **Custom Actions box**.
  - Add a control to the screen display box.
4. If the object has a name or label, you can enter the text directly in the new object. Otherwise, specify this value later, when you configure the object's properties.

### **Next**

Use the Properties page to view and edit the new object's properties. For example, right-click the object and select **Show Properties View**.

### **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 that have lists as child elements are not supported in mobile workflow forms.

---

## **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 style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul>
Editbox	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul>
Choice	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul>
Slider	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> </ul> <hr/> <p><b>Note:</b> When the <b>Optimize for performance</b> option is set in preferences, the Slider control is supported only on BlackBerry.</p>
Signature	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry 6.0</li> <li>• Android</li> </ul> <hr/> <p><b>Note:</b> When the Optimize for Performance option is set in preferences, the Signature control is not supported.</p>
HtmlView	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul>
Listview	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul>

Control	Supported Platform
AttachmentViewer	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul>
Button	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul> <hr/> <p><b>Note:</b> The button control works on all platforms without an image and with only label settings; or with an image without label settings.</p> <p>If the button has both label and image settings, it shows only the image on iOS and Windows Mobile (no text), and it shows both the image and text on BlackBerry 6 and Android.</p> <hr/>
Label	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul>
Link	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul>
Image	<ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• BlackBerry</li> <li>• Android</li> <li>• Windows Mobile Professional</li> </ul> <hr/> <p><b>Note:</b> Dynamic images are not supported on Windows Mobile.</p> <hr/>





# 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

- *Screen Design Palette Options* on page 36

## 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	<p>This property applies to all controls except HtmlView and Listview.</p> <ul style="list-style-type: none"> <li>• Label – enter the text that describes the control on the user interface.</li> <li>• Label position – the position of the label in relation to the control it describes.</li> </ul>
Input data binding	<ul style="list-style-type: none"> <li>• Key – the unique identifier within the scope of the widget, used when setting or retrieving a value. Click <b>New key</b> to create a new key for the control.</li> <li>• Default value – default value for the control.</li> </ul> <p><b>Note:</b> If the value is not unique, the Problems view reports a missing key error.</p>

## Configuring Control Properties

Property	Description
Read only	<p>This property applies to all controls except HtmlView and Listview. Select to make the control read-only.</p> <hr/> <p><b>Note:</b> The read-only property is not supported on the Windows Mobile platform for the Checkbox, Slider, and Choice controls.</p>

### *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	<p>This property applies to all controls except HtmlView and Listview.</p> <ul style="list-style-type: none"><li>• Label – enter the text that describes the control on the user interface.</li><li>• Label position – the position of the label in relation to the control it describes.</li></ul>

Property	Description
Input data binding	<ul style="list-style-type: none"> <li>Key – the unique identifier within the scope of the widget, used when setting or retrieving a value. Click <b>New key</b> to create a new key for the control.</li> <li>Default value – default value for the control.</li> </ul> <hr/> <p><b>Note:</b> If the value is not unique, the Problems view reports a missing key error.</p>
Read only	<p>This property applies to all controls except HtmlView and Listview. Select to make the control read-only.</p> <hr/> <p><b>Note:</b> The read-only property is not supported on the Windows Mobile platform for the Checkbox, Slider, and Choice controls.</p>

#### *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.
Locale specific	This option is enabled only with DATE, DATE-TIME, and TIME logical types. If selected, the content of the cell is rendered in Locale Display format.

Property	Description
Logical type	<p>Select the logical type for the editbox:</p> <ul style="list-style-type: none"> <li>• TEXT</li> <li>• NUMERIC</li> <li>• 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.</li> <li>• 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.</li> <li>• TIME – uses only absolute values, for example, “14:30” or “14:30:15.” Valid values are 00:00:00 to 23.59:59.</li> </ul> <hr/> <p><b>Note:</b> 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.</p>

### Advanced

These are the advanced properties that apply to the editbox.

Property	Description
Content size	<ul style="list-style-type: none"> <li>• Maximum length – specify the maximum number of characters allowed for the control.</li> <li>• Number of lines – specify the number of lines for the control.</li> <li>• 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.</li> </ul>

Property	Description
Validation	<ul style="list-style-type: none"> <li>• Minimum value – specify the minimal value for numeric input. Valid only in conjunction with numeric attributes.</li> <li>• Maximum value – specify the maximum value for numeric input. Valid only in conjunction with numeric attributes.</li> <li>• Required – specify if user input is required. A value is required for submission.</li> <li>• Validation regular expression – if the user enters text in the control, the input has to comply to the provided regular expression.</li> <li>• Validation message – enter the error message the user sees in case the input validation criteria do not match.</li> </ul>
Time precision	<p>Specifies the precision of the time portion displayed to the user and calculations.</p> <hr/> <p><b>Note:</b> This property is used only if the editbox logical type is datetime or time.</p>
Credential cache username	<p>If the widget supports cached credentials, select to specify that the editbox be treated as a user name.</p>
Credential cache password	<p>If the widget supports cached credentials, select to specify that the editbox be treated as a password.</p>

## 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.

## Configuring Control Properties

Property	Description
Label	<p>This property applies to all controls except HtmlView and Listview.</p> <ul style="list-style-type: none"> <li>Label – enter the text that describes the control on the user interface.</li> <li>Label position – the position of the label in relation to the control it describes.</li> </ul>
Input data binding	<ul style="list-style-type: none"> <li>Key – the unique identifier within the scope of the widget, used when setting or retrieving a value. Click <b>New key</b> to create a new key for the control.</li> <li>Default value – default value for the control.</li> </ul> <hr/> <p><b>Note:</b> If the value is not unique, the Problems view reports a missing key error.</p>
Read only	<p>This property applies to all controls except HtmlView and Listview. Select to make the control read-only.</p> <hr/> <p><b>Note:</b> The read-only property is not supported on the Windows Mobile platform for the Checkbox, Slider, and Choice controls.</p>

### *Choice general properties*

These are the general properties that apply to the choice control.

Property	Description
Logical type	<p>Select the logical type for the edit box from the drop-down list. Logical types include:</p> <ul style="list-style-type: none"> <li>TEXT</li> <li>NUMERIC</li> </ul> <hr/> <p><b>Note:</b> 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.</p>

Property	Description
Options	<ul style="list-style-type: none"> <li>• Static – the choice contents are static. Click <b>Add</b> to assign a display name and value pair to the choice item.</li> <li>• Dynamic – the choice contents are dynamic.</li> </ul>

### Advanced

These are the advanced properties that apply to the choice control.

Property	Description
Content size	<p>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.</p> <hr/> <p><b>Note:</b> This property is used only if the logical type for the choice control is set to numeric.</p>
Validation required	<ul style="list-style-type: none"> <li>• Required – specify if user input is required. A submission without a value is not possible.</li> <li>• Validation message – enter the error message displayed to the user in case the input validation criteria do not match.</li> </ul>

## 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 a unique 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 a unique name for the choice item as it will appear on the user interface.
Value	Enter a unique 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**, select the value for the key.
4. In **Display name key**, select the value that the user will see on the device interface.

---

#### **Notes:**

- Each key must be unique; otherwise the names or value returned in the form will be unpredictable. Because dynamic content is not resolved until run-time, the Forms Editor cannot prevent you from entering duplicate value keys and display name keys.
  - 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.
- 

### Configuring a Choice Control to Use Data from an MBO

Configuring a choice control whose items come from a data source

This task refers to the choice control configuration in the PurchaseOrderSample sample application, to illustrate several steps. The Sybase Unwired Platform sample applications are available at <https://cw.sdn.sap.com/cw/groups/sup-apps>.



1. Create a Mobile Business Object using a data source that defines the data to be included in the choice control.  
Example: in the sample application, the PurchaseOrderApprover MBO uses the PurchaseOrderApprover table.
2. In the Mobile Application Diagram, select the MBO. In the MBO properties, define an object query that accesses the records you want to use in your choice control.  
Example: in the sample application, the PurchaseOrderApprover MBO defines a query with these values:

**Name:** findByName

**Return type:** Multiple objects

**Create an index:** selected

**Query definition:** `SELECT x.* FROM PurchaseOrderApprover x  
WHERE x.name LIKE :parameter1`

---

**Note:** The parameter is resolved later, when the operation that invokes this query passes in its value.

---

3. Deploy the project.
4. Open the deployed XBW package file in the Mobile Workflow Forms Editor and add the screens for your application.
5. In your application's flow design, open the screen that will contain your choice control. Add the key that you will bind to the choice control:
  - a) In the Keys page, click **New**.
  - b) In the Key dialog, enter the name of the key you want to use. Select list as the type, and select the options, Enable Sent by server and MBO object query results.
  - c) Click **OK**.

Example: the sample application uses the key, **PurchaseOrderApprover\_findByName\_resultSetKey**, for the choice control.
6. In your MBO screen design, add a menu item from the Palette to the menu box.
7. Select the new menu item and configure its properties:
  - a) In the General properties page, specify the menu item name.
  - b) Choose the operation type.

These operations can invoke an online query:

Online Request

Submit Workflow

Add Key Collection

Update Key Collection

Delete Key Collection

- c) Specify the Mobile business object associated with the menu action.

## Configuring Control Properties

- d) Enable the option, **Invoke object query**, and choose the object query.
- e) In the Parameter Mappings properties page, select each parameter from the selected object query, then click **Edit** to map the parameter to an available key.

Example: the sample application invokes the **findByName** query, and binds parameter1 to the **approverNamePattern** key.

- 8. Add a choice control from the Palette to the MBO screen design. Select the new control and edit its properties to define data bindings:

- a) Specify the label name and position.
- b) Under Input Data Binding, click the **Key** control. Then select one of the available keys or specify a new key.

Example: the sample application binds

**PurchaseOrder\_create\_purchaseOrderApproverID\_paramKey**.

- c) Enable the **Dynamic** option. Bind each field to the key that you created in the screen's flow design.

Example: the sample application uses these keys:

- Value key - **PurchaseOrderApprover\_findByName\_resultSetKey.PurchaseOrderApprover\_ID\_attribKey**
- Display name key - **PurchaseOrderApprover\_findByName\_resultSetKey.PurchaseOrderApprover\_name\_attribKey**

- 9. Save your form.

## 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	<p>This property applies to all controls except HtmlView and Listview.</p> <ul style="list-style-type: none"><li>• Label – enter the text that describes the control on the user interface.</li><li>• Label position – the position of the label in relation to the control it describes.</li></ul>

Property	Description
Input data binding	<ul style="list-style-type: none"> <li>Key – the unique identifier within the scope of the widget, used when setting or retrieving a value. Click <b>New key</b> to create a new key for the control.</li> <li>Default value – default value for the control.</li> </ul> <hr/> <p><b>Note:</b> If the value is not unique, the Problems view reports a missing key error.</p>
Read only	<p>This property applies to all controls except HtmlView and Listview. Select to make the control read-only.</p> <hr/> <p><b>Note:</b> The read-only property is not supported on the Windows Mobile platform for the Checkbox, Slider, and Choice controls.</p>

#### *Slider control general properties*

These are the general properties that apply to the slider control.

**Table 1. 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 when the **Optimize for Performance** option is set in preferences.

---

## Configuring Control Properties

### *General control properties*

These are general properties for most controls.

Property	Description
Label	<p>This property applies to all controls except HtmlView and Listview.</p> <ul style="list-style-type: none"><li>• Label – enter the text that describes the control on the user interface.</li><li>• Label position – the position of the label in relation to the control it describes.</li></ul>
Input data binding	<ul style="list-style-type: none"><li>• Key – the unique identifier within the scope of the widget, used when setting or retrieving a value. Click <b>New key</b> to create a new key for the control.</li><li>• Default value – default value for the control.</li></ul> <hr/> <p><b>Note:</b> If the value is not unique, the Problems view reports a missing key error.</p>
Read only	<p>This property applies to all controls except HtmlView and Listview. Select to make the control read-only.</p> <hr/> <p><b>Note:</b> The read-only property is not supported on the Windows Mobile platform for the Checkbox, Slider, and Choice controls.</p>

### *Advanced*

These are the advanced properties that apply to the signature control.

Property	Description
Validation Required	<ul style="list-style-type: none"><li>• Required – specify if user input is required. A submission without a value is not possible.</li><li>• Validation message – enter the error message displayed to the user in case the input validation criteria do not match.</li></ul>

## HtmlView Properties

Select the HtmlView control on the Screen Design page to view and configure its properties.

**Table 2. HtmlView control properties**

Property	Description
Input data binding	<ul style="list-style-type: none"> <li>Key – the unique identifier within the scope of the workflow form, used when setting or retrieving a value. Click <b>New key</b> to create a new key for the control.</li> <li>Default value – default value for the control.</li> </ul> <hr/> <p><b>Note:</b> If the value is not unique, the Problems view reports a missing key error.</p>

## Listview Properties

Select the Listview control on the Screen Design page to view and configure its properties.

You can have only one Listview control per screen. The Listview control is bound to a list key whose child keys exist in every row. By default, the screen shows only keys that are bound to attributes; keys that are bound to parameters are not shown. That is, if you create a row using keys bound to attribute, the new row contains non-empty values in the visible columns. Conversely, if you create a row using keys bound to parameters, the non-empty values occur in invisible columns, and so the listview shows an empty row.

### *General Listview properties*

Property	Description
Input data binding	<p>Select the key with which to bind the Listview control from the drop-down, or click <b>New key</b> to create a new key.</p> <hr/> <p><b>Note:</b> If the value is not unique, the Problems view reports a missing key error.</p>

## Configuring Control Properties

Property	Description
Alternate row color	<p>Click <b>Color Picker</b> to select the color for the alternate rows in the list.</p> <p>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.</p>
On empty list	The text that appears on the screen when the List-view is empty.
Show header	When selected, column titles appear in the List-View control.
Wrap data in PRE and CODE tags	<p>If the text to be displayed on the device contains HTML, HTML tags, or HTML or XML escape characters, or is enclosed inside &lt; or &gt; (less than and greater than characters), use this option to display the data as text on the device. If this option is selected, HTML markup and tags, and other characters, are filtered out.</p> <p>For example, if this is the input: Sybase &lt;an SAP Company&gt; SUP and you do not select this option, the output looks like: Sybase SUP on the device. The text inside the &lt;&gt; characters is not displayed. To display the text, select this option.</p> <p>For example: &amp;apos; &amp;amp; &amp;lt; &amp;gt; &amp;#955; &amp;#X03bb;</p> <ul style="list-style-type: none"> <li>• \]</li> <li>• \xA9</li> <li>• \251</li> <li>• \u00AE</li> <li>• \uQA1029</li> <li>• &amp;quot;</li> </ul>

Property	Description
Locale specific display	This option is enabled only with DATE, DATE-TIME, and TIME logical types. If selected, the content of the cell is rendered in Locale Display format.
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 <b>Add</b> to add a new line. Select an existing line and click <b>Delete</b> to remove a line from the row. Use the <b>Up</b> and <b>Down</b> buttons to move the cell lines up or down.
Fields for cell line	Click <b>Add</b> to create a new field. Click <b>Edit</b> to edit an existing field. Click <b>Delete</b> to delete an existing field. Use the <b>Up</b> and <b>Down</b> buttons to move the cell line fields up or down.

## Adding or Editing a Listview Field

In the Properties view, add or modify Listview fields.

### Prerequisites

- In the Flow Design for your screen, define any keys that you bind to your fields. See *Screen Design Page Properties* on page 34.
- If you are adding any static images, include the images in your project. See *Adding Images to the Project* on page 8

### Task

1. Select the Listview control to display its properties.
2. In the left pane of the Properties view, click **Cell**.
3. To add a new cell line in the listview, click **Add** under *Cell lines*.
4. In the *Cell lines* table, select a line where you want to add or edit a field.

## Configuring Control Properties

5. Under *Fields for cell line*, click **Add** to add a new field to the cell, or **Edit** to modify an existing field.
6. In the Listview Field dialog, define the field properties.
7. Click **OK**.

### Listview Field Properties

Listview properties describe the appearance of a listview control on the screen.

Option	Description
Key	The key with which to associate the field. The drop-down control shows values defined in the screen's Flow Design properties.
Field Width	A percentage value between 1 – 100. The widths of all fields should add up to 100.
Fonts	One of these font types: <ul style="list-style-type: none"><li>• Plain</li><li>• Bold</li><li>• Italic</li></ul>
Field Header Title	A text label for the cell line.
Image Height	A dimension in pixels. A value in this field indicates that the field contains an image instead of text. <hr/> <b>Note:</b> Image scaling does not work correctly on BlackBerry 9700 devices.
Static Image	For an image field: <ul style="list-style-type: none"><li>• If selected, the image must be specified by the application backend database, and included in the application.</li><li>• If unselected, the image is not specified in the application, but at run time in server or browser code.</li></ul>

### Generated Keys

Listview controls are bound to a list key, however, the listview may only display a subset of that list key's child keys.

For example, when you have created a listview screen by dragging and dropping a mobile business object (MBO), you get a list key that has child keys bound to attributes and child keys bound to the create, update and/or delete operations' parameters. The listview, however, displays only the keys bound to attributes.



If you create a row using parameters bound to attribute keys, you get a new row with non-empty values visible in the listview. If you create a row using parameters bound to parameter keys, you get a new row with empty values visible in the listview, the non-empty values are in keys not displayed in the listview. For some MBO operations it would not be advisable to bind parameters to attribute keys (for example, an MBO with first\_name and last\_name attributes and a create operation that had one full\_name parameter); but for many MBO operations, specifically those automatically generated for MBOs with database backends, it is appropriate.

## AttachmentViewer Properties

---

Select AttachmentViewer on the Screen Design page to view and configure its properties.

### General

Property	Description
Label	<ul style="list-style-type: none"> <li>Label – enter the text that describes the control on the user interface.</li> <li>Label position – select the position of the label in relation to the control it describes.</li> </ul>
Input data binding	<p>Select the key with which to bind the AttachmentViewer control, or click <b>New key</b> to create a new key. The key represents a binary attribute, which will be the contents of the attachment.</p> <ul style="list-style-type: none"> <li>Key – the unique identifier within the scope of the workflow form, used when setting or retrieving a value.</li> <li>Default value – default value for the control.</li> </ul> <p><b>Note:</b> If the value is not unique, the Problems view reports a missing key error.</p>
Read-only	Select to make the control read-only.

Property	Description
<p>Content</p>	<p>The other way you can bind the Attachment-Viewer to data is by using an object query and mapping the parameters of the object query to keys. Indicate which binary attribute of the object query corresponds to the contents of the attachment.</p> <ul style="list-style-type: none"> <li>• Use object query – select to invoke a specified object query.               <ul style="list-style-type: none"> <li>• Mobile business object – select the mobile business object that contains the object query to invoke.</li> <li>• Object query – specify the object query to invoke.</li> <li>• Attribute –specify the binary attribute value that corresponds to the attachment contents.</li> </ul> </li> </ul> <hr/> <p><b>Note:</b> Ensure that the MBO binary attribute has a size specified that is big enough to support the size of the attachment when developing the MBO. To do this, you must change the MBO's binary attribute from BINARY to BINARY(%) where % is larger than the largest binary content you want to store.</p> <p><i>See Sybase Unwired WorkSpace - Mobile Business Object Development &gt; Develop &gt; Working with Mobile Business Objects &gt; Mobile Business Object Data Properties &gt; Datatype Support for information about setting the binary attribute size.</i></p> <hr/> <ul style="list-style-type: none"> <li>• MIME type key – MIME type for attachments that are supported, for example, *.doc, *.xls and so on.</li> <li>• File name.</li> <li>• File size.</li> </ul> <hr/> <p><b>Note:</b> Attachments requested on demand are limited to 5MB after Base64 encoding.</p>

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.

For information about these parameters and how they map to MBO operations and queries, please see *WorkflowClient.xml File in Sybase Unwired WorkSpace - Mobile Workflow Package Development*.

### *Personalization Key Mappings*

If you have a mobile business object with load parameters mapped to transient personalization keys, you can specify values for those personalization keys when you invoke an operation or object query from the Mobile Workflow package. This can, for example, be useful for loading only specific data into the consolidated database from large data sources.

## **Adding and Configuring the AttachmentViewer**

The AttachmentViewer control allows you to download and view attachments.

1. In the Mobile Workflow Forms Editor, select **Screen Design > Palette > Controls**.

2. Click **AttachmentViewer** and then click in the screen.

3. From the Properties view, configure the Attachment Viewer:

a) Select the **AttachmentViewer** control.

b) Click the **Properties** tab.

If you do not see the Properties tab, right-click anywhere in the Screen Design page, and select **Show Properties View**.

c) Click **General** in the left pane to configure the general properties for the AttachmentViewer.

You can bind the AttachmentViewer control to data in one of two ways:

- Keys, or
- Object queries

---

**Note:** MBO attributes displayed as attachments in a mobile workflow application must be binary type, no matter what the corresponding backend data type is.

---

4. (Optional) Click **Parameter Mappings** 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

For information about the parameters and how they map to MBO operations and queries, please see *WorkflowClient.xml File in Sybase Unwired WorkSpace - Mobile Workflow Package Development*.

5. (Optional) Click **Personalization Key Mappings** if you have a mobile business object with load parameters mapped to transient personalization keys, and want to specify values for those personalization keys when you invoke an operation or object query from the Mobile Workflow package.

## Configuring Control Properties

This can be useful for loading only specific data into the consolidated database from large data sources.

### 6. Select **File > Save**.

### **Supported Attachment Types**

This shows which device platforms support which attachment file type.

<b>Device Platform</b>	<b>File Types</b>
iPhone	*.jpg, *.jpeg, *.png, *.tif, *.tiff, *.gif, *.doc, *.docx, *.xls, *.xlsx, *.ppt, *.pptx, *.pdf, *.htm, *.html, *.txt
BlackBerry	*.gif, *.jpg, *.png, *.bmp, *.txt, *.html  Install an attachment viewer like Documents To Go Premium Edition to view additional file types, such as *.pdf, *.doc, *.xls, *.xlsx, *.ppt, and *.pptx.
Windows Mobile	*.doc, *.xls, *.xlsx, *.ppt, *.pptx, *.gif, *.jpg, *.png, *.bmp, *.txt, *.html, *.mp3  PDF support requires the user to install a PDF viewer on the device.
Android	*.gif, *.jpg, *.png, *.bmp, *.txt, *.html  Other file types, like Microsoft and Adobe, can be supported by installing additional applications on the device.

### **AttachmentViewer and Image Limitations**

There are some limitations on the size of the attachments and images that you can include as part of the Mobile Workflow message.

These limitations vary by platform.

Platform	Size Limit
BlackBerry 5.0 and BlackBerry 6.0	On BlackBerry 5.0, the maximum size of a JavaScript variable is 500KB and on BlackBerry 6.0 and later, the maximum size is 2MB, which means that the maximum size of the attachment included as part of a Mobile Workflow message must be less than that. Keep in mind that you have to include the rest of the Mobile Workflow message, in addition to the attachment contents, and also include allowances for an increase in the size of the attachment when it is Base64-encoded to allow it to be included in the XML message.
Windows Mobile	The maximum size of a JavaScript variable for Windows Mobile is 2MB, which allows for more memory. Warning messages are shown if the script continues for a long time, which can cause the memory to run out.
iOS	Large attachments can produce longer processing times.
Android	Large attachments can produce longer processing times. There is a 1MB limit for attachments on Android devices.

---

**Note:** When accessing very large binary (image) data in the mobile business object associated with the mobile workflow, ensure that the attribute set in the mobile business object is a **BigBinary** datatype, rather than Binary.

---

### See also

- *Image Properties* on page 66

## Button Properties

Select the Button control on the Screen Design page to view and configure its properties using the Properties page.

Property	Description
Key	<p>A key, unique within the current scope, that can be used to identify the button label.</p> <hr/> <p><b>Note:</b> If the value is not unique, the Problems view reports a missing key error.</p>
Default value	The text displayed in the label by default (optional).
Label position	<p>The position of the label in relation to the control it describes. Available positions:</p> <p>LEFT RIGHT TOP CENTER CENTER_TOP CENTER_BOTTOM</p>
Has image	When selected, indicates that the button uses an image, and enables the other image-related properties.
Key	A key used to identify the button image source. You can selected an existing keys, or click <b>New key</b> to create a new one.
Default value	The path of the image displayed in the button by default (optional).
Image Dimension	The width and height, in pixels, used to display the button image.

Property	Description
Options	<p>One of these:</p> <ul style="list-style-type: none"> <li>• Static image source – the value for the image is the file path to the image file, which may come from the client or server side.</li> <li>• Dynamic image source (Binary Data) – the value for the image is binary data. This can come from the client or server side.</li> </ul> <hr/> <p><b>Note:</b> Dynamic images are not supported on Windows Mobile.</p>
Type	One of these:

## Label Properties

---

Select the Label control on the Screen Design page to view and configure its properties using the Properties page.

- Key – the unique identifier within the scope of the widget, used when setting or retrieving a value.  
Click **New key** to create a new key for the control.
- Default value – default value for the control.

---

**Note:** If the key value is not unique, the Problems view reports a missing key error.

---

## Link Properties

---

Select the Link control on the Screen Design page to view and configure its properties using the Properties page.

Limitations of the Link control include:

- On the iOS platform, you cannot return to the mobile workflow once you switch to a different application. For example, if you click the link to send an e-mail message, you cannot return to the previously open mobile workflow; you must relaunch it.
- On the iOS simulator, clicking the **Mailto** link has no effect since there is no e-mail client on the simulator.

## Configuring Control Properties

Property	Description
Label	The text that appears comes from the key or, if that is not supplied, from the default value.
Input data binding	<ul style="list-style-type: none"><li>• Key – the unique identifier within the scope of the widget, used when setting or retrieving a value. Click <b>New key</b> to create a new key for the control.</li><li>• Default value – default value for the control.</li></ul> <hr/> <b>Note:</b> If the key value is not unique, the Problems view reports a missing key error. <hr/>
HREF settings	<ul style="list-style-type: none"><li>• Logical type – choose the logical type. The logical type controls the values of the text fields prefix and suffix, which are used to build the content of the anchor. For example: <pre>&lt;a href=mail-to:user@&lt;email_address&gt;send mail&lt;/a&gt;</pre> Selecting the logical type enables the prefix and suffix fields. Choose <b>Custom</b> to supply your own prefix and suffix.</li><li>• Prefix – if you select Custom as the logical type, you can enter your own custom prefix.</li><li>• Suffix – if you select Custom as the logical type, you can enter your own custom suffix.</li></ul>

## Image Properties

---

Select the Image control on the Screen Design page to view and configure its properties using the Properties page.

---

**Note:** The same limitations that apply to the AttachmentViewer limitations apply to the Image control. See the topic *AttachmentViewer Limitations*.

---



Property	Description
Alternative Text	(Optional) Appears as text on the image if the image source is invalid. In the generated HTML file, the alternate text is the value of the alt attribute for the <img> tag.
Input Image Source Binding	<ul style="list-style-type: none"> <li>• Key – the unique identifier within the scope of the widget, used when setting or retrieving a value. Click <b>New key</b> to create a new key for the control.</li> <li>• Default value – default value for the control.</li> </ul> <hr/> <p><b>Note:</b> If the value is not unique, the Problems view reports a missing key error.</p>
Image Dimension	The width and height, in pixels, used to display the button image.
Options	<p>One of these:</p> <ul style="list-style-type: none"> <li>• Static image source – the value for the image is the file path to the image file, which may come from the client or server side.</li> <li>• Dynamic image source (Binary Data) – the value for the image is binary data. This can come from the client or server side.</li> </ul> <hr/> <p><b>Note:</b> Dynamic images are not supported on Windows Mobile.</p>

**See also**

- *Attachment Viewer and Image Limitations* on page 62



# Configuring a Menu Item or Custom Action

Configure a menu item or custom action item in the Mobile Workflow application.

## Prerequisites

From the Screen Design Palette, add the item that you want to customize. See Adding Palette Objects to the Screen Design.

## Task

Whether you define a menu item or a custom action, the action results in a method being generated when you generate the files for the Mobile Workflow package. If the action is associated with a menu item, a native menu is generated for the supported BlackBerry, Android, and Windows Mobile platforms. On the iOS platform, menus are provided from HTML controls that are designed to emulate the iPhone navigation bar and toolbar.

If you create a menu item or action that invokes an object query, it automatically creates an output key, which is a list and which has keys that are bound to the attributes of the mobile business object.

## Menu Item and Custom Action Properties

---

The Properties view shows properties for a selected item in the menu custom action box on the Screen Design page.

### General

Property	Description
Name	Enter a valid name for the menu or custom action.  <b>Note:</b> To avoid wrapping of menu item names on Apple iOS devices, keep the name length to 10 or fewer characters.
Key	Select the key with which to bind the item.
Default	For a menu item, select to display the menu item prominently on the screen.

Property	Description
Type	<ul style="list-style-type: none"> <li>• Save screen – closes the current screen, validates the current input and, if validation succeeds, saves it.</li> <li>• Cancel screen – discards any changes on the current screen and closes it. No validation occurs.</li> <li>• Close workflow – discards any changes on any open screens and closes all open screens. No validation occurs.</li> <li>• Open screen – opens a different screen.</li> <li>• Online request – creates a synchronous call to the server, 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.</li> <li>• Submit workflow – creates an asynchronous call to the server, validates the data input on all open screens, closes the application, and sends the data to the server. You cannot use a Submit workflow action to invoke an operation on a nested, or child, element.</li> <li>• Select certificate – opens a dialog where the user can choose a certificate to use for the credentials.</li> <li>• Create key collection – opens the specified screen in the context of adding a new key collection to the list key bound to the list view.</li> <li>• Add/Update/Delete Key Collection – adds, updates, or deletes a key from a key collection from a list key bound to a Listview control. These operations are not invoked immediately, but are instead invoked when a Submit operation is made on the parent MBO to which this MBO is related. A key collection can be added only to screens that are navigated to as the result of a Listview details navigation, for example, by clicking a row in a Listview.</li> <li>• Custom – write your own custom code that dictates its behavior.</li> </ul>

Property	Description
Details	You can configure actions in the Details section. The properties available depend on the action type. See the topics for configuring the different menu or custom action types for information about configuring these properties.

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.

For information about these parameters and how they map to MBO operations and queries, please see *WorkflowClient.xml File in Sybase Unwired WorkSpace - Mobile Workflow Package Development*.

### *Personalization Key Mappings*

If you have a mobile business object with load parameters mapped to transient personalization keys, you can specify values for those personalization keys when you invoke an operation or object query from the Mobile Workflow package. This can, for example, be useful for loading only specific data into the consolidated database from large data sources.

### *Output Keys*

Lists the keys of which values will be filled by the server and returned to the client. Output keys are used only with an Online Request menu or custom action type.

## **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 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	<p>Enter a valid name for the menu item.</p> <hr/> <p><b>Note:</b> To avoid wrapping of menu item names on Apple iOS devices, keep the name length to 10 or fewer characters.</p>
Key	Select the key with which to bind the menu item.
Default	Select to display the menu item prominently on the screen.
Type	<p>Select the type of action the menu item will perform when selected:</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• Close – allows the user to leave the client-side component no matter which screen the client-side component currently shows.</li> <li>• 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.</li> <li>• 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.</li> </ul>
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 Workflow Menu Item

A Submit Workflow menu item allows the user to update, delete, or create data even when not connected to the network. When the user connects, data is automatically sent and received.

The submit workflow menuitem of the first screen in mobile workflow forms editor must be associated with a mobile business object (MBO) operation or an invoke parent update.

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 new menu item.
3. In the Properties view of the menu item, configure the properties for the submit menu item:

Property	Description
Name	Enter a valid name for the menu item.  <b>Note:</b> To avoid wrapping of menu item names on Apple iOS devices, keep the name length to 10 or fewer characters.
Key	Select the key with which to bind the submit menu item.
Default	Select to display the menu item prominently on the screen.
Type	Select <b>Submit Workflow</b> .

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 MBO with which the menu item is associated. Click <b>Search</b> to locate the MBO.
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 MBO operation with which the menu item is associated.  Select the operation from the list, which is populated when you select the MBO.
Submit confirmation message	(Optional) Enter text to show in a message box on the screen when the submit operation finishes successfully.

## Configuring a Menu Item or Custom Action

Property	Description
Resubmit confirmation message	If the option <b>Mark message as processed after processing</b> is enabled in the Flow Design page properties, show the resubmit confirmation message if the user attempts to submit the workflow after it has already been previously submitted.
Error screen	The screen to which the user navigates in case an asynchronous operation invocation fails. Select an existing screen, or click Generate Error Screen to create a new one.  If no error screen is specified, the client application receives no notification of the failure — failures appear only in the log files.
Subject	If an error occurs, a notification is sent to the device with the appropriate error log keys and specified Subject.  <b>Note:</b> The subject can reference the screen name and action name using macros (except in server-initiated workflows). For example:  "An error occurred trying to perform the <code>#{action}</code> action on the <code>#{screen}</code> screen."  <hr/>
From	When an error occurs and the notification appears in the inbox, the comment appears in the From field of the notification.  <b>Note:</b> The subject can reference the screen name and action name using macros (except in server-initiated workflows). See the <a href="#">Subject example</a> .  <hr/>

5. Select **File > Save**.

## Creating an Online Request Menu Item

---

Create an online request menu item for synchronous calls to Unwired Server, which means the device client waits for a response from the server.

1. From the palette on the Screen Design page, click **Menuitem**, then click the Menu box.
2. In the Properties view, select the menu item for which to configure the properties.



3. In the Properties view, configure the properties for the online request menu item:

Property	Description
Name	Enter a valid name for the menu item. <b>Note:</b> To avoid wrapping of menu item names on Apple iOS devices, keep the name length to 10 or fewer characters.
Key	Select the key with which to bind the online request menu item.
Default	Select to display the menu item prominently on the screen.
Type	Select <b>Online Request</b> .

4. If the menu item is an operation associated with a mobile business object (MBO), you can configure these properties in the Details section:

Property	Description
Mobile business object	The MBO with which the menu item is associated. Click <b>Search</b> to locate the MBO.
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 MBO operation with which the menu item is associated.  Select the operation from the list, which is populated when you select the MBO.
Invoke object query	Invoke the object query from the menu item.
Generate old value keys	By default, this is selected, which means old value keys are generated. If you need the old value keys for update or parent update operations, you should leave this selected. If you unselect it, old value keys will not be generated.
Show credential screen on authentication failure	If selected, the credential screen is shown when authentication fails. If unselected, the authentication failure is treated the same as other types of failures.

## Configuring a Menu Item or Custom Action

Property	Description
Submit error message	<p>Enter the message the user sees if there is an error during the submit operation.</p> <p>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 child instance corresponds to an error message.</p>
Timeout	Enter the allowed amount of time, in seconds, to wait for the server to respond before timing out and giving an error message.
On device cache timeout	Represents the time, in seconds, that query results are cached on the device. If the same query is made with the same parameter values within the specified timeout, the results are pulled from the device cache and no call to the server is made.
Error screen	<p>The screen to go to if there is an error when submitting the data to the server.</p> <p>Click <b>Generate Error Screen</b> to automatically generate error screens.</p>
Success screen	<p>Choose the Default Success Screen from the drop-down. This is the screen to go to upon successful communication with the server.</p> <p>Click <b>Add</b> to add a conditional navigation screen. A conditional navigation adds a callout during Mobile Workflow package generation to the <b>customConditionalNavigation</b> function, which returns a boolean. Each piece of custom logic is evaluated, in order, one at a time, and when the first condition comes back as true, the user is navigated to the matching screen for that condition.</p>

5. (Optional) Click **Parameter Mappings** to define the keys to map to parameters.

You see the parameters defined for the MBO and used to generate the object query. In addition to Key, you can map parameters to BackEndPassword, BackEndUser, DeviceId, DeviceName, DeviceType, UserName, MessageId, ModuleName, ModuleVersion, and QueueId.

For information about these parameters and how they map to MBO operations and queries, please see *WorkflowClient.xml File in Sybase Unwired WorkSpace - Mobile Workflow Package Development*.

6. (Optional) Click **Personalization Key Mappings** to select a personalization key and the key to which it maps.  
You can also fill the personalization key values from values extracted from the notification, depending on from where you are invoking the object query.
7. (Optional) Click **Output Keys** to view the keys that will be created and populated based on the selected action.
8. Select **File > Save**.

## Creating a Key Collection Menu Item

---

Create and configure a Key Collection menu item.

Key collection menu items can be used only on screens with a Listview control.

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. <b>Note:</b> To avoid wrapping of menu item names on Apple iOS devices, keep the name length to 10 or fewer characters.
Key	Select the key with which to bind the key collection menu item.
Default	Select to display the menu item prominently on the screen.
Type	Select one of these types: <ul style="list-style-type: none"> <li>• Add Key Collection</li> <li>• Update Key Collection</li> <li>• Delete Key Collection</li> </ul>

4. If the menu item is an operation associated with a mobile business object, configure these properties in Details.

## Configuring a Menu Item or Custom Action

Property	Description
Mobile business object	The mobile business object with which the menu item is associated. Click <b>Search</b> 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	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.
Invoke object query	Invoke the object query from the menu item.
Generate old value keys	By default, this is selected, so that old value keys are generated. If you need the old value keys for update or parent update operations, leave this selected. If you unselect it, old value keys are not generated, which results in a much smaller message, which improves performance.
ListView	Select the list key from the drop-down list.

5. Select **File > Save**.

## Configuring Parameter Mappings

---

Configure mapping between parameters and keys or context data.

The Properties view shows parameter mappings for the selected operation or object query. For information about the parameters and how they map to MBO operations and queries, please see *WorkflowClient.xml File in Sybase Unwired WorkSpace - Mobile Workflow Package Development*.

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.

For information about these parameters and how they map to MBO operations and queries, please see *WorkflowClient.xml File in Sybase Unwired WorkSpace - Mobile Workflow Package Development*.

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.

## Defining a Custom Menu Item or Custom Action

---

Associate a custom action with a menu item, custom action item, or control, and write your own code to dictate its behavior.

### Prerequisites

From the Screen Design Palette, add the item that you want to customize.

### Task

1. In the Screen Design page of the Mobile Workflow Forms Editor, select the menu item, control, or custom action that you want to configure.
2. In the Properties view Type field, select **Custom**.
3. Edit the `Custom.js` file, adding custom code in the `customBeforeMenuItemClick` to handle the action.  
In your code, call the custom function that is autogenerated in `workflow.js`
4. Save the `Custom.js` file.

## Configuring a Menu Item or Custom Action

# Localization and Internationalization

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 mobile workflow applications.

All the localizable strings in the Mobile Workflow Forms Editor XML model work as resource keys in the localization properties file. All the localization properties files are in the same directory as the Mobile Workflow packages (.xbw files).

Resource keys are divided into these categories, which include all the elements of the Mobile Workflow Forms Editor XML model:

- Menus
- Controls
- Screens

Localization consists of two levels of localization—the Mobile Workflow Forms Editor XML model localization and the Mobile Workflow client localization.

All locale properties files are saved in the same directory as the Mobile Workflow package.

To ensure that the correct locale is picked up for the Mobile Workflow container, the following mechanism is used:

1. If a precise match is found for language and country, for example, English - United States (en-us) is the locale and the file exists in `html\en-us\workflow*.html`, that file is used and the HTTP lang parameter is set to "en-us."
2. If a precise match for country is not found, the language is used. For example, English (en). If the file exists in `html\en\workflow*.html`, that file is used and the HTTP lang parameter is set to "en."
3. If a language match is not found, the default locale is used. If the file exists in `html\default\workflow*.html`, that file is used and the HTTP lang parameter is set to "default";
4. If a default match is not found, no locale is used. If the file exists in `html\workflow*.html`, that file is used and the HTTP lang parameter is set to "".

## Localization Limitations

---

Some restrictions for the locale properties files apply:

- Traditional Chinese characters are not supported on iOS.

## Localization and Internationalization

- 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.
- 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.

---

**Note:** BlackBerry 9800 Asia simulators do not have a place to specify a country name, so you can specify only a language.

---

- If you specify a variant, the country code must be a 2-letter code.

### See also

- *Localizing a Mobile Workflow Package* on page 82
- *Mobile Workflow Package Internationalization* on page 91
- *Internationalization on the Device* on page 93

## Localizing a Mobile Workflow Package

---

Use the Mobile Workflow Forms Editor to complete these tasks to localize Mobile Workflow packages (.xbw files).

1. *Changing the Encoding Type*  
Change the encoding type in Preferences.
2. *Creating and Validating a New Locale Properties File*  
Goal: Create a locale properties file as the default locale.
3. *Editing the Locale Properties File*  
Goal: Edit the locale properties file.
4. *Deploying the Mobile Application Project*  
Goal: Deploy the mobile business objects to Unwired Server.
5. *Configuring the Device for Localization*  
Goal: Configure the device.



## 6. *Removing a Locale*

Remove locale properties files.

## 7. *Updating the Current Locale*

Update the currently loaded locale properties file with the resource keys from the current Mobile Workflow Designer.

### **See also**

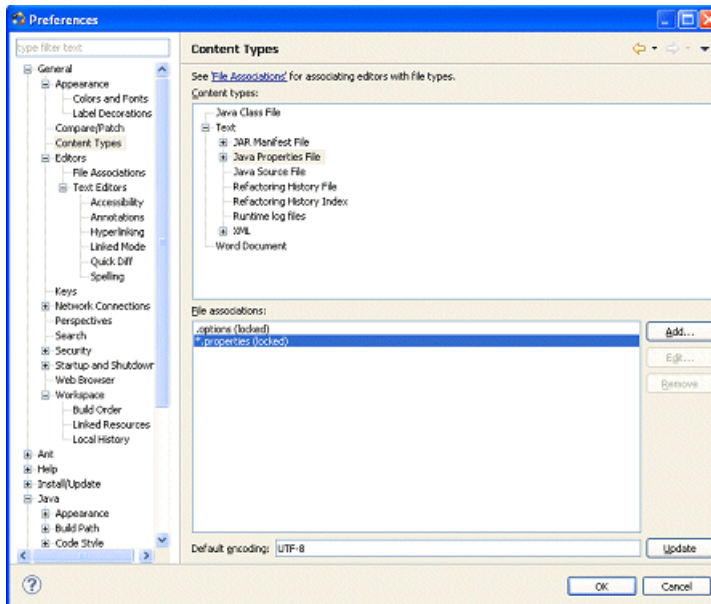
- *Localization Limitations* on page 81
- *Mobile Workflow Package Internationalization* on page 91
- *Internationalization on the Device* on page 93

## **Changing the Encoding Type**

Change the encoding type in Preferences.

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. 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**.



## Creating and Validating a New Locale Properties File

**Goal:** Create a locale properties file as the default locale.

### Prerequisites

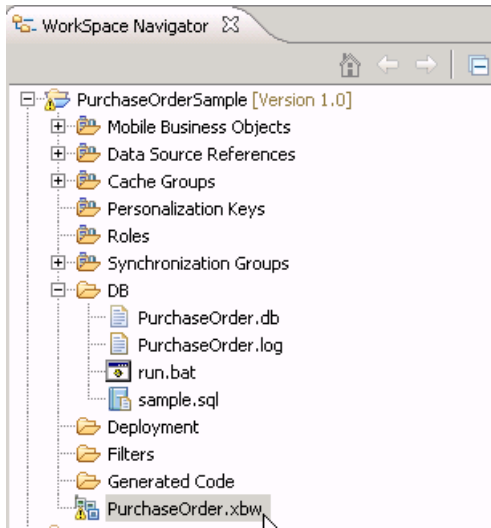
You must have an existing Mobile Workflow package before you create the locale properties file.

### Task

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 WorkSpace Navigator, double-click the <mobile\_workflow> .xbw file to open the Mobile Workflow Forms Editor.



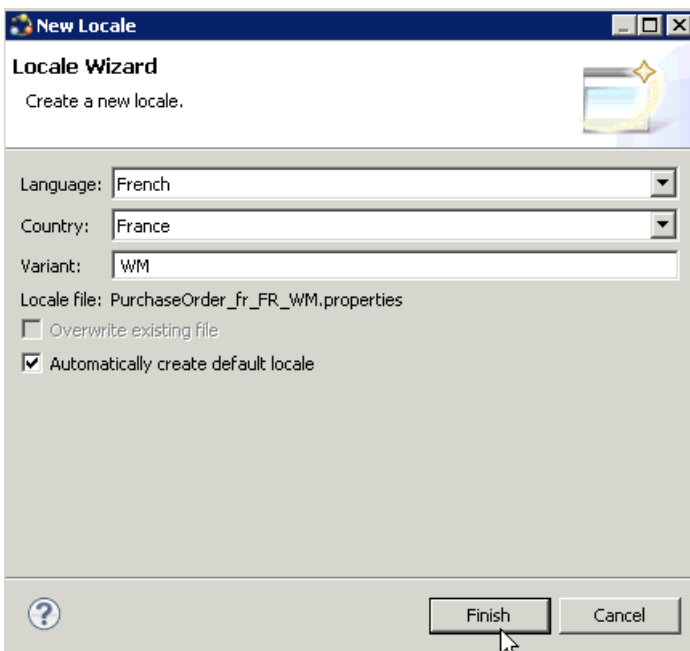
2. Click the **Flow Design** tab.
3. Right-click in a blank area on the Flow Design page, and select **Show Properties View**.
4. In the Properties view, on the left, click the **Localization** tab.
5. In the right pane, click **New**.
6. Select or enter the information for the new locale, select **Automatically create default locale**, and click **Finish**.

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.

Option	Description
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.

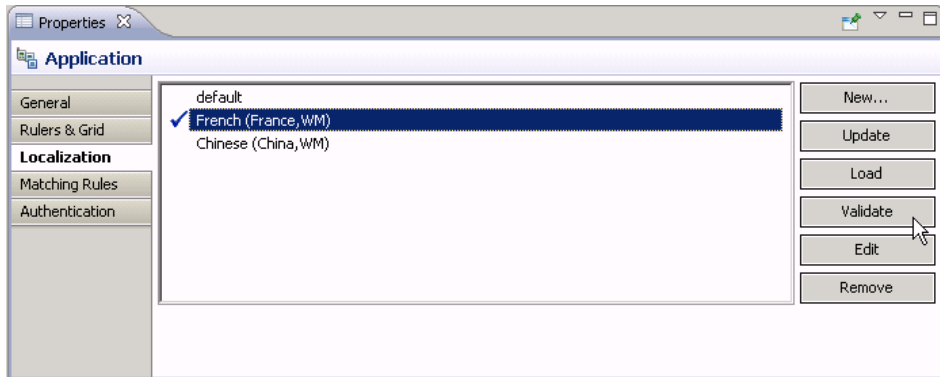
For example:

- Language – select **French**.
- Country – select **France**.
- Variant – enter a value to make this locale file unique from others, for example, WM for Windows Mobile.



This locale file is now the default locale file, and will be used when the regional setting of the device does not match that of any supplied locale file.

7. In the Properties view, in the Localization page, select the file to validate and click **Validate**.



The properties file is scanned and if there are any errors, a dialog appears. Click **Yes** to correct the errors automatically; click **No** to see the errors in the Problems view.

## Editing the Locale Properties File

**Goal:** Edit the locale properties file.

1. In WorkSpace Navigator, under the Generated Code folder, right-click the locale properties file you created, and select **Open With > Properties File Editor**.
2. You can make and save changes to the file in the Properties File editor, for example, you can replace all the values of the resource keys with Chinese characters.
3. Select **File > Save**.

The next time you open the locale properties file, notice that all of the ASCII characters have been changed.

4. In the Localization pane, select the localization file you edited, and click **Load**.

The elements of the application in the editor are translated into the language you specified if the localization file passes the loading validation.

## Deploying the Mobile Application Project

**Goal:** Deploy the mobile business objects to Unwired Server.

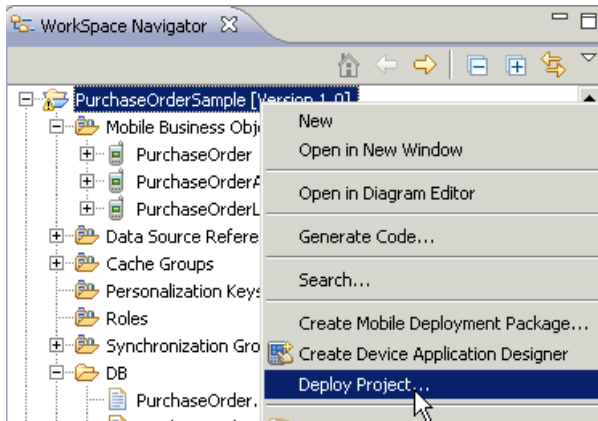
### **Prerequisites**

Unwired Server must be running.

### **Task**

Use this procedure to deploy any mobile application project you are using.

1. In WorkSpace Navigator, right-click the `<mobile_application_project>` folder, and select **Deploy Project**.



2. In the Deploy Mobile Application Project wizard, make your selections and click **Next**.
3. In the Contents page, select all the mobile business objects (MBOs) and click **Finish**.
4. In the Deployment status window, click **OK**.
5. When the deployment completes, click **File > Close**.
6. Generate the files for the Mobile Workflow package.

### Configuring the Device for Localization

Goal: Configure the device.

Keep in mind that the device you are testing may be different from the devices available to consumers in other geographies, and the locales available on your device may be different from those available on other devices.

1. For Windows Mobile:
  - a) Go to **Settings > System > Regional Settings**.
  - b) Click **Region**.
  - c) Change the region to the desired locale, for example, French (France) or Chinese (China).



---

**Note:** The locales that appear in the Regional Settings depend on what you have installed on your device.

---

2. For iPhone devices:
  - a) Go to **Settings > General > International > Language**.
  - b) Change the Language setting to the desired language, for example, Francais, and click **OK**.



- c) From **Settings > General > International > Language > Region Format**, select the corresponding Region Format and click **OK**.
3. For Android devices:
- Go to **Settings > Locale and text > Select locale**.
  - Change the region to the desired locale, for example, English (Australia).

## Removing a Locale

Remove locale properties files.

- In the Screen Design page Properties view, click **Localization**.
- Select the locale to remove and click **Remove**.
- Click **Yes** to confirm the deletion.



## 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.

---

**Note:** When you update the localization bundle, it removes all resources that are not explicitly bound to existing UI elements (screens, menuitems, controls, and so on). If you want to manually supply resources, you must do so after updating, and be careful not to update the resource bundles afterwards, or you will have to re-add those manually-supplied resources after updating.

---

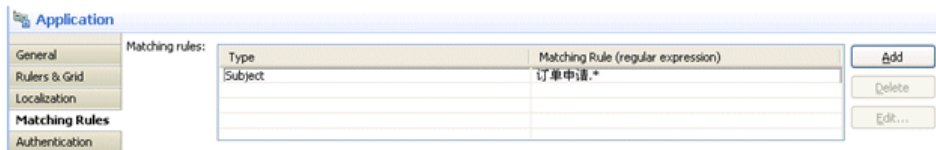
1. In the Screen Design page Properties view, click **Localization**.
2. Click **Update**.

## Mobile Workflow Package Internationalization

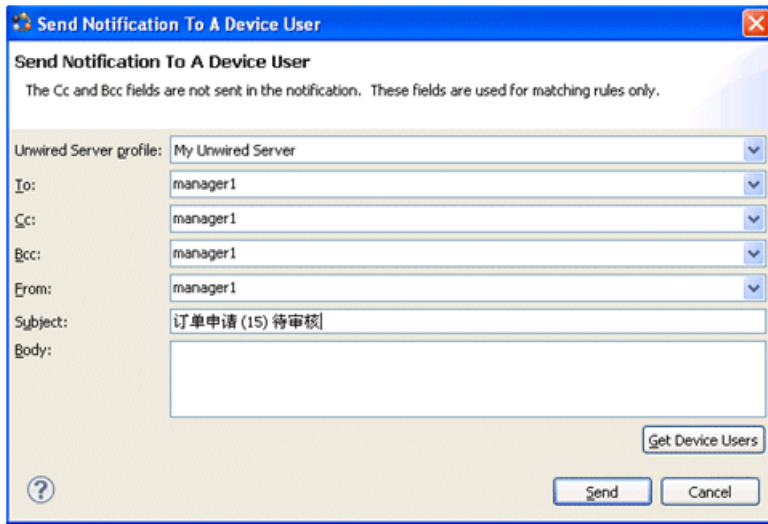
The internationalization feature depends on the internationalization setting on the operating system where Sybase Unwired Platform Mobile Workflow is running.

In the Mobile Workflow Forms Editor, you can use international data in:

- Matching rules for notifications.



## Localization and Internationalization



**Send Notification To A Device User**

The Cc and Bcc fields are not sent in the notification. These fields are used for matching rules only.

Unwired Server profile: My Unwired Server

To: manager1

Cc: manager1

Bcc: manager1

From: manager1

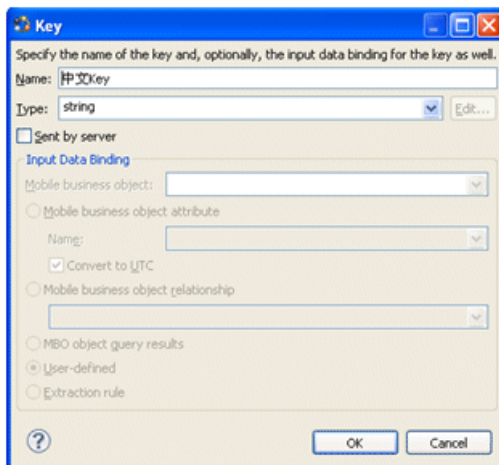
Subject: 订单申请 (15) 待审核

Body:

Get Device Users

Send Cancel

- Key names – you can create keys with names in other languages and map them to mobile business object parameters.



**Key**

Specify the name of the key and, optionally, the input data binding for the key as well.

Name: 订单Key

Type: string

Sent by server

Input Data Binding

Mobile business object:

Mobile business object attribute

Name:

Convert to UTC

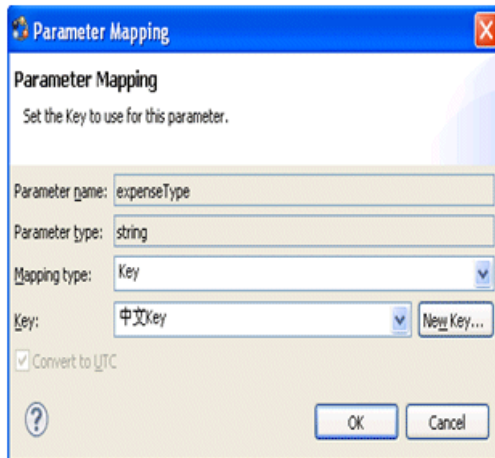
Mobile business object relationship

MBO object query results

User-defined

Extraction rule

OK Cancel



- Generated Code folder – you can include languages other than English in the code generation path based on the name of the selected language.

### See also

- *Localization Limitations* on page 81
- *Localizing a Mobile Workflow Package* on page 82
- *Internationalization on the Device* on page 93

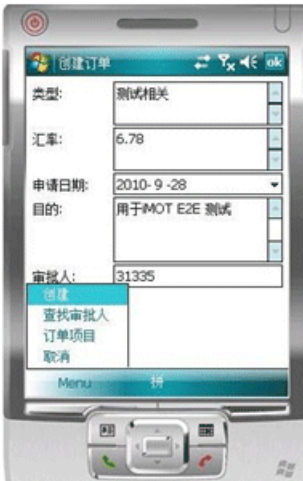
## Internationalization on the Device

---

On the device, e-mail messages and data can include languages other than English.

The internationalization feature depends on the internationalization setting on the device where the Mobile Workflow client running.

E-mail messages can be sent and received using Chinese, for example, which can then be used to extract the parameter. You can also create and update records in using international data, such as Chinese. For example:



### See also

- *Localization Limitations* on page 81
- *Localizing a Mobile Workflow Package* on page 82
- *Mobile Workflow Package Internationalization* on page 91

## 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.

## Sending a Test Notification

# Generating the Files for a Mobile Workflow Package

Use the Mobile Workflow Package Generation wizard to generate the files for the mobile workflow package, optionally deploy the generated package files to the server, and assign the package 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 WorkflowPackage**, 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.
Package Generation and Deployment	
Update generated code	<p>Generate the mobile workflow package and its files. When this option is unselected, the mobile workflow package files are not regenerated, so that modifications made to files that are normally regenerated are not overwritten. This also means, however, that changes made in the Mobile Workflow Forms Editor are not reflected in the generated files.</p> <hr/> <p><b>Note:</b> The <code>manifest.xml</code> and <code>work-flow_package.zip</code> files are generated even if this is not selected.</p>
Generate into the project	Place the generated mobile workflow package and its files in the current project.
Generate to an external folder	Place the generated ZIP file containing the mobile workflow package and its generated files into a location outside of the current project. Click <b>Browse</b> to select the alternate location.
jQuery Mobile theme	Choose a theme for devices that use the jQuery Mobile as the UI framework. The default is theme B. See the jQuery Mobile documentation at <a href="http://jquerymobile.com/">http://jquerymobile.com/</a> for information about the jQuery Mobile themes.

## Generating the Files for a Mobile Workflow Package

Option	Description
Unwired Server profile	Select the Unwired Server profile with which to associate the mobile workflow and extract the user name and password credentials if you are using static authentication.
Deploy to an Unwired Server	Deploy the mobile workflow package to an Unwired Server.
Deploy mode	<p>The deploy mode is automatically set; you cannot change it.</p> <ul style="list-style-type: none"> <li>• New</li> <li>• Replace</li> <li>• Update</li> </ul>
Assign workflow to user(s)	<p>The mobile workflow must be assigned to a device user before the mobile workflow is visible on the user's device. You can assign the same mobile workflow to multiple users. Separate multiple users with a comma. Device users must be registered in Sybase Control Center.</p> <p>Click <b>Get Users</b> to select device users from the list. You must have registered device users in Sybase Control Center to populate this.</p>
Validate controls as soon as the user tries to change focus away from them	<p>If this option is unselected, validation occurs only when the screen is saved. If selected, validation occurs as soon as the control loses focus. If validation fails, a help element appears and shows the error message.</p> <hr/> <p><b>Note:</b> Windows Mobile devices do not support this feature.</p>
Optimize JavaScript in the generated workflow package	<p>The public JavaScript files (API.js, Callbacks.js, Camera.js, and so on) contain the client API functions that you can access for use with your Mobile Workflow package customization. By default, the wizard generates a single JavaScript file (such as SUP0.js, SUP1.js, or SUP2.js), that concatenates these files. Unselect this option if you prefer to use the JavaScript files separately.</p> <hr/> <p><b>Note:</b> If you are deploying to a BlackBerry 7 or later device, selecting this option can make the workflow open more quickly.</p>



### 3. Click **Finish**.

A ZIP file containing the application and its generated files is created and placed in the specified location.

## Deployment Modes

---

These are the deployment modes when you generate the mobile workflow package.

### *New*

The New deployment mode initially generates and deploys the mobile workflow package.

### *Replace*

The Replace deployment mode removes an installed mobile workflow package and installs a new mobile workflow package with the same name and version. The Replace deployment mode acquires a list of assigned devices for the original package, uninstalls the original package, installs the new package with the same name and version, then assigns the original device list to the new package, thus preserving any device assignments associated with the original package.

Use the replace deployment mode for minor changes and updates to the mobile workflow, or during initial development.

### *Update*

The Update deployment mode installs a new mobile workflow package with the original package name and assigns a new, higher version number than the existing installed mobile workflow package. During the update operation, a list of assigned devices is acquired from the original package, the new package is installed and assigned a new version number and then the administrator specifies device assignments for the new package from the acquired list of assigned devices. Existing notifications are preserved.

Use the Update deployment mode for major new changes to the mobile workflow.

## Generated Mobile Workflow Files

---

When you generate mobile workflow package files, some files are generated every time and others are generated only under certain conditions.

These files are generated every time you generate the mobile workflow package:

- `manifest.xml` – describes how the contents of the Mobile Workflow package .zip file are organized.
- `workflow_name.zip` – contains all of the Mobile Workflow files, including the Web application files, look and feel files, the JavaScript files, and so on.

## Generating the Files for a Mobile Workflow Package

These files are regenerated only if you select the **Generate** option in the Mobile Workflow Package Generation wizard:

- *workflow\_name.html* – an HTML file that describes simple workflow screens and forms. Default name: *workflow.html*.
- *workflow\_name\_CustomLookAndFeel.html* – a workflow html file that adds Sybase JavaScript functions and CSS styles.
- *workflow\_name\_jQueryMobileLookAndFeel.html* – a workflow html file that adds jQuery Mobile functions and CSS styles.
- *WorkflowClient.xml* – contains metadata that specifies how to map the data in the workflow message to and from calls to Mobile Business Object (MBO) operations and object queries.
- *workflow\_name.xml* – look and feel file that uses the basic *workflow\_name.html* file.
- *Resources.js* – allows you to access localized string resources.
- *Workflow.js* – contains functions for common menu, screen, and database operations.

These files are generated only if you select the **Generate** option and the files do not exist:

- *API.js* and *Utils.js* – provide Mobile Workflow functions used to communicate with the Hybrid Web container.
- *Custom.js* – enables you to add JavaScript code to customize the Mobile Workflow application. Your file is preserved each time you regenerate the package.
- *WorkflowMessage.js* – provides functions to access Workflow Message resources.
- All *\*.css* files – defines formatting rules to render the screens in HTML.

When you generate the mobile workflow package into the current project, the ZIP file containing the mobile workflow application and its files is placed in the *Generated Workflow* folder in the project, for example, *C:\Documents and Settings\username\workspace\Project\_Name\Generated Workflow*. The files are shown in the Workspace Navigator. This shows the generated file structure for a project named *PurchaseOrder*.

The *Generated Workflow\project\_name* folder contains:

- *html* – this folder includes:
  - *workflow.html* – contains all the screens in the Mobile Workflow application, each in its own `<div>` section. This is used on BlackBerry, Android, and iOS platforms with the **Optimize for performance** look and feel. On Windows Mobile, it is used for all looks-and-feels.
  - *workflow\_customlookandfeel.html* – contains all the screens in the Mobile Workflow application. This is used with the **Optimize for appearance** look and feel on BlackBerry 5.0

- `workflow_jquerymobilelookandfeel.html` – contains all the screens in the Mobile Workflow application. This is used with the **Optimize for appearance** look and feel on iOS, BlackBerry 6.0, and Android.
- `js\Custom.js` – edit this file to customize the Mobile Workflow application. Since you can modify this file, it is generated only once, or when not already present in the generated files. This ensures that it is not overwritten if you subsequently regenerate the Mobile Workflow package. Examples of ways you can customize the Mobile Workflow application include:
  - Manipulating HTML elements.
  - Writing code that is called before or after generated behavior is invoked for menu items.
  - Implementing custom validation logic.
- PhoneGap JavaScript file. Typically named `js\phonegap-version.js`, for any Workflow package that is built for an Android or iOS device using the PhoneGap library. The file is copied from the JavaScript folder where PhoneGap is installed.
- `<project_name>.zip` – contains the mobile workflow application and its files, including the images, user interface, and controls

See *Developer Guide: Mobile Workflow Packages* for more information about generated files and customization.

### **Generated HTML Files**

The Mobile Workflow Forms Editor generates these HTML files.

- `workflow.html` – contains all the screens in the Mobile Workflow application, each in its own `<div>` section. This is used on BlackBerry, Android, and iOS platforms with the **Optimize for performance** look and feel. On Windows Mobile, it is used for all looks-and-feels.
- `workflow_customlookandfeel.html` – contains all the screens in the Mobile Workflow application. This is used with the **Optimize for appearance** look and feel on BlackBerry 5.0
- `workflow_jquerymobilelookandfeel.html` – contains all the screens in the Mobile Workflow application. This is used with the **Optimize for appearance** look and feel on iOS, BlackBerry 6.0, and Android.

---

**Note:** In Preferences, **Optimize for appearance** is the default look and feel.

---

### **Look and Feel Files**

By default, on BlackBerry 6.0, Android and iOS platforms, the jQuery Mobile look and feel is used. On BlackBerry 5.0, a custom look and feel is used as the default.

---

**Note:** In Preferences, **Optimize for appearance** is the default look and feel.

---

CSS files include:

## Generating the Files for a Mobile Workflow Package

- `jquery.mobile-1.0.css` – located in `Generated Workflow \<mobile_workflow_name>\html\css\jquery` folder and used on BlackBerry 6.0, Android, and iOS platforms. By default, pages are generated using the B data theme. Modify the `ui-body-a` class selector in this file to modify the look and feel, for example, the background image or color.
- `master.css` – located in `Generated Workflow \<mobile_workflow_name>\html\css\bb` and used on the BlackBerry 5.0 platform. This is used on the BlackBerry 5.0 platform when the Optimize for appearance preference is selected. Modify the `body` selector to change the look and feel, for example, the background color.
- `stylesheet.css` – located in `Generated Workflow \<mobile_workflow_name>\html\css`. This look and feel is considerably simpler, using no JavaScript code to manipulate the controls, and only a single CSS file. This style sheet is used on all platforms for the Optimize for performance preference is selected. To modify the background color for this look and feel, modify the `body` selector.

### BlackBerry 6.0, Android, and iOS Look and Feel

The default look and feel for BlackBerry 6.0, Android, and iOS is provided by the jQuery Mobile framework.

For this look and feel, the layout of the HTML at a high level is:

- Each screen has a block, contained in a `<div>` element, with a `data-role` of "page" and a `data-theme` of 'a.' Each `<div>` has a `<div>` with a `data-role` of "header," and a child element for the menu. Use the contents of the header `<div>` to manipulate the menu.

```
<div data-role="page" data-theme='a'
id="Department_createScreenDiv">
  <div data-role="header" data-position="inline">
    <a data-icon="arrow-l"
id="Department_createScreenDivCancel" name="Cancel"
onclick="menuItemCallbackDepartment_createCancel();" > Cancel</a>
    <h1>Department_create</h1>
    <a id="Department_createScreenDivCreate" name="Create"
onclick="menuItemCallbackDepartment_createSubmit_Workflow();" >
Create</a>
  </div>
```

- The menu has one anchor, `<a>`, for each menu item:

```
<a id="Department_createScreenDivCreate" name="Create"
onclick="menuItemCallbackDepartment_createSubmit_Workflow();" >
Create</a>
```

- In addition to a menu, each screen `<div>` has a `<div>` with a `data-role` of "content," a child element where the controls are hosted. The content `<div>` has a child `<div>` with a `data-role` of "scroller." This `<div>` in turn has a form with a number of `<div>`s. The "content" `<div>` is where you can do customizations, for example, branding.

```
<div data-role="content" class="wrapper" >
  <div data-role="scroller">
    <form name="Department_createForm"
id="Department_createForm">
      <div class="customTopOfFormStyle" ><span
```

```
id="Department_createForm_help" class="help"></span></div>
  <div class="customTopOfFormStyle"
id="topOfDepartment_createForm"></div>
  <div class="editbox">
    <label class="left"
for="Department_create_dept_name_paramKey">Dept name:</label>
    <input class="right" type="text"
id="Department_create_dept_name_paramKey"/><span
id="Department_create_Department_create_dept_name_paramKey_help"
class="help"></span>
  </div>
```

The first <div> is a block for use to display help, a <span> element.

The next <div> is a built-in element that can be used to find the top of the form. The last <div> is another built-in element that can be used to find the bottom of the form.

If you look into Custom.js file, it is recommended that you add customizations such as branding to the <div> "TopOf" ScreenKey "Form" and "bottomOf" screenKey "Form."

For example:

```
/*
var screenKey = getCurrentScreen();
var form = document.forms[screenKey "Form"];
if (form) {
var topOfFormElem = document.getElementById("topOf" screenKey
"Form");
! topOfFormElem.innerHTML = "Use this screen to ...";
var bottomOfFormElem = document.getElementById("bottomOf"
screenKey "Form");
bottomOfFormElem.innerHTML = "<a href=\"help.html\">Click here to
open help</a>";
}
*/
```

All the other <div>s in the form correspond to the controls put on that screen during design time in the Mobile Workflow Forms editor. You might see, for example, a <div> that holds a label, <label>, and a textbox, <input>. When the page is opened, the controls are enhanced by jQuery Mobile to supply additional functionality for controls like buttons, sliders, text inputs, and combo boxes.

A typical mobile workflow with this look and feel, without extraneous attributes, looks similar to this:

```
<html>
  <body onload="onWorkflowLoad();">
    <div data-role="page" data-theme='a'
id="Department_createScreenDiv">
      <div data-role="header" data-position="inline">
        <a data-icon="arrow-l" id="Department_createScreenDivCancel"
name="Cancel" onclick="menuItemCallbackDepartment_createCancel();">
Cancel</a>
        <h1>Department_create</h1>
        <a id="Department_createScreenDivCreate" name="Create"
onclick="menuItemCallbackDepartment_createSubmit_Workflow();">
Create</a>
      </div>
```

## Generating the Files for a Mobile Workflow Package

```
<div data-role="content" class="wrapper" >
  <div data-role="scroller">
    <form name="Department_createForm"
id="Department_createForm">
      <div class="customTopOfFormStyle" ><span
id="Department_createForm_help" class="help"></span></div>
      <div class="customTopOfFormStyle"
id="topOfDepartment_createForm"></div>
      <div class="editbox">
        <label class="left"
for="Department_create_dept_name_paramKey">Dept name:</label>
        <input class="right" type="text"
id="Department_create_dept_name_paramKey"/><span
id="Department_create_Department_create_dept_name_paramKey_help"
class="help"></span>
        </div>
      <div class="customBottomOfFormStyle"
id="bottomOfDepartment_createForm"></div>
    </form>
  </div>
</div>
</body>
</html>
```

### *BlackBerry 5.0 Look and Feel*

A custom look and feel is used, by default, for BlackBerry 5.0.

Each screen has a block, a <div>. Each <div> has a form, <form>, where the controls are hosted. Each form has a number of divs. The first div has a block put aside for use to display help, a <span> element. The next div is a built-in element that can be used to find the top of the form. The last div is another built-in element that can be used to find the bottom of the form. All the divs in the form correspond to the controls put on that screen in the Mobile Workflow Forms Editor. You might get, for example, a <div> that holds a label, <label>, and a textbox, <input>.

A typical mobile workflow with this look and feel, without extraneous attributes, looks similar to this:

```
<html>
  <body onload="onWorkflowLoad();">
    <div id="Department_createScreenDiv">
      <form name="Department_createForm"
id="Department_createForm">
        <div class="customTopOfFormStyle" ><span
id="Department_createForm_help" class="help"></span></div>
        <div class="customTopOfFormStyle"
id="topOfDepartment_createForm"></div>
        <div class="editbox">
          <label class="left"
for="Department_create_dept_name_paramKey">Dept id:</label>
          <input class="right" type="text"
id="Department_create_dept_name_paramKey"/><span
id="Department_create_Department_create_dept_id_paramKey_help"
```

```
class="help"></span>
  </div>
  </form>
</div>
</body>
</html>
```

### Optimized for Performance Look and Feel

This is a simple look and feel you can use on all platforms.

---

**Note:** Windows Mobile 6.x Professional platforms always use the Optimized for performance look and feel, as this platform is not supported by jQuery Mobile.

---

Choose the **Optimized for performance** option when you configure Mobile Workflow Forms Editor preferences. For this look and feel, the layout of the HTML at a high level is:

- Each screen has a block, a <div> element. Each of those <div> elements has an unordered list element, <ul>, a child element for the menu. The menu has one list item, <li>, for each menu item.
- In addition to a menu, each <div> has a form element, <form>, where the controls are hosted.
- Each form has a single table, <table>, with a number of table rows, <tr>. The first table row has a block to display help, a <span> element. The next table row is a built-in element, a table data or <td>, that can be used to find the top of the form.
- The last table row is another built-in element, a <td>, that can be used to find the bottom of the form.
- All the other rows in the form correspond to the controls put on that screen in the Mobile Workflow Forms editor. You might get, for example, a row with two table datas, the first holding a <label> and the second holding a textbox (<input>).
- A column can have only one width, so if you have more than one line, one column may contain different widths, which means the last width prevails. The contents of a field are wrapped only where there is a space. If there is no space, the contents are not wrapped. As a result, depending on the length of the data, Listviews may not respect the field widths specified in the Mobile Workflow Forms Editor with this look-and-feel.

A typical mobile workflow with this look and feel, without extraneous attributes, looks similar to this:

```
<html>
  <body onload="onWorkflowLoad();" >
    <div id="Department_createScreenDiv">
      <ul id="Department_createScreenDivMenu" class="menu">
        <li><a class="nav" name="Create"
onclick="menuItemCallbackDepartment_createSubmit_Workflow();" >Creat
e</a></li>
          <li><a class="nav" name="Cancel"
onclick="menuItemCallbackDepartment_createCancel();" >Cancel</a></
li>
        </ul>
      <form name="Department_createForm"
```

## Generating the Files for a Mobile Workflow Package

```
id="Department_createForm">
  <table class="screen">
    <tr>
      <td colspan="2"><span id="Department_createForm_help"
class="help"></span></td>
    </tr>
    <tr>
      <td colspan="2" id="topOfDepartment_createForm"></td>
    </tr>
    <tr>
      <td class="left"><label
for="Department_create_dept_name_paramKey">Dept name:</label></td>
      <td class="right"><input class="right" type="text"
id="Department_create_dept_name_paramKey" /><span
id="Department_create_Department_create_dept_name_paramKey_help"
class="help"></span></td>
    </tr>
    <tr><td colspan="2" id="bottomOfDepartment_createForm"></
td></tr></table>
  </form>
</div>
</body>
</html>
```

## Deploying the Mobile Workflow Package to Devices

---

After generating the Mobile Workflow package and its files and deploying it to Unwired Server, you must deploy the Mobile Workflow application to the device or simulator.

See the *Developer Guide for Mobile Workflow Packages* for information about how to configure devices or simulators for the Mobile Workflow Package.

See *Sybase Control Center for Unwired Platform* documentation for information about managing devices, Workflow assignments, and users.

## Activating the Workflow

---

The menu items on a Workflow screen can be either a Submit Workflow (asynchronous) or Online Request (synchronous) menu item type.

To complete the mobile workflow activation process, the last screen in the mobile workflow application must have a Submit Workflow menu item. This is necessary for the device and server-side to activate the mobile workflow for the device.

Mobile workflows are considered to have been processed and/or activated only if they are closed with a Submit Workflow menu item, and which may, or may not, have a corresponding mobile business object (MBO) operation tied to it.



# Adding Custom Code

Use JavaScript code to customize the Mobile Workflow application.

1. Use the Mobile Workflow Package Generation wizard to generate the Mobile Workflow package and its files.

When the Mobile Workflow package is generated, the `Custom.js` file is generated if not already present in the project. The `Custom.js` file is located in `Generated Workflows\<workflow_project_name>\html\js`.

2. Right-click the `Custom.js` file and select the editor with which to open the file.
3. Add your JavaScript code.

You can also add your own separate JavaScript files to `Generated Workflows \<workflow_project_name>\html\js`, then add custom code to the `Custom.js` file that calls the functions in the JavaScript files you added. This prevents the `Custom.js` file from becoming extremely long, which makes it difficult if multiple developers are working on the same Mobile Workflow application simultaneously.

4. Save and close the `Custom.js` file.

Since the `Custom.js` file is generated only if it is not already present in the Mobile Workflow project, this file will not be re-generated if you subsequently re-generate the Mobile Workflow package, so any modifications you make are preserved.

5. Deploy the Mobile Workflow package to Unwired Server.



# 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.



# 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 Guide: Android Object API Applications*
  - *Developer Guide: BlackBerry Object API Applications*
  - *Developer Guide: iOS Object API Applications*
  - *Developer Guide: Windows and Windows Mobile Object API Applications*
  - *Developer Guide: Mobile Workflow Packages*
- **Unwired Server API** – custom Java classes used to implement advanced data handling features in Unwired Server. See *Mobile Data Models: Using Mobile Business Objects* 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 *Developer Guide for Unwired Server Management API* and the Javadocs included in the installation directory.

Check the Sybase Product Documentation Web site regularly for updates: <http://sybooks.sybase.com/sybooks/sybooks.xhtml>, then navigate to the most current version.



# 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**<sup>®</sup> – A relational database management system that provides the messaging database for Sybase Unwired Platform. *See* messaging database.

**Afaria**<sup>®</sup> – 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 is a separately licensed product that can extend the Unwired Platform in a mobile enterprise. Afaria includes a server (Afaria Server), a database (Afaria Database), an administration tool (Afaria Administrator), and other runtime components, depending on the license you purchase.

**application** – In Unwired Server (and visible in Sybase Control Center), an application is the runtime entity that can be directly correlated to a native or mobile workflow application. The application definition on the server establishes the relationship among packages used in the application, domain that the application is deployed to, user activation method for the application, and other application specific settings.

**APNS** – Apple Push Notification Service.

**application connection** – A unique connection to the application on a device.

**application connection template** – a template for application connections that includes application settings, security configuration, domain details, and so forth.

**application node** – In Sybase Control Center, this is a registered application with a unique ID. This is the main entity that defines the behavior of device and backend interactions.

**application registration** – The process of registering an application with Sybase Unwired Platform. Registration requires a unique identity that defines the properties for the device and backend interaction with Unwired Server.

**artifacts** – Artifacts can be client-side or automatically generated files; for example: .xml, .cs, .java, .cab files.

**availability** – Indicates that a resource is accessible and responsive.

**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 Unwired Server cache database that store synchronization data. *See* cache database.

**cache group** – Defined in Unwired WorkSpace, MBOs are grouped and the same cache refresh policy is applied to their virtual tables (cache) in the cache database

**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 argument used by the operation to load data into the cache from the enterprise information system (EIS).

**cache database** – Cache database. The Unwired Server cache database 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 Guide: BlackBerry Native Applications*, *Developer Guide: iOS Native Applications*, and *Developer Guide: Windows and Windows Mobile Native Applications*.

**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 cache, messaging, and monitoring databases.



**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.

For proxy connections, a connection pool is a collection of proxy connections pooled for their respective back-ends, such as SAP Gateway.

**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 cache database through a push event.

**data refresh** – A data refresh synchronizes data between the cache 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 cache database (CDB), cluster, monitoring, and messaging databases.

**data vault** – A secure store across the platform that is provided by an SUP client.

**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 notification** – Replication synchronization 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 replication 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, 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 requirement 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, 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** – A synchronization method where data is delivered asynchronously using a secure, reliable messaging protocol. This method 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* 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 cache, 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 messaging 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 synchronization, messaging 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 cache 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.

**onboarding** – The enterprise-level activation of an authentic device, a user, and an application entity as a combination, in Unwired Server.

**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/argument) 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 arguments – perform a data refresh based on the value of the argument.
- 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: Transient, 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.

**perspective** – A named tab in Sybase Control Center that contains a collection of managed resources (such as servers) and a set of views associated with those resources. The views in a perspective are chosen by users of the perspective. You can create as many perspectives as you need and customize them to monitor and manage your resources.

Perspectives allow you to group resources ways that make sense in your environment—by location, department, or project, for example.

**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 cache database. On Windows Mobile, pull synchronization is supported only in replication 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.

**resource** – A unique Sybase product component (such as a server) or a subcomponent.

**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 Unwired Server 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.

**schedule** – The definition of a task (such as the collection of a set of statistics) and the time interval at which the task must execute in Sybase Control Center.

**scheduled data refresh** – Data is updated in the cache 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 its 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.

**SSO** – Single sign-on. SSO is a credential-based authentication mechanism.

**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, as well as monitor the environment. You can also deploy and MBO or workflow packages, as well as register applications and define templates



for them. 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 Control Center X.X Service** – Provides runtime services to manage, monitor, and control distributed Sybase resources. The service must be running for Sybase Control Center to run. Previously called Sybase Unified Agent.

**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 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** – A synchronization method where data is delivered synchronously using an upload/download pattern. For push-enabled clients, synchronization uses a "poke-pull" 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.

**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 Unwired Server cache database) or download event (from the cache 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.

**view** – A window in a perspective that displays information about one or more managed resources. Some views also let you interact with managed resources or with Sybase Control Center itself. For example, the Perspective Resources view lists all the resources managed by the current perspective. Other views allow you to configure alerts, view the topology of a replication environment, and graph performance statistics.

**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

## A

- Activate starting point 29
- adding images 8
- adding menu items, custom actions, and controls to the screen 39
- Async Request Error connection 29
- AttachmentViewer 59, 62
  - configuring 61
  - limitations 62

## B

- BlackBerry 5.0 104
- button
  - properties 64

## C

- certificate picker 22
- checkbox properties 43
- choice control 50
- choice item
  - adding 49
  - deleting 50
  - editing 50
- Client-initiated starting point 29
- complex data type 39
- conditional start 18
- configuring control properties 43
- connection properties 32
- connections 32
- Connections,
  - Async Request Error 29
  - Goto 29
  - Listview details 29
  - Operation error 29
  - Operation success 29
- content type preference, changing 83
- context menus 9
- controls 39
  - adding 39
  - button 64
  - listview 55
  - listview properties 58

- properties 55
- creating keys 24
- Credential request starting point 29
- Credentials
  - dynamic 20
  - static 20
- custom actions
  - adding 39, 69
  - configuring 79
  - properties 69
  - types 69
- Custom.js 107

## D

- data source 7
- default locale, creating 84
- deploy mode 87
- deploying
  - mobile application project 87
- deployment mode
  - replace 99
  - update 99
- Dynamic authentication 22

## E

- editing
  - locale properties file 87
- Empty screen 29
- encoding type
  - changing 83
  - default 83
  - ISO-8859-1 83
  - non-ASCII 83
  - UTF-8 83

## F

- file associations 83
- Flow Design
  - general properties 25
  - localization 25
  - matching rules 25
  - page properties 25

## Index

rulers and grid 25

## G

generate old key values 77

generated files 99

    workflow\_customlookandfeel.html 101

    workflow\_jquerymobilelookandfeel.html 101

    workflow.html 101

glossaries

    Sybase Unwired Platform terms 113

GoTo connection 29

## I

image 62, 66

images

    adding 8

internationalization

    Mobile Workflow Forms editor 91

    on the device 93

## J

jquery.mobile-1.0.css 101

## K

key collection menu item 77

Keys 25

keys, creating 24

## L

label 65

link 65

listview 55, 58

    editing fields 57

listview control 39

Listview Details and Add connections 29

listview, generated keys 58

locale

    editing 87

    properties file 87

locale properties file

    creating 84

    validating 84

localization 81

    creating a new locale 84

    device settings 88

    limitations 81

    Mobile Workflow package 82

    task flow 82

Localization 25

    current locale 91

    updating the current locale 91

look and feel files 101

## M

manually adding a starting point 11

master.css 101

matching rules, adding 18

menu item

    cancel 71

    close 71

    open 71

    properties 69

    save 71

    types 69

menu item, creating 71

menu items

    adding 39, 69

    custom 79

    online request 74

message-based application 87

mobile application package 7

mobile application project, deploying 87

mobile business object (MBO) 7

mobile business object relationships 39

Mobile Workflow Designer

    button 36

    checkbox 36

    choice 36

    editbox 36

    htmlview 36

    listview 36

    menu item 36

    signature 36

    slider 36

Mobile Workflow form, designing 1

Mobile Workflow Forms editor 1

Mobile Workflow Forms Editor 3

    Flow Design palette options 29

    preferences 4

    preferences, Mobile Workflow Forms Editor

        4

    setting preferences 3

Mobile Workflow Generation wizard 97

mobile workflow package  
generated files 99

## N

notification extraction rules 17  
notification extraction rules, editing 17  
Notification Mailbox 95  
notification, matching rules 18  
notification, sending a test 95

## O

online request menu item 74  
Operation Error connection 29  
Operation Success connection 29  
optimized for performance 105

## P

parameter mappings 78  
preferences  
appearance 83  
content types 83  
general 83  
properties  
button control 64  
image 66  
label 65  
link 65  
server-initiated 14  
Properties  
Flow Design page 25  
PurchaseOrderSample 84

## R

regional settings 88

## S

Screen Design  
page properties 34  
Screen Design palette options  
checkbox 36  
choice 36  
editbox 36  
htmlview 36

listview 36  
menu item 36  
signature 36  
slider 36  
Screen, empty 29  
screens  
adding 7  
sending a test notification 95  
server-initiated properties 14  
server-initiated starting point 12  
Server-initiated starting point 29  
settings  
device 88  
iPhone 88  
regional 88  
Windows Mobile 88  
signature control 53  
starting point  
adding manually 11  
starting points  
Activate 10  
Client-initiated 10  
Credential request 10  
editing 18  
server-initiated 12, 18  
Server-initiated 10  
Starting points  
Activate 29  
Client-initiated 29  
Credential request 29  
Server-initiated 29  
static authentication 21  
stylesheet.css 101  
submit menu item 72  
supported attachment types 62  
Sybase Mobile Workflow Forms Editor  
accessing 6  
creating a new 6

## T

terms  
Sybase Unwired Platform 113  
testing a notification 95

## U

UTF-8 encoding 83

**W**

workflow.html 102

workflow.html file 101