



**Developer Reference for Mobile Workflow  
Packages**

---

**Sybase Unwired Platform 1.5.5**

DOCUMENT ID: DC01218-01-0155-01

LAST REVISED: December 2010

Copyright © 2010 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.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. 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 Sun Microsystems, Inc. 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>Introduction to Developer Reference for Mobile</b> |           |
| <b>Workflow Package</b> .....                         | <b>1</b>  |
| Documentation Road Map for Unwired Platform .....     | 1         |
| <b>Mobile Workflow Development</b> .....              | <b>5</b>  |
| Architectural Overview .....                          | 5         |
| Mobile Workflow Package Overview .....                | 6         |
| <b>Mobile Workflow Development Life Cycle</b> .....   | <b>9</b>  |
| Server-Initiated Versus Client-Initiated Mobile       |           |
| Workflows .....                                       | 9         |
| Identify a Business Process for Workflow              |           |
| Development .....                                     | 10        |
| Define Mobile Business Objects .....                  | 10        |
| Mobile Business Objects .....                         | 10        |
| MBO Deployment Requirements .....                     | 11        |
| Performing Server-side Development .....              | 11        |
| <b>Mobile Workflow Data Change Notification</b> ..... | <b>17</b> |
| Mobile Workflow Data Change Notification              |           |
| Development Life Cycle .....                          | 19        |
| Mobile Workflow DCN Request using HTTP                |           |
| Authentication .....                                  | 20        |
| Non HTTP Authentication Workflow DCN Request .....    | 21        |
| Implementing the Data Change Notification Filter      |           |
| Class .....   | 21        |
| Mobile Workflow DCN Request Response .....            | 21        |
| <b>Develop a Mobile Workflow Package</b> .....        | <b>23</b> |
| The Mobile Workflow Forms Editor .....                | 23        |
| Creating a New Mobile Workflow Form .....             | 24        |
| Flow Design Page Properties .....                     | 25        |
| Flow Design Palette Options .....                     | 29        |
| Screen Design Page Properties .....                   | 30        |
| Screen Design Palette Options .....                   | 31        |

|  |    |
|--|----|
| Mobile Workflow Forms Editor Context Menus ....                                | 34 |
| Sending a Test Notification .....  | 35 |
| Starting Points .....  | 35 |
| Adding a Starting Point Manually .....   | 36 |
| Creating a Server-initiated Starting Point .....                               | 37 |
| Credentials .....  | 42 |
| Creating Keys .....  | 44 |
| Configuring a Mobile Workflow Package for Online                               |    |
| Mobile Business Object Data Access .....                                       | 46 |
| Binding the findByParameter Object Query to a<br>Menu Action .....             | 46 |
| Binding Transient Personalization Keys to<br>Mobile Workflow Keys .....        | 48 |
| Mobile Business Object Relationships .....                                     | 49 |
| Creating Screens Automatically by Dragging and<br>Dropping a Data Source ..... | 50 |
| Adding Controls to the Screen .....  | 50 |
| Mobile Workflow Controls .....   | 51 |
| Adding a Menu Item .....   | 51 |
| Configuring Control Properties .....   | 60 |
| Checkbox Properties .....  | 60 |
| Editbox Properties .....   | 61 |
| Choice Properties .....  | 63 |
| Slider Properties .....  | 66 |
| Signature Properties .....   | 67 |
| HtmlView Properties .....  | 68 |
| Listview Properties .....  | 69 |
| Adding Connections to Screens .....  | 71 |
| Connections .....  | 72 |
| Connection Properties .....  | 72 |
| Activating the Workflow .....  | 73 |
| Localization .....   | 73 |
| Changing the Encoding Type .....   | 74 |
| Creating a New Locale .....  | 75 |
| Editing a Locale Properties File .....   | 76 |

|  |            |
|--|------------|
| Updating the Current Locale .....  | 76         |
| Removing a Locale .....  | 77         |
| Loading a Locale Properties File .....   | 77         |
| Validating the Locale Properties Files .....   | 77         |
| Generating the Files For a Mobile Workflow Package ..  | 77         |
| <b>Preparing Devices .....</b>   | <b>81</b>  |
| Preparing iPhone Devices for the Workflow Package ...  | 81         |
| Installing the Mobile Workflow Package on the<br>Device Using the Mobile Workflow Container<br>Downloaded From App Store ..... | 81         |
| Deploying the Mobile Workflow Package to the<br>iPhone Simulator .....   | 82         |
| Preparing Windows Mobile Devices for the Mobile<br>Workflow Package .....  | 84         |
| <b>Launching a Mobile Workflow on the Device .....</b>   | <b>89</b>  |
| <b>Testing Mobile Workflow Packages .....</b>  | <b>91</b>  |
| Testing Server Initiated Mobile Workflow Packages .....  | 91         |
| <b>Configuring the Mobile Workflow Package in Sybase<br/>    Control Center .....</b>  | <b>93</b>  |
| Deploying a Mobile Workflow Package .....  | 93         |
| Registering and Reregistering Messaging Devices .....  | 93         |
| Assigning and Unassigning Device Users from the<br>Workflow Node .....   | 94         |
| Enabling and Configuring the Notification Mailbox .....  | 95         |
| <b>Troubleshoot .....</b>  | <b>97</b>  |
| HTTP Error Codes .....   | 97         |
| Recovering from EIS Errors .....   | 98         |
| Mapping of EIS Codes to Logical HTTP Error Codes ...   | 99         |
| Credentials Are Lost after User Successfully Passes<br>Activation Screen .....   | 100        |
| Mobile Workflow Exception Handling .....   | 100        |
| <b>Index .....</b>   | <b>103</b> |



# Introduction to Developer Reference for Mobile Workflow Package

This developer reference provides information about using Sybase® Unwired Platform features to create Mobile Workflow packages. The audience is Mobile Workflow developers.

This guide describes requirements for developing a Mobile Workflow package, how to generate Mobile Workflow package code, and how to deploy the Mobile Workflow package to the device or simulator.

Companion guides include:

- *Sybase Unwired WorkSpace – Mobile Business Object*
- *Sybase Unwired WorkSpace – Device Application Development*
- *Tutorial: Mobile Workflow Package Development*
- *Troubleshooting for Sybase Unwired Platform*

## Documentation Road Map for Unwired Platform

---

Learn more about Sybase® Unwired Platform documentation.

**Table 1. Unwired Platform documentation**

| Document  | Description   |
|---|---|
| <i>Sybase Unwired Platform Installation Guide</i> | <p>Describes how to install or upgrade Sybase Unwired Platform. Check the <i>Sybase Unwired Platform Release Bulletin</i> for additional information and corrections.</p> <p>Audience: IT installation team, training team, system administrators involved in planning, and any user installing the system.</p> <p>Use: during the planning and installation phase.</p> |
| <i>Sybase Unwired Platform Release Bulletin</i>   | <p>Provides information about known issues, and updates. The document is updated periodically.</p> <p>Audience: IT installation team, training team, system administrators involved in planning, and any user who needs up-to-date information.</p> <p>Use: during the planning and installation phase, and throughout the product life cycle.</p>                      |

| Document   | Description   |
|--|---|
| <i>New Features</i>                                      | <p>Describes new or updated features.</p> <p>Audience: all users.</p> <p>Use: any time to learn what is available.</p>  |
| <i>Fundamentals</i>                                      | <p>Describes basic mobility concepts and how Sybase Unwired Platform enables you design mobility solutions.</p> <p>Audience: all users.</p> <p>Use: during the planning and installation phase, or any time for reference.</p>  |
| <i>System Administration</i>                             | <p>Describes how to plan, configure, manage, and monitor Sybase Unwired Platform. Use with the <i>Sybase Control Center for Sybase Unwired Platform</i> online documentation.</p> <p>Audience: installation team, test team, system administrators responsible for managing and monitoring Sybase Unwired Platform, and for provisioning device clients.</p> <p>Use: during the installation phase, implementation phase, and for ongoing operation, maintenance, and administration of Sybase Unwired Platform.</p>  |
| <i>Sybase Control Center for Sybase Unwired Platform</i> | <p>Describes how to use the Sybase Control Center administration console to configure, manage and monitor Sybase Unwired Platform. The online documentation is available when you launch the console (<b>Start &gt; Sybase &gt; Sybase Control Center</b>, and select the question mark symbol in the top right quadrant of the screen).</p> <p>Audience: system administrators responsible for managing and monitoring Sybase Unwired Platform, and system administrators responsible for provisioning device clients.</p> <p>Use: for ongoing operation, administration, and maintenance of the system.</p> |



| Document   | Description  |
|--|--|
| <i>Troubleshooting</i>   | <p>Provides information for troubleshooting, solving, or reporting problems.</p> <p>Audience: IT staff responsible for keeping Sybase Unwired Platform running, developers, and system administrators.</p> <p>Use: during installation and implementation, development and deployment, and ongoing maintenance.</p>  |
| Getting started tutorials  | <p>Tutorials for trying out basic development functionality.</p> <p>Audience: new developers, or any interested user.</p> <p>Use: after installation.</p> <ul style="list-style-type: none"> <li>• Learn mobile business object (MBO) basics, and create a mobile device application: <ul style="list-style-type: none"> <li>• <i>Tutorial: Mobile Business Object Development</i></li> <li>• <i>Tutorial: BlackBerry Application Development using Device Application Designer</i></li> <li>• <i>Tutorial: Windows Mobile Device Application Development using Device Application Designer</i></li> </ul> </li> <li>• Create native mobile device applications: <ul style="list-style-type: none"> <li>• <i>Tutorial: BlackBerry Application Development using Custom Development</i></li> <li>• <i>Tutorial: iPhone Application Development using Custom Development</i></li> <li>• <i>Tutorial: Windows Mobile Application Development using Custom Development</i></li> </ul> </li> <li>• Create a mobile workflow package: <ul style="list-style-type: none"> <li>• <i>Tutorial: Mobile Workflow Package Development</i></li> </ul> </li> </ul> |
| <i>Sybase Unwired WorkSpace – Mobile Business Object Development</i> | <p>Online help for developing MBOs.</p> <p>Audience: new and experienced developers.</p> <p>Use: after system installation.</p>  |

| Document   | Description   |
|--|---|
| <p><i>Sybase Unwired WorkSpace – Device Application Development</i></p>  | <p>Online help for developing device applications.<br/>Audience: new and experienced developers.<br/>Use: after system installation.</p>  |
| <p>Developer references for device application customization</p>   | <p>Information for client-side custom coding using the Client Object API.<br/>Audience: experienced developers.<br/>Use: to custom code client-side applications.</p> <ul style="list-style-type: none"> <li>• <i>Developer Reference for BlackBerry</i></li> <li>• <i>Developer Reference for iOS</i></li> <li>• <i>Developer Reference for Mobile Workflow Packages</i></li> <li>• <i>Developer Reference for Windows and Windows Mobile</i></li> </ul> |
| <p>Developer reference for Unwired Server side customization – <i>Reference: Custom Development for Unwired Server</i></p> | <p>Information for custom coding using the Server API.<br/>Audience: experienced developers.<br/>Use: to customize and automate server-side implementations for device applications, and administration, such as data handling.<br/>Dependencies: Use with <i>Fundamentals</i> and <i>Sybase Unwired WorkSpace – Mobile Business Object Development</i>.</p>  |
| <p>Developer reference for system administration customization – <i>Reference: Administration APIs</i></p>                 | <p>Information for custom coding using administration APIs.<br/>Audience: experienced developers.<br/>Use: to customize and automate administration at a coding level.<br/>Dependencies: Use with <i>Fundamentals</i> and <i>System Administration</i>.</p>   |

# Mobile Workflow Development

Mobile workflow forms allow the user to update, delete, or create data even when they are not connected to the network. When they are connected, the data is automatically sent and received.

When designing mobile workflow forms using the Mobile Workflow Application Designer, the data source for the application is supplied by mobile business objects that are designated as messaging mobile business objects when deployed to the server. Using the Mobile Workflow Application Designer allows you to develop mobile workflow screens that can call on the create, update, and delete operations, as well as object queries, of a mobile business object.

The Mobile Workflow package is a lightweight form container consisting of XML files that reference an MBO package, an MBO in that package, and the operation or object query to call along with a mapping of which key values map to parameter values. This enables the creation or modification of MBO data on multiple different device platforms without requiring custom development. Workflow packages are .zip files, which can easily be moved between installations.

On Windows Mobile platforms, Mobile Workflow packages are integrated with the Outlook Mail inbox. On iOS, messages are sent to a container that is provided by the Workflow device client.

Mobile Workflow packages are deployed through the Mobile Workflow Forms Editor and managed through the Sybase Control Center console—no device interaction is required from the administrator. Once a mobile workflow package is deployed into an existing installation, the administrator can configure the mobile workflow package and assign it to any active user in the system.

## Architectural Overview

---

Sybase Unwired Platform is a solution architecture that includes multiple components:

- **Sybase Unwired Server:** Sybase Unwired Server allows users to package and deploy mobile applications. The runtime server handles enterprise data source and application access, communication between the back-end data source and the mobile device, security, transaction processing, and scheduling.
- **Sybase Control Center:** Sybase Control Center (SCC) for Unwired Server is a Web-based console used to configure and manage Unwired Server and its components. Because it is a Web console, multiple administrators can access different components from a single access point.
- **Sybase Unwired WorkSpace:** Sybase Unwired WorkSpace is a development environment designed specifically for mobile application development, testing, and deployment to

assist developers in rapidly moving from development to deployment. You can use Unwired WorkSpace to mobilize data, applications, services, and processes, with support for both online and offline access. Unwired WorkSpace works like a plug-in to your Eclipse development environment, providing the tools you need to create mobile applications.

Experienced developers will already be familiar with WorkSpace's consistent use of Windows, Explorers, Views, Wizards, and Preferences specific to their development environment. Developers new to Eclipse can use online help cheat sheets (Eclipse edition only), samples, and tutorials to quickly become familiar.

## Mobile Workflow Package Overview

---

The basic components of a Mobile Workflow package include starting points, screens and navigations, controls and keys, and menu items.

### *Starting points*

There are four kinds of starting points:

- Client-initiated – initiated manually by the user when he or she opens a Mobile Workflow form from the Workflow menu on the device.
- Server-initiated – initiated when the user opens an e-mail message on the device in which the field contents of the message meet the conditions specified in the Mobile Workflow packages' matching rules.
- Credential Request – sent by the server when it needs authentication information for Unwired Platform.
- Activate – initiated when the Workflow package is used for the first time on the user's device.

### *Screens and navigations*

Navigations represent transitions from a starting point to a screen, or from one screen to another.

New screens are opened when the user clicks a menu item or a row in a Listview control. Screens can also be opened upon success or failure of an operation.

Screens series are designed from a starting point. Screens appear one at a time on the device. Screens are opened on a stack, and closed in the reverse order in which they were added. Each screen contains one or more controls.

### *Controls and keys*

There are seven types of controls—Checkbox, Editbox, Choice, Slider, Signature, Htmlview and Listview.

Htmlview and Listview controls, if present on a screen, must be the only control on that screen.

Editbox controls have a logical type—Text, Numeric, Date, DateTime, or Time. Depending on the logical type, the control at runtime may behave differently, for example, for Date, a datepicker control is available.

The contents of controls are bound to keys. When the screen is opened, the key value determines what appears in the control. When the control's contents are updated, key value changes accordingly.

For server-initiated starting points, you can specify where the value associated with a given key comes from, for example, as an MBO attribute.

When a menu item is bound to an MBO operation or object query, each parameter of the operation or object query is bound to a key. When the operation or object query is invoked, the value passed into each parameter is the value associated with the corresponding key.

### *Menu items*

There are seven types of menu items:

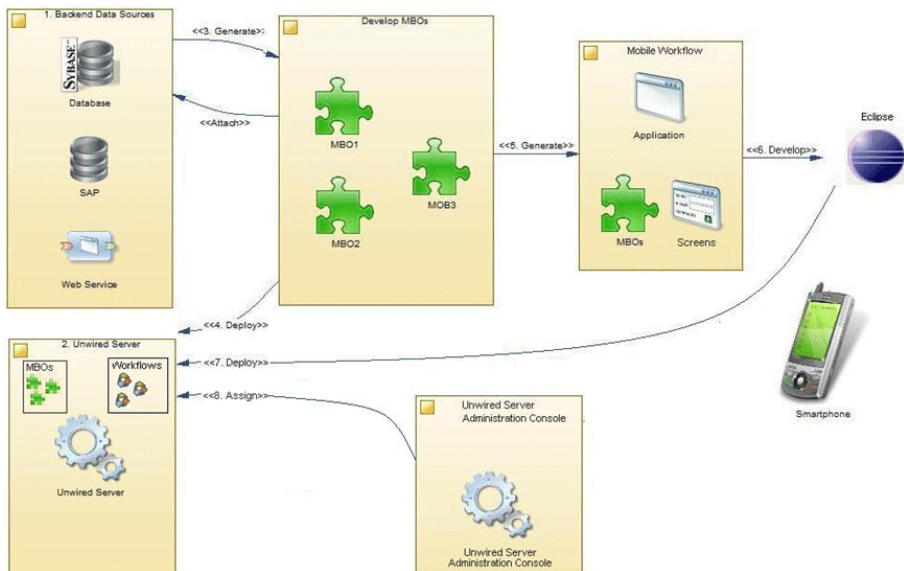
- Save – closes the current screen, validates the current input and, if validation succeeds, saves it.
- Cancel – discards any changes on the current screen and closes it. No validation occurs.
- Close – discards any changes on any open screens and closes all open screens. No validation occurs.
- Open – opens a different screen.
- Add – opens a different screen, where you can add a new row to a listview.
- Submit – validates the entered data on open screens and sends it to the server. There, the server invokes the specified MBO operation or object query. If the Submit menu item has an asynchronous communication style, the client application closes once the request is sent. If the Submit menu item has a synchronous communication style, the client application stays open.
- Add/Update/Delete Listview Row – specifies the MBO operation that is associated with the appropriate action on the Listview. 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. It 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.



# Mobile Workflow Development Life Cycle

Developing a Mobile Workflow package includes these basic tasks.

1. Open or import a mobile application project with predefined mobile business objects (MBOs).
2. Connect to Unwired Server.
3. Use the Mobile Workflow Forms editor to create a new Mobile Workflow package.
4. Generate screens by dragging and dropping MBOs and MBO operations from WorkSpace Navigator to the Flow Design page.
5. Manually create, delete, and edit screens, controls, menus, screen navigations, and so on.
6. Generate the Mobile Workflow package and deploy it to Unwired Server.
7. Log in to Sybase Control Center and assign the Mobile Workflow package to the device user.
8. On the device, run, test and debug the Mobile Workflow package.



## Server-Initiated Versus Client-Initiated Mobile Workflows

The server-initiated and client-initiated Mobile Workflow models can coexist within one Mobile Workflow package.

A server-initiated Mobile Workflow is always initiated by a notification. Unwired Server receives notifications from a Notification Mailbox account and, in turn, delivers a notification

message to an individual device based on configured distribution and matching rules. The matching rules are used to indicate whether a notification should be sent to a particular Mobile Workflow or not. Once a Mobile Workflow is sent to a user, the extraction rules are used to pull values out of the notification to be used as parameter values.

With the client, or device-initiated Mobile Workflow, the user can open the workflow form on his or her mobile device and create a mobile business object (MBO), which is then replayed against the linked server MBO package. You can also run object queries to retrieve existing MBOs, at which point you can then perform update and delete operations on those instances.

## **Identify a Business Process for Workflow Development**

---

The first step is identifying whether a workflow package can implement a decision point in a particular business process.

Workflow packages enable a decision step or triggering of a business process, essentially mobilizing a small decision window in a business process. While some business processes require a thick application with business logic and access to reference data, some others do not. Sometimes a business process can be made mobile simply by providing the ability to capture a single "Yes" or "No" from a user, or by providing the ability to send data in structured form into the existing backend systems.

A typical Workflow package allows creating a mobile business object (MBO) and sending it to the Unwired Server, or retrieving an MBO from the Unwired Server and displaying that information in a decision step. A more complex Workflow package could involve an application that uses synchronous Submit menu items to invoke various create, update, or delete operations and/or object queries all in the same flow. Keep in mind that with workflow packages, there is no local data persistence.

An example of a business process that would be a suitable mobile workflow would be the ability of an employee to use a mobile device to submit an expense report while out of office, or to report on their project activities.

## **Define Mobile Business Objects**

---

Define and develop the mobile business objects to use in the Mobile Workflow package.

### **Mobile Business Objects**

Mobile business objects help form the business logic for mobile applications.

A mobile business object (MBO) is derived from a data source (such as a database server, Web service, or SAP® server). MBOs are deployed to Unwired Server, and accessed from mobile device application clients. MBOs include:



- Implementation-level details – metadata columns that include information about the data from a data source.
- Abstract-level details – attributes that correspond to instance-level properties of a programmable object in the mobile client, and map to data source output columns. Parameters correspond to synchronization parameters on the mobile client, and map to data source arguments. For example, output of a SQL SELECT query are mapped as attributes, and the arguments in the WHERE clause are mapped as synchronization parameters, so that the client can pass input to the query.  
MBO operations include parameters that map to data source input arguments. Operation parameters determine information a client passes to the enterprise information system (EIS).
- Relationships – defined between MBOs by linking attributes and parameters in one MBO, to attributes and parameters in another MBO.

You can define MBOs using either a top-down approach—first designing attributes and parameters, then binding them to a data source; or a bottom-up approach—first specifying a data source, then automatically generating attributes and parameters from it.

A mobile application package includes MBOs, roles, and data source connection mappings, and other artifacts that are delivered to the Unwired Server during package deployment.

### **MBO Deployment Requirements**

When developing a workflow package, the mobile business object must be deployed as a messaging MBO in a Mobile Application project to the Unwired Server.

For more information on the message-based synchronization model, refer to the *Sybase Unwired Platform Fundamentals Guide*.

### **Performing Server-side Development**

This describes creating the server-side code for a mobile application, including creating mobile business objects and binding to a data source.

You can follow similar workflows to create mobile applications using a variety of data sources (like a Web service, database, SAP®, Remedy, or document).

1. Create a data source connection profile. A connection profile contains the connection property information needed to connect to a server in your enterprise.
2. Create a new Sybase Unwired Server connection profile to supply runtime connection information.
3. Create a mobile application project. A mobile application project is the container for the mobile business objects that forms the business logic of mobile applications.
4. Create a mobile business object (MBO), define the attributes and operations, and bind directly to the data source by dragging and dropping the data source onto the Mobile Application Diagram.

5. Deploy a Mobile Application project directly to an Unwired Server, and optionally create a reusable deployment profile.

After completing these procedures, the server-side development is complete. You can continue development by creating the workflow package for the client device as described in *Develop a Mobile Workflow Package*.

### **Creating a Data Source Connection Profile**

A connection profile contains the connection property information needed to connect to a server in your enterprise.

1. In the Enterprise Explorer, right-click one of the following connection categories and select **New**.
  - Database Connections
  - SAP Servers
  - Web Services
  - Rest Web Services
2. Follow the instructions on the wizard pages to create the selected connection.

### **Creating a Sybase Unwired Server Connection Profile**

Create a new Sybase Unwired Server connection profile to supply runtime connection information.

1. In Enterprise Explorer, right-click **Unwired Servers** and select **New**.
2. Enter a name and optional description of the connection profile and click **Next**.
3. Complete the following information on the wizard pages:

**Table 2. Specify Connection Details page**

| <b>Field</b>  | <b>Description</b>  |
|---------------|---|
| Host          | The host name on which the server resides. The default is the machine name.         |
| IP domain     | (Optional) The domain name. The default is the Windows domain of the local machine. |
| Port          | The port to which you are connecting. The default port is 2000, use 2001 for HTTPS. |
| User Name     | Enter the name for the server login. The default user name is supAdmin.             |
| Password      | Enter the password for the server login, if required. For example, s3pAdmin.        |
| Save Password | Select the checkbox to save the password.   |

| Field               | Description  |
|---------------------|--|
| Secure              | <p>Select the checkbox to enable a secure connection with the server, which is different than a secure connection session.</p> <p>To establish a secure session with Unwired Server:</p> <ol style="list-style-type: none"> <li>1. Enable <b>Secure management port</b> on Unwired Server's administration listener. See the Sybase Control Center help topic for information.</li> <li>2. Enter 2001 as the <b>Port</b>.</li> <li>3. Add the certificate file path to the &lt;Unwired_Platform_Install_Dir&gt;\Eclipse\UnwiredWorkspace.bat file.</li> </ol> <hr/> <p><b>Note:</b> Unless the Unwired Server administration listener certificate is signed by a trusted certificate authority, or the certificate is imported into the %JAVA_HOME%\jre\lib\cacerts trust store, you see an error message when testing the Unwired Server connection: "unable to find valid certification path to requested target".</p> |
| Cluster information | <p>The read-only properties for this Server's cluster displays Relay Server information, or, if no relay server is set, displays non-cluster Unwired Server connection information.</p> <p>Relay Server connection information includes host, port, and URL suffix for messaging-based and replication-based synchronization.</p>  |

4. Click **Next** to review a summary of this connection profile, or **Finish** to create the connection profile. The connection profile displays under Unwired Servers in the Enterprise Explorer.

### **Creating a Mobile Application Project**

A mobile application project is the container for the mobile business objects that forms the business logic of mobile applications.

You must create a mobile application project before you can create mobile business objects.

1. Select **File > New > Mobile Application Project** from the main menu bar.
2. Enter a:

- Name
- Location (if other than the default).

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

An empty Mobile Application Diagram opens.

### **Creating a Mobile Business Object by Dragging and Dropping a Data Source**

Create a mobile business object (MBO), define the attributes and operations, and bind directly to the data source by dragging and dropping the data source onto the Mobile Application Diagram.

### **Deploying a Mobile Application Project**

Deploy a Mobile Application project directly to an Unwired Server, and optionally create a reusable deployment profile.

### **Prerequisites**

A connection profile to at least one Unwired Server must be available before you deploy a Mobile Application project to a server.

### **Task**

To avoid errors or inconsistent behavior, client applications must be regenerated whenever a package has been redeployed. Restarting the client application is not sufficient to reset the client for a package that has been redeployed.

#### 1. Right-click the Mobile Application project and select **Deploy Project**.

Alternatively, you can launch the deployment wizard, which automatically sets the Unwired Server portion of the wizard, by dragging a Mobile Application project folder from Workspace Navigator and dropping it on the Unwired Server in Enterprise Explorer to which you are deploying.

---

**Note:** As an option, you can press F9 when your cursor is in the Mobile Application Diagram to launch the Deployment wizard for the corresponding project. If a deployment profile exists for the project, F9 performs quick deployment of the project according to the profile.

---

2. Enter the deployment mode, target version, Package namespace, and click **Next**.
3. Select the contents to be deployed and click **Next**.
4. Create or add required JAR files for MBOs that use Resultset Filter and click **Next**.
5. Select a target server for the project and click **Next**.

If a server connection has been open a long time without activity (a day or two for example), the connection may time out without your knowledge. While you may still be able to select the target server, deployment fails. Verify you have a working connection by

pinging the server. If necessary, reconnect to the Unwired Server to which you are deploying from Enterprise Explorer.

6. Map connection profile to server connections – You must map design-time connection profiles to server-side (run-time) enterprise information system (EIS) data sources referenced by the MBOs in the project. Deployment fails if the EIS data sources are not running and available to connect to. To map the connection profile to a server connection, select the connection profile from the list of available connection profiles then select the corresponding server connection to which it maps.

Contact the system administrator in cases where your development environment permits access to systems that the Unwired Server prohibits.

---

**Note:** You can also modify server connection properties (Web service connections only).

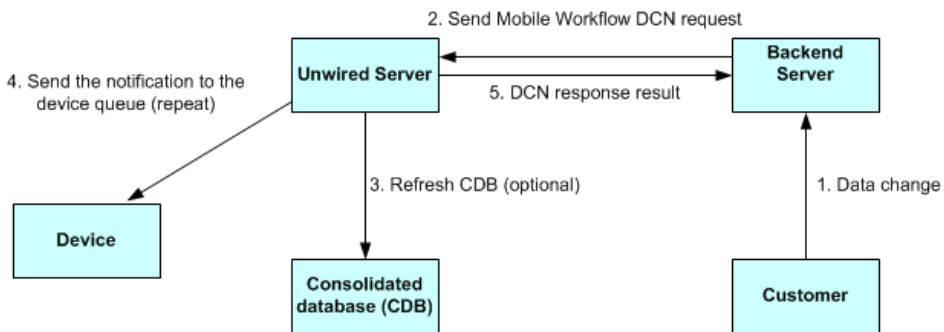
7. If a logical role is defined in your MBO, map logical roles to physical roles. If there are no logical roles defined, this page is skipped. Click **Next**.
8. (Optional) Specify the name and location for the new deployment profile:
  - Save the deployment settings as a deployment profile – if you do not save your settings to a deployment profile, they are lost when you exit the Deploy wizard.
  - Enter or select the parent folder – by default, Deployment is the folder in which the deployment profile is saved.
  - File name – the name of this deployment profile. The deployment profile is assigned a `.deploy` extension.
9. Click **Finish** to deploy the project to the Unwired Server's Packages folder.



# Mobile Workflow Data Change Notification

Mobile Workflow data change notification (DCN) requests enable you to process the general DCN request and send notification to the device to keep the Mobile Workflow in sync with the back-end server when a change to the back-end server is made.

A typical scenario might look like this:



All DCN commands support both GET and POST methods. The EIS developer creates and sends a DCN to Unwired Server through HTTP GET or POST operations. The portion of the DCN command parameters that come after `http://host:8000/dcn/HttpAuthDCNServlet`, can all be in POST; any `var=name` can be in either the URL (GET) or in the POST. The HTTP POST method is more secure than HTTP GET methods; therefore, Sybase recommends that you include the `authenticate.password` parameter in the POST method, as well as any sensitive data provided for attributes and parameters.

---

**Note:** Enter the HTTP request on a single line.

---

## Mobile Workflow DCN format

The Mobile workflow DCN request is a JSON string consisting of these fields:

1. Operation name(op) **:upsert** – the message must contain name/value pairs for every required attribute, and the name must exactly match the MBO attribute name.  
**:delete** – provide only the name/value pairs for the primary key column(s).  
These operations respectively insert or update, or delete a row in the CDB. Calling either of these operations does not trigger any other refresh action.
2. Message ID (id) of the Mobile Workflow – used for correlation (a **:delete** for a previously submitted request with **:upsert** is possible)

3. Username (to) – the Sybase Unwired Platform user name. For the user to be recognized by the Workflow DCN mechanism, the device user should first have established communication using the activation mechanism in Sybase Control Center.

---

**Note:** The "To" field must match the SUP user name—it is not the user name that is used for registering the device.

---

4. Subject (subject) – subject of the Mobile Workflow message.
5. Originator <from> – who the Mobile Workflow message is from.
6. Body of the workflow message <body> – it can embed customized information.
7. <received> – received time of the Mobile Workflow message.
8. <read> – whether the Mobile Workflow message is read.
9. <priority> – whether the Mobile Workflow message has a high priority.
10. List of dcn request <data> – JSON format string.

Example DCN request in JSON format:

```
{
  "op":":upsert",
  "id":"WID123",
  "to":"SUPAdmin",
  "subject":"Trip request approval required",
  "from":"user321",
  "body":"This is a message just used to do a test",
  "received":"2009-03-29T10:07:45+05:00",
  "read":false,
  "priority":true,
  "data":
  [
    {
      "id": "1",
      <general dcn request>
    }
    ...
    {
      "id": "4",
      < general dcn request>
    }
  ]
}
```

### *Mobile Workflow DCN request flow*

When the Mobile Workflow DCN request is received, Unwired Server gets the **cmd** value from the request first. If the value of **cmd** is *wf* the DCN request is a Mobile Workflow DCN request. The flow of the Mobile Workflow DCN request is as follows:

1. Unwired Server invokes `preProcessFilter` if the DCN filter is specified.
2. Unwired Server receives a raw HTTP POST body to generate and return a Mobile Workflow DCN request message object.
3. The JSON format string is parsed into a Mobile Workflow DCN request object.



4. The DCN request in the Mobile Workflow message object is parsed and any of them within the scope of a single transaction per DCN request object in the array are executed. Results are recorded for a report after completing the Mobile Workflow DCN request.
5. From the CDB, the server looks up all users assigned to the indicated Mobile Workflow package in the “to” attribute of the Mobile Workflow message, then matches them with the receiver list.  
For every receiver, the server generates multiple Mobile Workflow messages (all Mobile Workflow messages are created within one transaction), one per device identified (one user might have multiple devices), and then sends them to the JMS queues.  
The lookup of the logical id is performed by combining the username in the “to” list to the “securityProfile” specified in the HTTP POST REQUEST URL parameter list.
6. If any errors occur in step 4, step 5 is not executed. If any errors occur in step 5, step 5 is not committed. If any errors occur in either of those steps, an HTTP 500 error is received.
7. Unwired Server invokes the `postProcessFilter`, if specified.
8. If no errors occur, the server returns success to the caller HTTP 200 with the body of the JSON string (or any opaque data returned from the `postProcessFilter`) of the Mobile Workflow DCN Result. Otherwise, the server returns an HTTP 500 error with the body of the JSON log records.

## Mobile Workflow Data Change Notification Development Life Cycle

---

To use a DCN with the Mobile Workflow package:

1. Develop the Sybase Unwired Platform package and deploy it to Unwired Server.
2. Develop the Mobile Workflow form associated with the back-end database.
3. Log into Sybase Control Center and deploy the Mobile Workflow Package to Unwired Server.
4. Assign the Mobile Workflow package to a device.
5. Run the Mobile Workflow package on the device and subscribe.
6. Write the DCN request to update the back-end database.
7. Generate a Mobile Workflow request in the JavaScript Object Notation (JSON) data format. This request can include one, or many, general DCN requests.
8. Invoke the Mobile Workflow DCN servlet in one of two ways—HTTP authentication or non HTTP authentication.
9. Process the HTTP response. If the response code is 200, the Mobile Workflow DCN result can be read from the HTTP response, for example:

```
int returnCode = con.getResponseCode();
    if (returnCode != 200)
    {
        String rspErrorMsg = "Error getting response from the
```

```
server (error code "+ returnCode + ");
    throw new AdminException(AdminException.Type.SERVER_ERROR,
        rspErrorMsg);
}
else
{
    in = new BufferedReader(new InputStreamReader(con
        .getInputStream(), "UTF-8"));
    String line;
    while ((line = in.readLine()) != null) {
        xmlResponse.append(line).append("\n");
    }
    System.out.println("xmlResponse: " + xmlResponse);
}
```

### 10. Verify the Mobile Workflow DCN request:

- a) Check the CDB to verify it has been refreshed.
- b) If the operation of the Mobile Workflow DCN request is "upsert," check the mail inbox on the device to see if there is new mail, or if the mail message has been updated.
- c) If the operation of the request is ":delete," verify on the device that the mail message was deleted.

## Mobile Workflow DCN Request using HTTP Authentication

If you do not want to use URL parameters to send login and password information, you can send the information to the HTTP server as part of the HTTP request header.

The DCN request messages can also be sent as HTTP POST data. The URL of an HTTP authentication Mobile Workflow DCN request is:

```
http://host:8000/dcn/HttpAuthDCNServlet?
cmd=wf&security=admin&domain=default&dcn_filter=aa.bb
[post data]
```

The *[post data]* is the content of the Mobile Workflow request.

### Mobile Workflow DCN request using HTTP authentication:

```
URL url = new URL("http://<host>:8000/dcn/HttpAuthDCNServlet?
cmd=wf&security=default");
    HttpURLConnection huc = (HttpURLConnection)
url.openConnection();
    huc.setDoOutput(true);
    huc.setRequestMethod("POST");
    final String login = "supAdmin";
    final String pwd = "s3pAdmin";
    Authenticator.setDefault(new Authenticator()
    {
        protected PasswordAuthentication
getPasswordAuthentication()
    {
        return new PasswordAuthentication(login,
pwd.toCharArray());
    }
    });
```

```

    });
}

StringBuffer sb = new StringBuffer();
sb.append(wfdcn_request);
OutputStream os = huc.getOutputStream();
os.write(sb.toString().getBytes());
os.close();

```

## Non HTTP Authentication Workflow DCN Request

You can also sent the Mobile Workflow DCN request in a non HTTP authentication way.

The URL is:

```

http://host:8000/dcn/DCNServlet?
cmd=wf&security=admin&domain=default&username=supAdmin&password=s3p
Admin&dcn_filter=aa.bb&dcn_request=<wfrequestdata>

```

## Implementing the Data Change Notification Filter Class

Write and deploy preprocess and postprocess Mobile Workflow data change notification (DCN) filters to Unwired Server.

A system classloader is used to load the DCN filter class for the Mobile Workflow. To use a DCN filter class, jar the class file and put the jar file under the following folder.

1. Package your DCN filter class in a JAR file.
2. Place the JAR file in <UnwiredPlatform\_InstallDir>\Unwired Platform\Servers\Unwired Server\lib\ext.
3. At a command prompt, run: <UnwiredPlatform\_InstallDir>\Unwired Platform\Servers\Unwired Server\bin\configure-mms.bat <hostname>.

## Mobile Workflow DCN Request Response

After processing of the Mobile Workflow DCN request, Unwired Server sends the response to notify the caller whether the request was processed successfully.

The response includes two parts:

1. The result of processing the Mobile Workflow request.
2. The result of processing the general DCN requests.

The response is also in a JSON format string:

```

{
<wf dcn result>

```

## Mobile Workflow Data Change Notification

```
"result":
[
  {
    <general dcn result>
  },
  {
    <general dcn result>
  }
]
```

An example response is:

```
{
  "id": "1",
  "success": false,
  "statusMessage": "there is error in processing dcn",
  "result":
  [
    {
      "id": "1",
      "success": true,
      "statusMessage": ""
    },
    {
      "id": "2",
      "success": false,
      "statusMessage": "bad msg2"
    }
  ]
}
```

# Develop a Mobile Workflow Package

Sybase Mobile Workflow Forms Editor lets you create or modify mobile business objects on different device platforms without custom device development.

Mobile Workflows support the occasionally connected user and addresses the replication and synchronization issues those users present for the back-end system. The user can update, delete, or create data even when not connected to the network. When the user connects, data is automatically sent and received.

A Mobile Workflow application requires an integration module on the server side, which is implemented by a static set of logic that processes Mobile Workflow-specific metadata to map keys to and from mobile business object attributes, personalization keys, and parameters. This integration module processes the notifications identified by matching rules configured for the server-initiated starting point and also processes the responses sent to the server from the device.

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

When you use the Mobile Workflow Forms Editor, the data source for the application is supplied by mobile business objects that are designated as message-based synchronization mobile business objects when deployed to the server. The Mobile Workflow Forms Editor allows you to develop mobile workflow screens that can call on the create, update, and delete operations, as well as object queries of a mobile business object.

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

- Apple iOS
- Windows Mobile

See the *Sybase Unwired Platform Installation Guide* for supported version levels.

When designing the user interface for Mobile Workflow forms for mobile devices, keep it simple. The Mobile Workflow form should deliver clear, concise content with minimal navigation menus. The Mobile Workflow should be compatible with a wide variety of devices, and support both horizontal and vertical screen layouts. Consider the limitations of mobile devices when designing a Mobile Workflow form.

## The Mobile Workflow Forms Editor

---

Use the Mobile Workflow Forms Editor to develop mobile workflow forms.

The Mobile Workflow Forms Editor contains these pages:

### **Introduction**

## Develop a Mobile Workflow Package

Create a new mobile workflow or double-click an existing one, to open the Introduction page of the Mobile Workflow Forms Editor. This page is the starting point for designing your mobile workflow form. You can access help from this page.

### Flow Design

The Flow Design page displays an empty canvas if you did not choose any of starting points in the New Mobile Workflow Forms Editor. If you selected a starting point when creating the new mobile workflow form, the selected starting point appears.

Use this page to specify how starting points and screens link together. The Palette view on the right shows available starting points, screens, and connections. You can drag and drop items from the Palette to the canvas. Screens and starting points that you drag to the Flow Design canvas appear as icons. Double-click a screen to open the Screen Design page.

Right-click anywhere in the Flow Design page to open a context menu.

### Screen Design

Use this page to design the screen of your mobile workflow form. The Palette view shows the menu items and controls you can use to design a new screen. Drag controls to design the user interface.

## 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.                       |

| Field                           | Description  |
|---------------------------------|--|
| 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. |

- In the Starting Points page, select the ways the mobile workflow will start:
  - Can be started on demand from the client** – allows the user to start the mobile workflow on the device and submit data to the server.
  - Responds to server-driven e-mail notifications** – when a notification that matches the defined matching rules arrives in the Notification Mailbox, it is then sent as a notification to the devices that match those matching rules. When you select this option, you will set up the server-initiated starting point in the next screens of the wizard. Follow the procedure for *Creating a Server-initiated Starting Point*.
  - Has setup options that need to be set when the client application is first activated** – the first time the workflow is launched, the screen pointed to by the Activation screen is shown.
  - Credentials (authentication) may be requested dynamically from the client application** – select to create a credential request starting point with pre-defined username and password keys.
- (Optional) If you selected the **Responds to server-driven e-mail notifications** option in the previous step, follow the procedure in *Creating a Server-initiated Starting Point*.
- Click **Finish**.
- In the Mobile Workflow Forms Editor, click **Flow Design** to start designing the mobile workflow form.

## 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. |

## Develop a Mobile Workflow Package

| Property                                   | Description   |
|--|---|
| Module description                         | Enter a description for the module.   |
| Module display name                        | Enter the name that appears on the workflow application when it is viewed on the client.  |
| Delete message after processing            | If this option is selected, server-initiated workflow messages are removed from the device Inbox after they have been processed. A message is considered to have been processed if the corresponding workflow application is closed as a result of an asynchronous submit menu item being activated.  |
| Mark message as processed after processing | If this option is selected, server-initiated workflow messages are marked as processed in the device Inbox (indicated by a green check mark over the original icon) after they have been processed. A message is considered to have been processed if the corresponding workflow application is closed as a result of an asynchronous submit menu item being activated. |
| Client icon                                | Select the icon that appears on the client.   |
| Credentials cache key                      | Specify a key to use for the credential starting point. If you do not enter anything, each workflow will have a different credentials key. If there is more than one workflow with the same credentials key, then all of those workflows can share the same credentials so that authentication for each workflow does not have to be separately defined.                |
| Requires activation key                    | Specify a key to use for activation. If you do not enter anything, each workflow will have a different activation key, which means that each workflow will have to be activated separately. If you specify an activation key, then when you activate one workflow, all the other workflows sharing the specified activation key are automatically activated.            |



*Rulers and grid*

| Property       | Description  |
|----------------|--|
| Display        | <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 e-mail messages are redirected at runtime.

## Develop a Mobile Workflow Package

| 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.  |
| Unwired Server domain                            | Optionally specify the Sybase Unwired Platform domain to use. If specified, a context variable is created in the application for which the initial value is set to this domain. At runtime, when determining authentication, @<domain> is added to the user name, thus allowing the user access to that domain. |

## Flow Design Palette Options

Use the items in the Mobile Workflow Forms Editor Flow Design palette to configure the screens and connections for the mobile workflow application.

### *Connections*

| Connection        | Description  |
|-------------------|--|
| Go to             | Specifies which screen the user goes to upon completing an action on the current screen, such as clicking OK or Next.                                  |
| Operation success | Specifies which screen the user automatically goes to upon successful completion of an operation.  |
| Operation error   | Specifies the screen that the user goes to automatically when an error occurs while attempting to complete an operation.                               |
| Listview details  | Specifies which screen to open when a row in a listview is clicked. Once on the details screen, you can add, update, or delete rows from the listview. |
| Listview add      | Specifies a screen that can be directly navigated to in order to add a new row to the listview.  |

### *Screen*

Select the screen  and click the Flow Design page to create a new empty screen.

### *Starting points*

| Starting point | Description  |
|----------------|--|
| Activate       | The Activate starting point indicates a screen that is displayed to the user the only the first time they open the application. It is not displayed the second and subsequent times the application is opened. For example, you might want to use an Activate starting pint to show usage instructions for the application to the user the first time they open the application. |

| Starting point     | Description  |
|--------------------|--|
| 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.   |

## 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> <hr/> <p><b>Note:</b> The background color for screens can be set only for Windows Mobile devices. It is not supported for iOS devices.</p> |

| Property  | Description   |
|-----------|---|
| OK action | Select the action that will be performed when the user clicks OK on the screen. |

### *Rulers and grid*

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

| Property       | Description  |
|----------------|--|
| Display        | <ul 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.           |

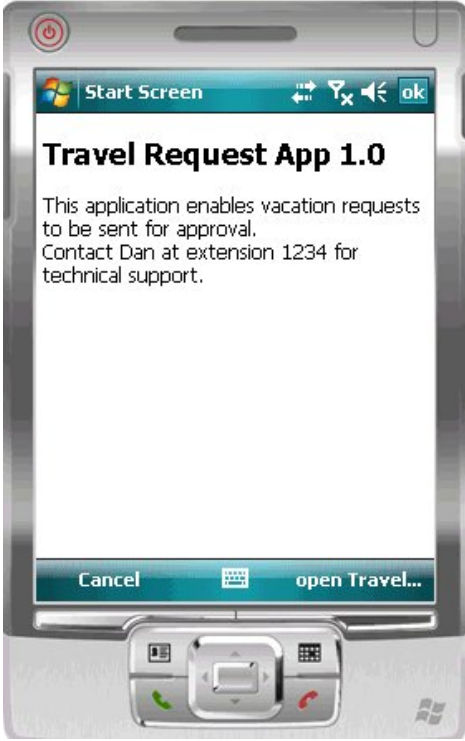
## **Screen Design Palette Options**

Use the items on the palette to design the screen for the mobile workflow application.

| Option    | Description   |
|-----------|---|
| Menu item | Use to provide additional options for the top-level menu on the screen.   |
| Checkbox  | Displays a true/false (or yes/no) option on the screen.   |
| Editbox   | Select the logical type for the edit box, then enter text and assign values to strings, dates, decimals, integers, and so on. |
| Choice    | Display a collection of items on the screen as a drop-down list from which the user can choose.                               |
| Slider    | Represent numeric values within a range.  |

## Develop a Mobile Workflow Package

| Option    | Description   |
|-----------|---|
| Signature | Add a signature box to the screen. The value is a list of points that, combined, make up the signature that gets or is drawn to the control, for example, -1x-1,5x7,5x8,-1x-1, where -1x-1 is interpreted as an interruption of a line drawn to the screen. |

| Option          | Description   |
|-----------------|---|
| <p>HtmlView</p> | <p>Allows text with HTML markup to display on the screen. For example, this code:</p> <pre data-bbox="716 288 1170 440">&lt;H2&gt;Travel Request App 1.0&lt;/H2&gt; This application enables vacation requests to be sent for approval.&lt;br&gt; Contact Dan at extension 1234 for technical support.</pre> <p>displays like this on the device:</p>  <p><b>Note:</b> The HtmlView control fills the entire screen of a workflow form, so no other controls can be placed on a screen that hosts an HtmlView control.</p> |

| Option           | Description   |
|------------------|---|
| <p>List View</p> | <p>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.</p> <hr/> <p><b>Note:</b> The ListView control fills the entire screen of a workflow form, so no other controls can be placed on a screen that hosts a ListView control.</p> |

### **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  |
|--|--|
| <p>Edit</p>                              | <p>Edit the selected object.</p>   |
| <p>Delete</p>                            | <p>Delete the selected object.</p>   |
| <p>Mobile Business Object References</p> | <p>Shows current mobile business object references and allows you to create new mobile business object references and edit or delete existing mobile business object references.</p> <hr/> <p><b>Note:</b> This menu item appears only on the Flow Design page.</p>  |
| <p>Generate Mobile Workflow Package</p>  | <p>Generate the mobile workflow form code.</p>   |
| <p>Send a notification</p>               | <p>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.</p> <hr/> <p><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.</p> |



| Option               | Description  |
|----------------------|--|
| 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.   |
| 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.   |

## 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.

## Starting Points

Starting points specify which screen on the Mobile Workflow a user sees when they perform a certain action.

Starting points include:

## Develop a Mobile Workflow Package

- Client-initiated – this starting point is activated when the user opens the Mobile Workflow on the device.
- Activate – this is the starting point that is activated when the user first opens the workflow application. It does not activate on the second and subsequent times the workflow application is started. Multiple workflow applications can share the same Activate starting point. Different workflows can specify an activation key, and workflows with the same activation key share their activation status. For example, if Workflow A and Workflow B both specify an activation key of AB (via the key attribute on the RequiresActivation tag), when Workflow A gets activated, it also activates Workflow B such that when Workflow B is invoked for the very first time, its activation screen will not be displayed.
- Credential request – this starting point is activated when the workflow application requires a user name and password and the program does not have one in its cache, or the values in the cache are no longer valid.
- Server-initiated – this starting point is activated when a notification message is sent to the device that matches the matching rules specified in the Mobile Workflow application.

Each starting point has a list of keys with which it is associated. View the keys associated with the starting point in the Properties view for the starting point.

### **Adding a Starting Point Manually**

Every screen must be part of a flow that connects to a starting point.

---

**Note:** If you manually create a server-initiated starting point that invokes an object query, the list key will not have any children. To create the children for the list key, you must use the New Mobile Workflow Forms creation wizard to create the server-initiated starting point, or you can drag and drop the mobile business object onto the Flow Design page. Either of these two methods will create the associated children key for a server-initiated starting point that invokes an object query. Otherwise, you will have to manually set up the list key's children.

---

1. From the palette on the Flow Design page, select a starting point, then click on the Flow Design canvas.
2. Add a screen to the starting point by either:
  - Dragging and dropping a data source, for example, a mobile business object or mobile business object operation, onto the Flow Design canvas, or
  - Selecting an empty screen from the Flow Design palette, then clicking on the Flow Design page.
3. Add a connection between the starting point and the screen.
4. Select **File > Save**.

You can also automatically create starting points based on selections you make in the New Mobile Workflow Form wizard when you create a new Mobile Workflow form.

## Creating a Server-initiated Starting Point

Create a server-initiated starting point.

---

**Note:** Each Mobile Workflow Forms Editor can have only one server-initiated starting point.

---

1. In the Mobile Development perspective, select **File > New > Mobile Workflow Forms Editor**.
2. Follow the instructions in the New Mobile Workflow Forms Editor wizard:

| Field                             | Description  |
|-----------------------------------|--|
| Enter or select the parent folder | Select the mobile application project in which to create the mobile workflow form.   |
| File name                         | Enter a name for the mobile workflow form. The extension for mobile workflow forms is .xbw.  |
| Advanced                          | Link the mobile workflow form to an existing file in the file system.  |
| Link to file in the file system   | Click <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. |

3. In the Starting Points page, select **Responds to server-driven e-mail notifications**.
4. In the E-mail Processing wizard, click **Search** to locate the mobile business object with which to associate the server-initiated starting point.
5. In the Search for Mobile Business Object wizard, click **Search**.
6. Select the mobile business object from the list, and click **OK**.
7. (Optional) Select an **Object query** and click **Next**.

---

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

---

8. Enter the contents for a sample notification message, and click **Next**.
9. Select the text that will serve as the matching rule by right-clicking and selecting **Select as Matching Rule**; then click **Next**.
10. From **Parameters**, select each parameter for which there is a matching rule, and specify from where the parameter receives data in the Parameter Properties.

---

**Note:** The supported syntax for the regular expression is that used by the C# Match() methods(s) of the class System.Text.RegularExpressions.Regex.

---

| Property  | Description  |
|---|--|
| Field   | Select the field from the notification message from which the parameter receives data, for example, Subject. Selecting Subject indicates that the parameter receives data from the subject line of the notification message.                               |
| Start tag   | Enter or select the text for the start tag regular expression, for example, "Approval Request\"( ".  |
| End tag   | Enter or select the text for the end tag regular expression, for example, "\)was sent".  |
| Format  | Supports locale-specific parameter extraction. For example, you can indicate that a given date parameter will be in the form of YYYY-MM-DD in the e-mail, and another date parameter will be in the form of YYYY-DD-MM. This applies to the DateTime type. |
| Sample e-mail field contents                      | The value of the field previously chosen as the matching rule in the Specify Matching Rules page of the wizard.  |
| Value extracted from sample e-mail field contents | The value the start tag and end tag matching rules would arrive at given the sample field value.   |

**11. Click Finish.**

**Server-initiated Starting Point Properties**

View and configure the server-initiated starting point properties.

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

**Table 3. General properties for the server-initiated starting point**

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

| Property         | Description   |
|------------------|---|
| Object query     | <p>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.</p> <p>Object queries that return multiple instances are supported, as well as object queries that return only one instance.</p> |
| Error screen     | Select the screen to go to if the operation fails.  |
| Jumpstart        | Launch the Notification Processing wizard, where you can compose a sample notification message from which to specify parameter value extraction rules and matching rules.   |
| Extraction rules | Shows the notification extraction rules for the server-initiated starting point. Select a rule and click <b>Edit</b> to change the transformation rule.   |

Click **Keys** in the left pane of the Properties view to see the keys that are associated with the server-initiated starting point.

**Table 4. Keys**

| Property | Description  |
|----------|--|
| Key name | The name of the associated key.                      |
| Type     | The type of the associated key, for example, string. |

| Property     | Description  |
|--------------|--|
| Data binding | <p>Shows the data source to which the key is bound:</p> <ul style="list-style-type: none"> <li>• Mobile business object attribute</li> <li>• Mobile business object relationship</li> <li>• MBO object query results</li> </ul> <p>If you select an object query that returns multiple MBO instances, you can bind the result set to a Listview control using a key that is of the list type.</p> <ul style="list-style-type: none"> <li>• User-defined</li> <li>• Extraction rule</li> </ul> <p>If you create a server-initiated starting point that invokes an object query with a parameter, this value is extracted from the email message automatically.</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> |

### **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 e-mail without needing to use them right away as parameter values. You can instead bind them to controls and display them to the user, or use them later on in the workflow to submit a create/read/update/delete (CRUD) operation.

Notification extraction rules include a regular expression that determines from where in the notification message the parameter value starts, and another regular expression that determines from where in the notification message the parameter value ends. Define the notification extraction rules when you create a server initiated starting point.

### **Editing Notification Extraction Rules**

Edit notification extraction rules in the server-initiated starting point General properties.

1. On the Flow Design page, select the Server-initiated starting point.
2. In the General Properties view, select the notification extraction rule to edit and click **Edit**.

## 3. In the Edit Notification Extraction Rule dialog:

| Property             | Description   |
|----------------------|---|
| MBO search parameter | Enter the search parameters for the mobile business object (MBO).   |
| Field type           | Select the field in the notification from which the parameter value is extracted, for example, Subject, which indicates the parameter value is extracted from the subject line of the notification message. |
| Tag before parameter | Enter the tag to indicate what to extract from the beginning of the field's contents, for example, "Approval Request\"( ".  |
| Tag after parameter  | Enter the tag that indicates what to extract from the end of the field's contents, for example, "\"\)was sent".   |

---

**Note:** The supported syntax for the regular expression is that used by the C# Match() methods(s) of the class System.Text.RegularExpressions.Regex.

---

4. Click **OK**.**Editing the Server-initiated Starting Point**

Edit the Server-initiated starting point.

1. In the Flow Design page, right-click the **Server-initiated starting point** to edit and select **Edit Server-initiated starting point**.
2. Make your changes in the Notification Processing wizard and click **Finish**.
3. From the main menu, select **File > Save**.

**Adding Matching Rules**

Matching rules determine how notification messages are redirected at runtime.

To see the Matching Rules section of the Properties view, verify there are no objects selected on the Flow Design page.

1. In the Properties view, click **Matching Rules**.
2. Click **Add**.
3. In the matching rules dialog, specify the type and regular expression for the matching rule, and click **OK**.

## **Credentials**

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

The user name and password values are required when the mobile workflow application invokes a mobile business object operation. These authentication values can be provided statically (at design time), or dynamically (by the user at runtime).

Dynamic credentials mean that the user sets the user name and password on a screen that is pointed to by the credential request starting point. The text fields must have the corresponding Credential Cache User Name and Password checkbox checked to indicate the value is to be used to provide the user name and password on the client. When the user logs in, the credentials are authenticated using the stored credentials.

---

**Note:** If an e-mail triggered workflow form has dynamic cached credentials, the cached credentials are not cached between invocations of the workflow form through an email trigger.

---

Static credentials mean that everyone who has access to the resource uses the same user name and password. By default, static credentials are used. The static credential user name and password for the mobile workflow application can be extracted from the selected Sybase Unwired Platform profile user name and password when the mobile workflow application is generated, or they can be hard-coded using the Properties view. After deployment, you can change static credentials in the Sybase Control Center.

The application can also have a credential screen (Credential Request) that appears if the mobile workflow application detects that the cached credentials are empty or incorrect.

### **Setting Up Static Authentication**

With static authentication, everyone who has access to the resource uses the same user name and password.

Set up static credentials in the Authentication section of the Properties tab. To see the Properties page, verify there are no objects selected on the Flow Design page.

1. In the Properties view, click **Authentication**.
2. Select **Use static credentials**.
3. Select from these options:
  - Use SUP Server connection profile authentication – selected by default. When the code is generated for the mobile workflow application, the user name and password associated with the SUP connection profile are used.
  - Use hard-coded credentials – sets the user name and password. When you select this option, the User name and Password fields are activated.
4. (Optional) If you selected the **Use hard-coded credentials** option in the previous step, enter the User name and Password that are to be used for authentication.
5. Select **File > Save**.



## **Setting Up Dynamic Authentication**

Use dynamic authentication when you want the user to set the name and password on the client.

You can create the Credential Request starting point with a Credential screen automatically when you initially create a new mobile workflow, or you can create the Credential Request starting point and associated screen manually. This procedure shows how to create the Credential Request starting point automatically when you create a new mobile workflow.

1. In the Mobile Development perspective, select **File > New > Mobile Workflow Forms Editor**.
2. Follow the instructions in the New Mobile Workflow Forms Editor wizard:

| <b>Field</b>                      | <b>Description</b>   |
|-----------------------------------|--|
| 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. |

3. In the Starting Points page, select **Credentials (authentication) may be requested dynamically from the client application**.
4. Continue with the New Mobile Workflow wizard as appropriate to create the type of mobile workflow application you want to create. Click **Finish**.
5. When the Mobile Workflow Forms Editor opens, click **Flow Design**.

The Credential Request starting point and its associated Credential Request screen appear on the Flow Design page.

Select the Credential Request starting point. You will see the two pre-defined keys (cc\_username and cc\_password) in the Properties view.

6. Double-click the **Credential Request** screen to go to the Screen Design page.

Two editbox controls, which are bound to the pre-defined cc\_username and cc\_password keys appear on the screen.

7. Select the **Username** editbox, then click **Advanced** on the left side of the Properties view. The Username editbox has the **Credential cache username** checkbox selected. Select the Password editbox and note that 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. Select **File > Save**.

The first time the mobile workflow is started following deployment, the credential screen is shown. The username and password values are then cached in the credential cache.

---

**Note:** If an e-mail triggered workflow form has dynamic cached credentials, the cached credentials are not cached between invocations of the workflow form through an email trigger.

---

## Creating Keys

Create new keys that can be bound to controls.

1. On the Screen Design page, select the control to which you want to bind the new key.
2. In General Properties, in the Input Data Binding section, click **New key**.
3. In the Key dialog, enter:

| Property                 | Description  |
|--------------------------|--|
| Name                     | Enter a name for the key.  |
| Type                     | Select the key type.   |
| Sent by server           | Select this checkbox to input data binding for the key.  |
| Input Data Binding       |  |
| MBO attribute            | Select to bind the key to a mobile business object (MBO) attribute: <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.  |
| User defined             | Bind the key to user-defined data.   |

4. Click **OK**.

## **Keys**

Keys are unique identifiers within the scope of a workflow application, used when setting or retrieving a value.

Keys can be bound to the mobile business object (MBO), whether it is to an MBO attribute, an MBO parameter, an MBO operation parameter, a subscription parameter, an object query, or an MBO relationship (list type keys only).

Keys can be of the following types:

- string
- int
- double
- decimal
- bool
- DateTime
- list – can be bound to relationships, or nothing. Keys of other types cannot be bound to relationships. List type keys can also be bound to MBO object query results, which returns multiple objects. List type keys have one or more child keys to which they are bound.

This example of an .xml file shows how keys are used in a workflow application. In this example, the keys associated with the editbox control are bound to the parameters of the mobile business object's (TravelRequest) create operation.

```

<screens default="TravelRequestcreate">
  <screen key="TravelRequestcreate" text="TravelRequest_create" >
    <actions>
      <action key="Submit" type="submit" text="Create" />
      <action key="Cancel" type="cancel" text="Cancel" />
    </actions>
    <controls>
      <datepicker key="TravelRequest_create_trvl_date_paramKey"
value="" label="Trvl date:" labelpos="LEFT" maxlength="32767" />
      <editbox key="TravelRequest_create_trvl_location_paramKey"
value="" label="Trvl location:" labelpos="LEFT" maxlength="20" />
      <editbox key="TravelRequest_create_est_cost_paramKey"
value="" label="Est cost:" labelpos="LEFT" numeric="true"
maxlength="32767" />
      <editbox key="TravelRequest_create_purpose_paramKey"
value="" label="Purpose:" labelpos="LEFT" maxlength="200" />
    </controls>
  </screen>
</screens>

```

## **Binding Controls to Keys**

Use the Properties view to bind keys to controls.

1. On the Screen Design page, select the control to which you want to bind the key.

2. In General Properties, in the Input Data Binding section, select the key with which to bind the control from the drop-down list.
3. (Optional) In Default Value, enter a default value for the key.
4. Select **File > Save**.

### **Specifying Values for Session Personalization Keys**

If you have a mobile business object with load parameters mapped to session personalization keys, you can specify values for those personalization keys when you invoke an operation or object query from the Mobile Workflow package.

This is useful for loading only specific data into the consolidated database (CDB) from large data sources.

1. Create a mobile business object that has load parameters. (Set the load parameters in the MBO Properties view, from the Load Parameters tab.)
2. Create a corresponding session personalization key for each load parameter.
3. Map each load parameter to the corresponding personalization key.
4. In the Mobile Workflow Forms Editor, map the personalization keys to the Mobile Workflow keys for the menu item (in the same way you map parameters to Mobile Workflow keys).

You can also fill the personalization key values from values extracted from the e-mail message, depending on from where you are invoking the object query.

## **Configuring a Mobile Workflow Package for Online Mobile Business Object Data Access**

---

Bind Personalization key values or the `findByParameter` object query to mobile workflow keys to control caching of mobile business object (MBO) data.

### **Binding the `findByParameter` Object Query to a Menu Action**

For Synchronous, real-time data access, define a Submit menu action and bind it to the `findByParameter` object query.

#### **Prerequisites**

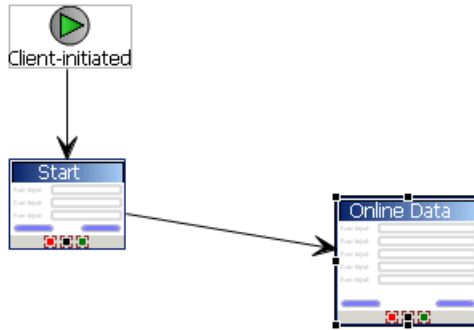
You must have propagate-to attributes mapped to MBO load parameters, and the deployed MBO must use an Online cache group policy. Unwired Platform services must be running.

#### **Task**

1. From Unwired WorkSpace, launch the Mobile Workflow Forms Editor. You can use either server-initiated or client-initiated starting points.

2. From the Flow Design screen Double-click the screen for which you are defining a mapping to open it in the Screen Design tab.

For example, you could have a client-initiated starting point with a Start screen that connects to the Online Data screen.



3. Highlight the menu item you want to map, or create a new menu item.
4. Define a Submit action that invokes the findByParameter object query:
  1. From the General tab, enter **Submit** as the Type.
  2. In the Details section, select **Search** to locate the MBO that contains the findByParameter object query.
  3. Click the **General** tab, select **Invoke object query** and select **findByParameter**.

| Menu Item [Find]   |   |
|--|---|
| <b>General</b>   | Name: <input type="text" value="Find"/>   |
| Parameter Mappings   | Key: <input type="text" value="Find"/>    |
| Personalization Key Mappings   | <input type="checkbox"/> Default          |
| Communication Style  | Type: <input type="text" value="Submit"/> |
| <b>Details</b>   |   |
| Mobile business object: <input type="text" value="OnlineRequestCacheTest/Emp"/>  |   |
| <input type="radio"/> Invoke parent update<br><input type="radio"/> Invoke operation<br><input checked="" type="radio"/> Invoke object query |   |
| Operation: <input type="text"/>  |   |
| Object query: <input type="text" value="findByParameter"/>   |   |

If you select the Parameter Mappings tab, you see all the load parameters defined for the MBO and used to generate the findByParameter object query. In addition to Key, you can

## Develop a Mobile Workflow Package

map parameters to BackEndPassword, BackEndUser, DeviceId, DeviceName, DeviceType, UserName, MessageId, ModuleName, ModuleVersion, and QueueId.

Unmapped parameters can get their value from the default value, if specified, or from the personalization key value they are mapped to, if that is specified. If the key is unmapped, and the parameter has no default value and is not mapped to a personalization key value, the parameter value is empty (NULL for string, 0 for numeric, and so on).

### **Binding Transient Personalization Keys to Mobile Workflow Keys**

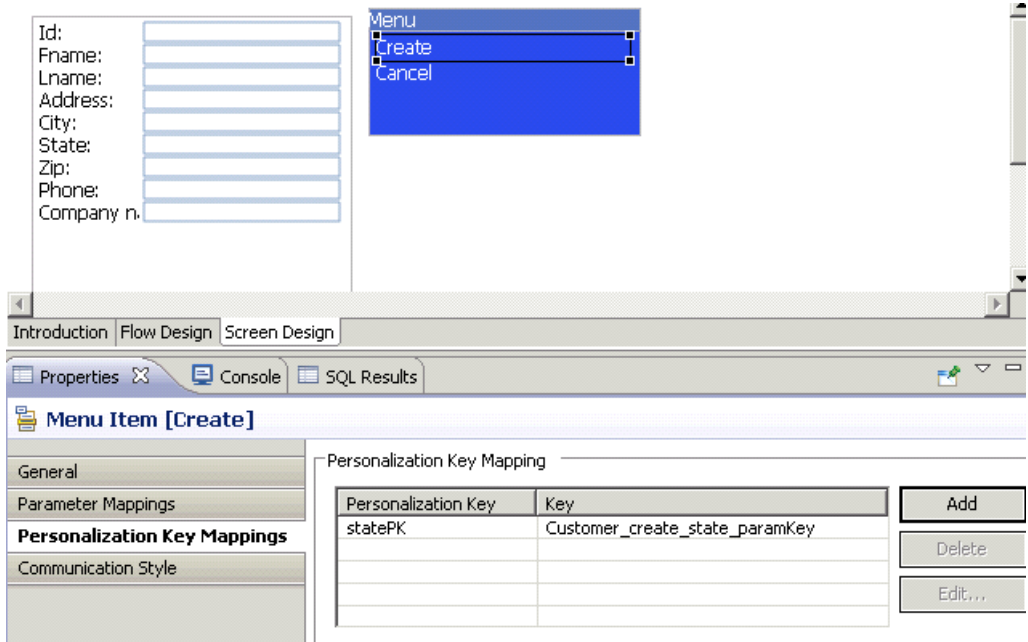
Use transient personalization key values to determine the data to be cached.

#### **Prerequisites**

You must have transient personalization keys mapped to Mobile Business Object load parameters.

#### **Task**

1. Create a new mobile workflow package and launch the Mobile Workflow Forms Editor from Unwired WorkSpace:
  - a) Select **File > New > Mobile Workflow Forms Editor**.
  - b) Select the parent folder that contains the MBO with a load parameter mapped to a transient personalization key. Name the file and click **Next**.
  - c) Select **Responds to server-driven email notifications** from the Starting Points screen and click **Next**.
  - d) Select the MBO that contains the load parameter to transient key mapping in the Search for MBO screen and click **OK**, then click **Next**.
  - e) Specify sample e-mail contents and click **Next**.
  - f) Specify the matching rules used to trigger a screen flow by highlighting the text, right-clicking it, and selecting **Select as matching rule**.
  - g) Click **Finish**.
2. In the Mobile Workflow Forms Editor, map the personalization keys to the Mobile Workflow keys for the menu item:
  - a) From the Flow Design screen select the operation for which you are defining a mapping.
  - b) Select the Screen Design tab, and highlight the menu item you want to map.
  - c) Select **Personalization Key Mappings**, click **Add**, and select a personalization key from the drop-down list and the key to which it maps.



You can also fill the personalization key values from values extracted from the e-mail, depending on from where you are invoking the object query.

When the application runs, the values are sent from the client which are used to fill the load parameter values, and determine what data is cached in the Unwired Server cache (CDB) and returned to the client.

## Mobile Business Object Relationships

When you are using two mobile business objects with a one to many or one to one composite relationship defined, use the listview control on the mobile workflow application to show the list of related objects.

The listview control allows you to navigate the one to many or one to one relationship between the two mobile business objects. The one to one cascade relationships use regular controls or keys that bind to the related mobile business object's attributes.

When you create the relationship between the two mobile business objects, you must select the One to many or One to one and Composite options.

---

**Note:** Structured mobile business objects and complex parameters are not supported in mobile workflow forms.

---

## Creating Screens Automatically by Dragging and Dropping a Data Source

---

Drag and drop a data source, for example a mobile business object (MBO) or operation onto the Flow Design page to create screens, controls, keys, menus, and navigations automatically.

---

**Note:** Mobile business objects used as the data source must be contained in a deployed mobile application package that is identified as "messaging."

---

When you use an MBO operation, a screen is created with a menu item that invokes that operation. Controls bound to keys that are bound to that operation's parameters are also created.

When you use an MBO as the data source, screens and keys for the operations are created, along with a details screen for the MBO itself, and also screens for any related MBOs and their operations.

1. From the Flow Design page palette, select a starting point and click on the Flow Design page.
2. From Workspace Navigator, expand the project that contains the data source you want to use.
3. Select the data source, for example, a mobile business object, and drag and drop it onto the Flow Design page.

If the mobile business object you are using as the data source has relationships with other mobile business objects defined, all of the screens and connections between screens for the initial mobile business object and its relationships are automatically created. You can change the order of the screens and create your own connections manually.

---

**Note:** Structured mobile business objects and complex parameters are not supported in mobile workflow forms.

---

## Adding Controls to the Screen

---

Use the Screen Design palette to add controls to the mobile workflow screen.

1. In the Mobile Workflow Forms Editor, select **Screen Design > Palette > Controls**, then select the control you want to add to the screen and click on the screen.
2. Click the control to select it, then use the Properties page to configure it.



## **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.

| <b>Control</b> | <b>Supported Platform</b>  |
|----------------|--|
| Checkbox       | <ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• Windows Mobile Professional</li> </ul> |
| Editbox        | <ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• Windows Mobile Professional</li> </ul> |
| Choice         | <ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• Windows Mobile Professional</li> </ul> |
| Slider         | <ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• Windows Mobile Professional</li> </ul> |
| Signature      | <ul style="list-style-type: none"> <li>• Windows Mobile Professional</li> </ul>                      |
| HtmlView       | <ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• Windows Mobile Professional</li> </ul> |
| Listview       | <ul style="list-style-type: none"> <li>• Apple iOS</li> <li>• Windows Mobile Professional</li> </ul> |

## **Adding a Menu Item**

Add a menu item to the menu.

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

1. On the Screen Design page, from Palette, select **MenuItem**, then click the Menu box.
2. Select the menu item to configure the properties in the Properties view.

### **Menu Items**

Menu items function as actions and are associated with mobile business object operations.

Menu items can be one of these types:

## Develop a Mobile Workflow Package

- **Open** – must be associated with another screen. The open menu item is created automatically when a Go To connection is created on the Flow Design page.
- **Save** – allows the user to close the current open screen and save the input made to the collection of response values to be sent to the server using the submit action. Using the save menu item, a user can navigate back one single screen at a time.
- **Cancel** – allows the user to close the current open screen without performing any data validation. The cancel action type immediately navigates the user back to the underlying screen.
- **Submit (synchronous)** – the submit menu item on a screen must be associated with a mobile business object operation or an object query. If the user has multiple screens open during the submit execution, and one of the underlying screens fails validation, the user is prompted with the configured warning message and the submit process stops.
- **Submit (asynchronous)** – the submit menu item on a screen must be associated with a mobile business object operation.
- **Close** – allows the user to leave the client-side component no matter which screen the client-side component currently shows.
- **Listview** – includes AddListView Row, Delete ListView Row, UpdateListView Row, which allows the user to add, delete, or update a listview row. The AddListView Row menu type allows you to navigate directly from the listview details screen and perform an add from the subsequent screen.
- **Add** – available for ListView controls. The Add menu item type allows you to navigate directly from the listview itself and perform an add from the subsequent screen.

### **Menu Item Properties**

Select the menu item on the Screen Design page to view and configure its properties using the Properties page.

#### *General*

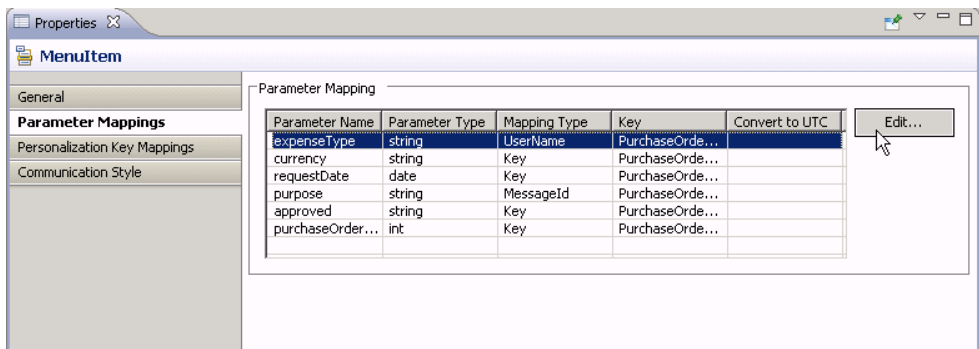
| <b>Property</b> | <b>Description</b>   |
|-----------------|--|
| Name            | Enter a valid name for the menu item.                      |
| Key             | Select the key with which to bind the menu item.           |
| Default         | Select to display the menu item prominently on the screen. |

| Property | Description  |
|----------|--|
| Type     | <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> <li>• Submit – Closes the client application and invokes a mobile business object method using the values of the current flow.</li> <li>• The Listview actions can be invoked only within the context of a Listview, for example, on a screen that is arrived at after navigating from a listview. It does not immediately execute the MBO create operation in question, but rather stores it so that later, when the parent MBO is submitted, this, and other, MBO operations will be invoked. <ul style="list-style-type: none"> <li>• Add Listview Row</li> <li>• Update Listview Row</li> <li>• Delete Listview Row</li> </ul> </li> <li>• Add – available for ListView controls. The Add menu item type allows you to navigate directly from the listview itself and perform an add from the subsequent screen.</li> </ul> |

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

### Parameter Mappings

Use the Parameter Mapping section to bind the parameters (if any) of an operation or object query to keys or, context data, such as DeviceName, BackEndPassword, BackEndUser, and so on.



### Communication Style

| Property     | Description   |
|--------------|---|
| Asynchronous | Validates the data input on all open screens, closes the application, and sends the data to the server. Enter the confirmation message to submit to the server. |

| Property    | Description   |
|-------------|---|
| Synchronous | <p>Validates the data input on all open screens, saves it, send it to the server and waits for a response from the server before continuing. Enter:</p> <ul style="list-style-type: none"> <li>• Submit error message – enter the message the user sees if there is an error.</li> <li>• Timeout – enter the allowed amount of time to wait for the server to respond before timing out and giving an error message.</li> <li>• Success screen – select the screen to go to upon successful communication with the server.</li> <li>• Error screen – select the screen to go to if there is an error when submitting the data to the server.</li> </ul> |

### **Creating a Save, Open, Close, or Cancel Menu Item**

Create a menu item that performs a save, open, close, or cancel action on the device application screens.

---

**Note:** The first screen in each mobile workflow form should have a Close or Cancel menu action assigned to it so that the user can exit from the mobile workflow form on the device. On a Windows Mobile device, the user can exit by clicking the OK button, but not all devices have that function.

---

1. On the Screen Design page, from Palette, click **Menuitem**, then click the Menu box.
2. Select the menu item to configure the properties in the Properties view.
3. In the Properties view:

| Property | Description  |
|----------|--|
| Name     | Enter a valid name for the menu item.                      |
| Key      | Select the key with which to bind the menu item.           |
| Default  | Select to display the menu item prominently on the screen. |

| Property | Description  |
|----------|--|
| Type     | <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   | <p>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.</p>  |

4. Select **File > Save**.

**Creating a Submit Menu Item**

Create a "submit" menu item.

The submit menu item of the first screen in a mobile workflow form must be associated with a mobile business object operation or an object query.

If the user has multiple screens open at the time of the submit execution, and one of the underlying screens fails validation, the user is prompted with the configured warning message and the submit process stops.

1. From the palette on the Screen Design page, click **MenuItem**, then click the Menu box.
2. Select the menu item to configure the properties in the Properties view.
3. In the Properties view, configure the properties for the submit menu item:

| Property | Description   |
|----------|---|
| Name     | Enter a valid name for the menu item.                   |
| Key      | Select the key with which to bind the submit menu item. |

| Property | Description  |
|----------|--|
| Default  | Select to display the menu item prominently on the screen. |
| Type     | Select <b>Submit</b> from the drop-down list.              |

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

| Property               | Description  |
|------------------------|--|
| Mobile business object | The mobile business object with which the menu item is associated. Click <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       | Invoke the selected mobile business object operation with which the menu item is associated.<br><br>Select the operation from the drop-down list.<br>The list is populated when you select the mobile business object. |
| Invoke object query    | Invoke the object query from the menu item.<br><br><b>Note:</b> The menu item must be the Submit type menu item and the communication style must be synchronous.   |

Configure these properties in Advanced:

| Property                   | Description   |
|----------------------------|---|
| Flow-specific username key | User name with which to connect when invoking this operation.     |
| Flow-specific password key | User password with which to connect when invoking this operation. |

5. Select **File > Save**.

#### Configuring Communication Style

Choose whether to make the submit action for the menu item synchronous or asynchronous.

1. In the Properties view for the menu item, select the **Communication Style** tab.
2. Select from the options:

| Property     | Description  |
|--------------|--|
| Asynchronous | Validates the data input on all open screens, closes the application, and sends the data to the server. Enter the confirmation message to submit to the server.  |
| Synchronous  | <p>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.</p> <hr/> <p><b>Note:</b> When executing a synchronous Submit menu item, then subsequently closing the application, it is not treated as if it was processed. It is only treated as having been processed if the application was closed through an asynchronous Submit menuitem.</p> <hr/> <p>Enter:</p> <ul style="list-style-type: none"> <li>• Submit error message – enter the message the user sees if there is an error.<br/>When an operation is invoked, the user can receive any error messages that may occur on the client application. The error messages are a single key of the list type with a single key of the string type as its child. Each of the child instances corresponds to an error message.</li> <li>• Timeout – enter the allowed amount of time to wait for the server to respond before timing out and giving an error message.</li> <li>• Success screen – select the screen to go to upon successful communication with the server.</li> <li>• Error screen – select the screen to go to if there is an error when submitting the data to the server.</li> </ul> <p>Click <b>Generate Error Screen</b> to generate error screens automatically.</p> |

3. Select **File > Save**.

**Creating a ListView Menu Item**

Create and configure a Listview menu item.

1. In the Screen Design page, from the palette, click **Menuitem**, then click the Menu box.
2. Click the menu item to select it and configure the properties in the Properties view.



3. In the Properties view:

| Property | Description   |
|----------|---|
| Name     | Enter a valid name for the menu item.   |
| Key      | Select the key with which to bind the Listview menu item.   |
| Default  | Select to display the menu item prominently on the screen.  |
| Type     | Select one of these types: <ul style="list-style-type: none"> <li>• Add Listview Row</li> <li>• Update Listview Row</li> <li>• Delete Listview Row</li> </ul> |

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

| Property               | Description   |
|------------------------|---|
| Mobile business object | The mobile business object with which the menu item is associated. Click <b>Search</b> to locate the mobile business object with which to associate the menu item.              |
| Operation              | The mobile business object operation with which the menu item is associated. Select the operation from the list, which is populated when you select the mobile business object. |
| Parameter mapping      | Parameter mapping for the selected operation is shown. Click <b>Edit</b> to edit the parameter mapping for the operation.   |
| Listview               | Select the Listview from the drop-down list.  |

5. Select **File > Save**.

### **Configuring Parameter Mappings**

Configure mapping between parameters and keys or context data.

Parameter mappings for the selected operation or object query are shown in the Properties view.

1. In the Properties view for the menu item, select the **Parameters Mapping** tab.
2. Click **Add** to map a key or context data, such as BackEndUser, BackEndPassword, DeviceName, UserName, MessageID, and so on to a parameter.
3. Select a key, and click **Edit** to edit mapping between a parameter and a key or context data.

4. Select a key and click **Delete** to delete existing mapping between a key or context data and a parameter.

## Configuring Control Properties

---

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

1. On the Screen Design page, select the control for which to configure the properties. The configurable properties for the selected control appear in the Properties view.
2. Configure the properties for the control and select **File > Save**.

## 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.<br/>Click <b>New key</b> to create a new key for the control.</li> <li>• Default value – default value for the control.</li> </ul>                       |
| Read only          | <p>This property applies to all controls except HtmlView and Listview. Select to make the control read-only.</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> |
| 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.<br/>Click <b>New key</b> to create a new key for the control.</li> <li>Default value – default value for the control.</li> </ul>                       |
| Read only          | This property applies to all controls except HtmlView and Listview. Select to make the control read-only.  |

### *Editbox general properties*

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

| Property | Description  |
|----------|--|
| Password | If selected (set to true), text entered in the editbox is replaced with asterisks. The default is false. |

| Property            | Description  |
|---------------------|--|
| <p>Logical type</p> | <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  |
|---------------------|--|
| <p>Content size</p> | <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 | If the widget supports cached credentials, select to specify that the editbox be treated as a user name.   |
| Credential cache password | If the widget supports cached credentials, select to specify that the editbox be treated as a password.  |

## Choice Properties

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

### *General control properties*

These are general properties for most controls.

| Property           | Description  |
|--------------------|--|
| Label              | <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.<br/>Click <b>New key</b> to create a new key for the control.</li> <li>• Default value – default value for the control.</li> </ul>                       |
| Read only          | <p>This property applies to all controls except HtmlView and Listview. Select to make the control read-only.</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> <hr/> |
| 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 the Display Name and Value for the choice item, and click **OK**.

| Property     | Description   |
|--------------|---|
| Display Name | Enter the name for the choice item as it will appear on the user interface. |
| Value        | Enter the value for the choice item.  |

### **Editing Choice Items**

Edit choice items for the choice control.

1. In the General Properties view for the choice control, select the choice control item to edit and click **Edit**.
2. In the Choice Option Item dialog, make the changes and click **OK**.

| Property     | Description   |
|--------------|---|
| Display Name | Enter the name for the choice item as it will appear on the user interface. |
| Value        | Enter the value for the choice item.  |

### **Deleting Choice Items**

Delete choice items from the choice control.

1. In the General Properties view for the choice control, select the choice control item to delete.
2. Click **Delete**.

### **Creating Dynamic Contents for the Choice Control**

Specify contents of the choice control dynamically.

1. In the Screen Design page, select the Choice item from **Palette > Controls** and drag it onto the screen.
2. In the General Properties view for the choice control, from the Options section, select **Dynamic**.
3. In **Value key**, enter the value for the key.
4. In **Display name key**, enter the value that the user will see on the device interface.

---

**Note:** The display name key and the value name key can be different or the same. The two keys must be children of a key that is a list type.

---

## **Slider Properties**

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

### *General control properties*

These are general properties for most controls.

| <b>Property</b> | <b>Description</b>  |
|-----------------|---|
| 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.<br/>Click <b>New key</b> to create a new key for the control.</li> <li>Default value – default value for the control.</li> </ul> |
| Read only          | This property applies to all controls except HtmlView and Listview. Select to make the control read-only.  |

### *Slider control general properties*

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

**Table 5. Value constraints**

| Property      | Description                |
|---------------|----------------------------|
| Minimum value | Specify the minimum value. |
| Maximum value | Specify the maximum value. |

### *Advanced*

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

| Property           | Description   |
|--------------------|---|
| Validation message | Specifies the error message displayed to the user in case the input validation criteria do not match. |

## **Signature Properties**

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

---

**Note:** The signature control is not supported on Apple devices.

---

### *General control properties*

These are general properties for most controls.

| Property           | Description  |
|--------------------|--|
| Label              | <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.<br/>Click <b>New key</b> to create a new key for the control.</li> <li>Default value – default value for the control.</li> </ul>                       |
| Read only          | <p>This property applies to all controls except HtmlView and Listview. Select to make the control read-only.</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.

The HtmlView control fills the entire screen of a workflow form, so no other controls can be placed on a screen that hosts an HtmlView control.

**Table 6. 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.<br/>Click <b>New key</b> to create a new key for the control.</li> <li>Default value – default value for the control.</li> </ul> |

## Listview Properties

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

### *General Listview properties*

| Property                | Description  |
|-------------------------|--|
| Input data binding      | Select the key with which to bind the Listview control from the drop-down, or click <b>New key</b> to create a new key.  |
| 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> <p><b>Note:</b> The alternate row color attribute is supported on Windows Mobile devices only.</p> |
| On empty list           | The text that appears on the screen when the Listview is empty.  |
| Listview details screen | Select the screen that will be opened when the user clicks a row.  |

*Cells*

| Property   | Description  |
|------------|--|
| Cell lines | The lines that appear in each row of the control. Click <b>Add</b> to add a new line. Select an existing line and click <b>Delete</b> to remove a line from the row. |
| Fields for | Click <b>Add</b> to create a new field.<br>Click <b>Edit</b> to edit an existing field.<br>Click <b>Delete</b> to delete an existing field.                          |

**Adding a Listview Field to a Line**

Use the Properties view to add fields to Listview lines.

1. Select the **Listview** control to see the Properties view.
2. In Properties, in the left pane, click **Lines**.
3. In **Listview lines**, select the line for which to add the field.
4. In **Fields for** click **Add**.
5. In the Listview Field dialog:

| Option       | Description  |
|--------------|--|
| Key          | Select the key with which to associate the field.  |
| Field width  | Enter the field width. This value is a percentage, so the values entered should be between 1 – 100 and add up to 100.  |
| Font options | Select the appearance of the font: <ul style="list-style-type: none"> <li>• Bold</li> <li>• Italics</li> </ul>   |
| Field format | Enter the field format. A standard format string contains a format specifier, which is an alphabetic character, for example, G, that defines the string that represents the object to which it is applied.<br><br><b>Note:</b> This attribute is supported only on Windows Mobile Platforms. |

6. Click **OK**.


### **Editing a ListView Field**

Edit ListView control fields in the Properties view.

1. Select the ListView control to see the Properties view.
2. In Properties, in the left pane, click **Lines**.
3. In Fields for ListView Line, select the line for which to edit the field and click **Edit**.
4. In the ListView Field dialog, make your changes and click **OK**.

## **Adding Connections to Screens**

---

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

1. From the Flow Design Palette, select the connection you want to use:

**Note:** Some connections cannot be explicitly created; they can only be implicitly created when creating or editing menu item properties.

| <b>Connection</b> | <b>Description</b>  |
|-------------------|---|
| Go to             | Specifies which screen the user goes to upon completing an action on the current screen, such as clicking OK or Next.                                 |
| Operation success | Specifies which screen the user goes to upon successful completion of an operation.   |
| Operation error   | Specifies the screen that the user goes to automatically when an operation fails to execute.  |
| Listview details  | Specifies which screen to open when a row in a listview is clicked. Once on the details screen you can add, update, or delete rows from the listview. |
| Listview add      | Specifies a screen that can be directly navigated to in order to add a new row to the listview.   |

2. In the Flow Design canvas, click the source screen and drag to connect it to the target screen.  
The connection line appears between the two screens.

## 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 |   |

| Property   | Description  |
|------------|--|
| 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> |

## Activating the Workflow

---

The last screen must have an asynchronous submit action.

The submit action on a Workflow screen may be either synchronous or asynchronous.

To complete the workflow activation process, the last screen must have an asynchronous submit action. This is necessary in order for the device and sever side to activate the workflow for the device.

## Localization

---

You can localize different objects in the Mobile Workflow Forms Editor, such as the names of screen controls, screens, and mobile business objects.

You can localize the mobile workflow by creating locale properties files. You can then load, update, and generate localized device applications.

You can create or update resource bundles for the Mobile Workflow Forms Editor using the resource keys in the locale properties file. The values for the resource keys are defined by the corresponding elements in the current Mobile Workflow Forms Editor. You can also upload specific locale resource bundles into the Mobile Workflow Forms Editor.

Some restrictions for the locale properties files apply:

## Develop a Mobile Workflow Package

- When you specify a country for the language, the basic language locale must also be available. For example, if you create a locale and specify English as the language and the United States as the country, then a locale for English (the basic language) must also be available.
- If you create a locale that specifies language, country, and variant, the locale for the basic language and the locale for the basic language and the country must be available. For example, if you create a locale and specify English as the language, United States as the country, and WIN as the variant, then English (United States) and English locales must also be available.
- The language code must be a 2-letter code, and the country code can be either a 2-letter or 3-letter code.
- If you specify a variant, the country code must be a 2-letter code.

All locale properties files are saved in the same directory as the device application.

When localizing the Mobile Workflow Forms Editor:

- Mobile workflow applications that have names that begin with numbers or special characters cannot be localized; you will receive an error when you generate the code. Make sure that any mobile workflow you want to localize does not have a file name that begins with a number or special character.
- If you manually localize the locale properties file using an external editor, you must make sure the file is encoded in ASCII, so that the content can be correctly read and converted to Unicode.

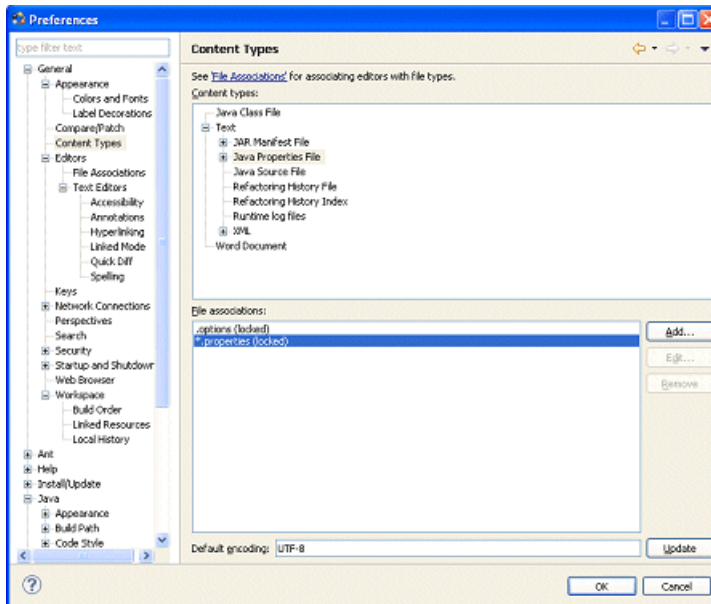
## Changing the Encoding Type

**Goal:** Change the encoding type in Preferences.

The localization file is encoded in standard ISO-8859-1. All non-ASCII character values will be converted to escaped Unicode hexadecimal values before writing 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 a New Locale

Create a new locale with resource keys and values extracted from localizable strings in the current device application file.

When you create a new locale, keep in mind:

- When you specify a country for the language, the basic language locale must also be available. For example, if you create a locale and specify English as the language, then there must also be a locale for English (the basic language).
- If you create a locale that specifies language, country, and variant, the locale for the basic language and the locale for the basic language and the country must be available. For example, if you create a locale and specify English as the language, United States as the country, and WIN as the variant, then English (United States) and English locales must also be available.

1. In the Properties view, click **Localization** in the left pane.
2. In the Localization page, click **New**.
3. In the New Locale wizard:

| Option   | Description          |
|----------|----------------------|
| Language | Select the language. |
| Country  | Select the country.  |

| Option                              | Description  |
|-------------------------------------|--|
| Variant                             | Enter the variant, which is the vendor or browser-specific code. For example, enter "WIN" for Windows, "MAC" for Macintosh and "POSIX" for POSIX. If there are two variants, separate them with an underscore, and put the most important one first. For example, a Traditional Spanish collation might construct a locale with parameters for language, country, and variant as: "es", "ES", "Traditional_WIN". |
| Overwrite existing file             | Overwrite an existing localization file.   |
| Automatically create default locale | Automatically create the default locale properties file. For example, if you specify the language as "English" and the country as the "United States" for a device application called <code>test</code> , then both <code>test_en_us.properties</code> and <code>test.properties</code> files are created.   |

4. Click **Finish**.

The properties file is created in the same directory as the device application file.

## Editing a Locale Properties File

Use the Properties File Editor to edit the localization properties file.

1. In the Screen Design page Properties view, click **Localization**.
2. Select the locale to edit and click **Edit**.

**Note:** If you use an external editor to manually edit the locale properties file, make sure the file is encoded in ASCII so that the content can be correctly read and converted to Unicode.

3. Make the changes to the file.
4. Select **File > Save**.

## Updating the Current Locale

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

If the locale properties file does not already exist, it is created. If the current locale is not defined in the mobile workflow application file, the updated locale is used as the default, and the file name is `{device_application}.properties`. Otherwise, the locale defined in the mobile workflow application file is updated.

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

2. Click **Update**.

## **Removing a Locale**

Remove locale properties files.

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

## **Loading a Locale Properties File**

Load a locale properties file.

1. In the Screen Design page Properties view, click **Localization**.
2. Select the locale properties to load and click **Load**.  
You see a message if there are missing resource keys.
3. (Optional) Click **Yes** to automatically fix the properties file.  
If you choose to not fix the locale properties file, loading does not continue.

Loading the locale property file updates the current locale of the device application file.

## **Validating the Locale Properties Files**


Validate locale properties files.

Validation verifies that all the resource keys for the current mobile workflow application file exist in the locale properties file.

1. In the Flow Design Properties view, click **Localization**.
2. Select the locale for which to validate the properties file, and click **Validate**.  
You see a message if there are missing resource keys.
3. (Optional) Click **Yes** to automatically fix the properties file.

## **Generating the Files For a Mobile Workflow Package**

Use the Mobile Workflow Package Generation wizard to generate the files for the mobile workflow form, optionally deploy the generated workflow package files to the server, and assign it to one or more users' devices.

1. Right-click in either the Flow Design or Screen Design page of the Mobile Workflow Forms Editor and select **Generate Mobile Workflow Package**, or click the code generation icon  on the toolbar.
2. In the Mobile Workflow Package Generation wizard, enter or select:

## Develop a Mobile Workflow Package

| Option  | Description  |
|---|--|
| Favorite configurations   | (Optional) Select a configuration.   |
| Code Generation and Deployment  |  |
| Generation folder   | Select the folder in which to save the .zip file containing the application and its generated files.   |
| Deploy to an Unwired Server   | Select to deploy the mobile workflow application to an Unwired Server.   |
| Unwired Server Profile  | Select the Unwired Server profile with which to associate the mobile workflow application and extract the username and password credentials if you are using static authentication.  |
| 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. It can be assigned 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.</p> |
| Deploy the Sybase messaging client to a connected Microsoft Windows Mobile device or emulator | <p>Select to deploy the mobile workflow application to a Windows Mobile connected device or emulator.</p> <hr/> <p><b>Note:</b> This option requires Microsoft ActiveSync 4.5 (if you are using Windows XP) or Windows Mobile Device Center (Windows Vista, Windows 7, Windows 2008) to be installed and connected.</p> <hr/>      |
| Windows Mobile device type  | <p>Select the Mobile Windows device type from the drop-down list.</p> <ul style="list-style-type: none"> <li>• Windows Mobile Professional</li> </ul>  |
| Perform silent install  | Select to perform the deployment on the device without requiring user input.   |
| Deployment timeout (minutes)  | Enter the number of minutes allowed before the deployment times out.   |
| Start Device Emulator Manager   | Start the Device Emulator Manager. The Device Emulator Manager is used to connect and cradle the selected device.  |

3. (Optional) Click **Start Device Emulator Manager** to launch the Emulator Manager so you can connect to different Windows Mobile device types.
4. Click **Finish**.  
A `.zip` file containing the application and its generated files is created and placed in the specified location.

Develop a Mobile Workflow Package

# Preparing Devices

To enable deploying Workflow packages to a device, you must download, install, and configure a Workflow container on the device.

Refer to the appropriate procedure for your device type.

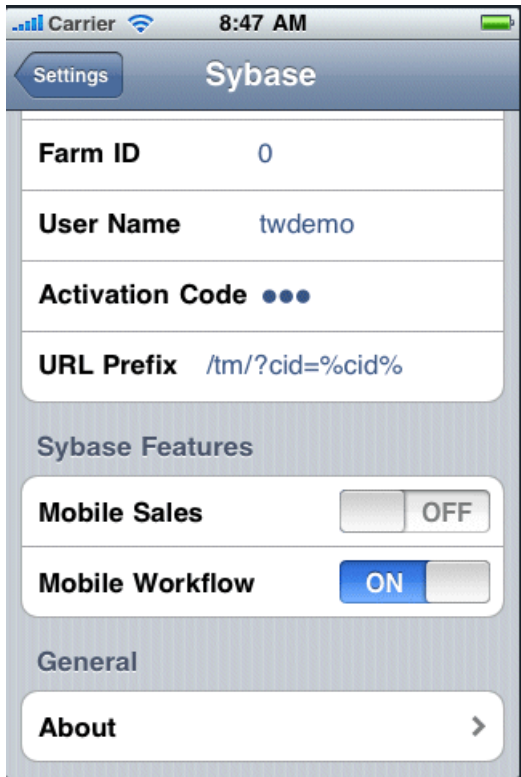
## Preparing iPhone Devices for the Workflow Package

Install the Mobile Workflow package on the device using the App Store, or use the sample Workflow container to deploy to the iPhone simulator from the Xcode project.

### Installing the Mobile Workflow Package on the Device Using the Mobile Workflow Container Downloaded From App Store

Install the Mobile Workflow container from the iPhone App Store.

1. On the iPhone home page, tap **App Store**.
2. Search for **Sybase**.
3. When the Sybase Mobile Sales & Workflow application appears, tap **Free**.
4. Tap **Free** again on the Mobile Sales information page.
5. Tap **Install** to download the application.
6. In **Settings > Sybase**, for Connection Info, enter:
  - a) In Connection Info, enter:
    - Server Name – your unwiredserver.company.com or your relayserver.company.com
    - Server Port – 5001 your Unwired Server port or relay server port
    - Farm ID – company ID, or Farm ID registered in the relay server
    - User name – the user name you entered for the device in Sybase Control Center
    - Activation Code – the activation code for the device
  - b) In Sybase Features, turn the Mobile Workflow option to **On** and the Mobile Sales option to **Off**.



7. Scroll to the page that contains the **Sybase** icon, then tap to launch. The **Workflows** page appears.
8. To view the inbox e-mails, click the **Messages** Tab bar, then tap **Messages** to view the Workflows.

## Deploying the Mobile Workflow Package to the iPhone Simulator

Deploy the Mobile Workflow package to an iPhone simulator for testing.

### Prerequisites

- You must have a registered device user in Sybase Control Center to which to assign the mobile workflow. See the Sybase Control Center online help topic *Configure > Configuring Unwired Platform > Device Users > Devices > Messaging Devices*.
- You must have a Mac with iPhone SDK 4.0.x installed.
- Xcode 3.2.3 or higher.
- Generate the Mobile Workflow package.



**Task**

The Workflow Container referenced in this procedure is a sample container. You can use Xcode to build your own customized user interface and configure other resources.

1. Log into Sybase Control Center and assign the mobile workflow package to registered device users.

See the Sybase Control Center online help topic *Configure > Configuring Unwired Platform > Mobile Workflows > Configuring a Mobile Workflow Package*.

2. Copy these directories from your Sybase Unwired Platform installation %Sybase%\UnwiredPlatform\Servers\UnwiredServer\ClientAPI\ObjectiveC to a location on your Mac:

- includes
- libs
- libs.iPad
- images
- strings
- WorkflowContainer

You can copy these files into whatever location you choose, but they should all be at the same directory level.

3. In the WorkflowContainer folder, unzip WorkflowContainer-iPhone.zip into the same directory level in which you copied the other folders.

This creates a SUPWorkflows folder.

4. In the images folder, unzip the WorkflowImages.zip file in its current location.

It unzips as a workflowimages folder.

5. In the SUPWorkflowContainer folder, double-click **SUPWorkFlows.xcodeproj** so that it opens in the XCode IDE.

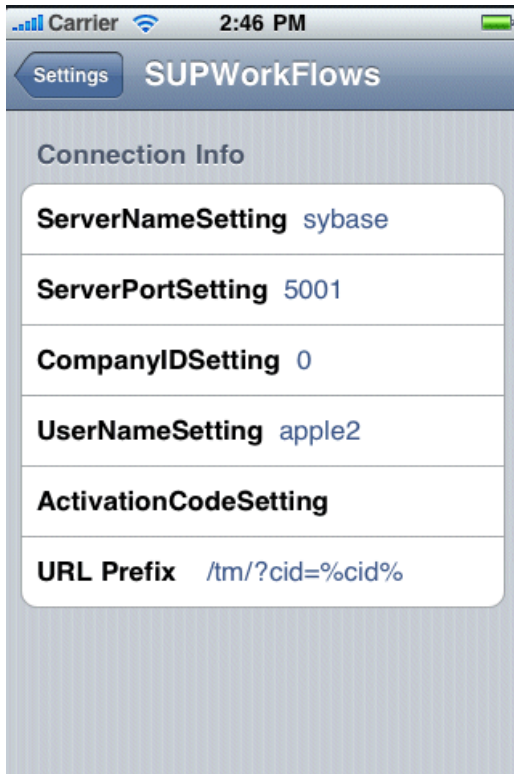
6. (Optional) If you are building a project for the iPad simulator, go into the Target settings for the project, and change the **Library Search Paths** to replace libs with libs.iPad.

7. In XCode, select **Build > Build**.

The project builds and the simulator starts.

8. In the Settings for the simulator, choose **SUPWorkflows** and verify the connection settings.

The ServerNameSetting should be the name of the machine on which the Mobile Workflow package is deployed to Unwired Server.



9. In the applications screen, open the SUPWorkFlows applications.  
The deployed Mobile Workflow package is shown in the SUPWorkFlows menu.

## Preparing Windows Mobile Devices for the Mobile Workflow Package


---

Install and configure the container required to prepare a Windows Mobile device to run Mobile Workflow packages.

1. Navigate to `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\ClientAPI\MoMessaging\wm`.
2. Copy the Windows Mobile Professional device file, `SUPMessaging_Pro.cab`, to the device's **My Documents** folder.
3. Cradle the Windows Mobile device.
4. Connect a USB cable between the PC and device, and transfer the `.cab` file.
5. Open the `.cab` file from the Windows Mobile device. This installs the container.

6. In the Connection screen, enter the connection settings. These settings should match the values you used when you registered the device in Sybase Control Center.
  - Name/IP Address – your unwiredserver.company.com or your relayserver.company.com
  - Host Port – 5001 your Unwired Server port or relay server port
  - CompanyID –company ID, or Farm ID registered in the relay server
  - User name – the user name you entered for the device in Sybase Control Center
  - Activation Code – the activation code for the device

---

**Note:** Select the right arrow icon () to view the container log. This is useful for checking the connection, or retrieving other debugging information.

---

7. Access your Microsoft Outlook inbox on the device. For example:
  - On Windows Mobile Touch Screen devices, tap **Start > Messaging**, and then scroll to and tap Outlook Inbox.
  - On Windows Mobile Non-touch Screen devices, click **Start > Messaging** on the Home screen. Scroll to a Outlook Inbox and press **ENTER**.



8. Select **Menu > Workflows**.



9. Select a workflow package from the list of available packages.

## Preparing Devices



## Launching a Mobile Workflow on the Device

Launch the deployed Mobile Workflow from the device inbox.

When you select the Workflow menu item in the device inbox, only the latest version of the Workflow appears. If you click the Workflow notification itself, the Workflow version that is associated with the e-mail notification is launched whether it is the latest version or not.

### *Example*

You develop a Mobile Workflow application that has both client-initiated and server-initiated starting points. You deploy the initial version, which is called version 1, and a Mobile Workflow e-mail notification is sent.

Next, make some changes and deploy a second version, version 2. Again, a Mobile Workflow e-mail notification is sent.

There are now three ways that this Mobile Workflow application can be launched, and the way that it is launched determines which version of the Workflow is launched:

- If you launch the application from the Workflows menu, the last deployed version of the Workflow, in this case, version 2, is launched. Although version 1 of the Mobile Workflow application still exists somewhere on the device it is never used as long as you launch the Workflow from the Workflows menu.
- If you launch the Workflow by opening the initial e-mail notification, the version that corresponds with the latest version that existed at the time the notification was sent, is used. In this case, that is version 1; it does not matter that a later version (version 2) exists.
- If you launch the Workflow by opening the second notification, the version that corresponds with the latest version that existed at the time the notification was sent is used. In this case, that is version 2.





# Testing Mobile Workflow Packages

To test a Mobile Workflow package:

1. Launch and/or connect to the mobile device or emulator.
2. Deploy the Mobile Workflow package to the device.
3. Establish the connection to the server on the device.
4. For user-initiated Mobile Workflow packages, go to the Workflows menu of the e-mail inbox and click on the appropriate Workflow package.
5. For e-mail subscription Mobile Workflow packages, send the e-mail to the device, either automatically, for example, database trigger, or manually, through the Send E-mail dialog; then open that e-mail on the device.
6. Enter data and execute menu items appropriately.
7. Verify that the backend is updated correctly.
8. Check the logs.

## Testing Server Initiated Mobile Workflow Packages

Test a server-initiated Mobile Workflow package.

1. In the Mobile Workflow Forms editor, open the Mobile Workflow form, `<workflow_form>.xbw`.
2. Click **Flow Design**.
3. Right-click in the editor, and select **Send a notification**.
4. In the Send a Notification window:
  - a) Select the Unwired Server profile and click **Get Device Users**.
  - b) Choose the desired user and fill in the fields according to the matching rules specified when creating the Mobile Workflow form.
5. Click **Send**.
6. On the client, from the applications screen, open SUPWorkflows.
7. In the client application, click **WorkFlow Messages**. This contains the server-initiated Mobile Workflow form.



# Configuring the Mobile Workflow Package in Sybase Control Center

After developing and generating the Mobile Workflow package, use Sybase Control Center to deploy the Mobile Workflow package, and to perform the configuration tasks required to make it available to application users on a messaging device.

The Workflows node in Sybase Control Center allows administrators to view deployed Mobile Workflow packages, including Mobile Workflow display name, module name, and module version. Administrators deploy Mobile Workflow packages into the Unwired Platform cluster through this node, as well as manage e-mail settings configuration.

## Deploying a Mobile Workflow Package

Use the Deploy wizard to make mobile workflow packages available on the Unwired Server.

1. In the left navigation pane, click **Workflows**.
2. From the **General** tab, click **Deploy**.
3. Select the file to upload and click **Open**.  
The package is added to the list of deployed packages, which are sorted by Display Name.

### **Next**

Configure the package if you want the deployed package to have a different set of properties in a production environment.

## Registering and Reregistering Messaging Devices

Use Sybase Control Center to trigger the registration and device activation process, which allows messaging mobile business objects (MBOs) to handle messages belonging to different data sources.

---

**Note:** When using a Windows Mobile emulator to register a device in Sybase Control Center, the device ID changes each time you reinstall the device. Before reinstalling, you must delete the original device from Unwired Server. Then, reregister the device. Otherwise, the device log shows a `Wrong Device for Code` error when the device attempts to connect after registration. This problem occurs only with Windows Mobile emulator devices.

---

1. In the left navigation pane, click the **Device Users** node.
2. In the right administration pane, click the **Devices** tab.

3. Click **Register** to register a new device, or **Reregister** to update the device used of an existing device user.
4. In the **Register Device** or the **Reregister Device** dialog:
  - a) For new device registrations only, type the name of the user that will activate and register the device. For reregistrations or clones, the same name is used and cannot be changed.
  - b) Select the name of the template for initial device registration. If you have not created any templates, only **Default** appears in the list.  
The template you choose supplies initial values in the subsequent device activation fields.
5. Change the default activation field values for the template you have chosen. If you are using the default template, you must provide the server name, which is empty.  
These values are available in the General tab of the Cluster node.
  - **Server name** – the DNS name or IP address of the primary Unwired Server, such as "myserver.mycompany.com". If using Relay Server, the server name is the IP address or fully qualified name of the Relay Server host.
  - **Port** – the port used for messaging connections between the device and Unwired Server. If using Relay Server, this is the Relay Server port. Default: 5001.
  - **Farm ID** – a string associated with the Relay Server farm ID. Can contain only letters A – Z (uppercase or lowercase), numbers 0 – 9, or a combination of both. Default: 0.  

---

**Note:** If the device uses Relay Server to connect to Unwired Server, the farm ID should be the name of the Unwired Server farm configured in the Relay Server for messaging-based synchronization applications. If the device connects to Unwired Server directly, the farm ID should be 0.

---
  - **Activation code length** – the number of characters in the activation code. If you are reregistering or cloning a device, this value cannot be changed.
  - **Activation expiration** – the number of hours the activation code is valid.
6. (Optional) Select the check box adjacent to **Activation Code** to enter the code sent to the user in the activation e-mail. This value can contain letters A – Z (uppercase or lowercase), numbers 0 – 9, or a combination of both. Acceptable range: 1 to 10 characters.  
If the activation code is automatically generated, the code for the device can be retrieved from the **Connections** group of the **Device Properties** dialog
7. Click **OK**.

## Assigning and Unassigning Device Users from the Workflow Node

---

Assign mobile workflow packages to make them available to an activation user. You can also unassign mobile workflow packages at any time.

1. In the left navigation pane, click **Workflows > MyWorkflow**.
2. In the right administration pane, click the **Devices** tab.
3. Locate the device to assign a mobile workflow package to, then:
  - a) Click **Assign Workflow**.
  - b) List the activation users to assign the mobile workflow package to.

By default, no users are listed in this window. Search for users by selecting the user property you want to search on, then selecting the string to match against. Click **Go** to display the users.
  - c) Click **OK**.
4. To unassign a mobile workflow package, select the Activation User Name and click **Unassign Workflow**.

## Enabling and Configuring the Notification Mailbox

---

Configure the notification mailbox settings that allow Unwired Server to transform e-mail messages into mobile workflows.

The notification mailbox configuration uses a listener to scan all incoming e-mail messages delivered to the particular inbox specified during configuration. When the listener identifies an e-mail message that matches the rules specified by the administrator, it sends the message as a mobile workflow to the device that matches the rule.

---

**Note:** Saving changes to the notification mailbox configuration deletes all e-mail messages from the account. Before proceeding with configuration changes, consult your e-mail administrator if you want to back up the existing messages in the configured account.

---

1. In the left navigation pane, click **Workflows**.
2. In the right administration pane, click **Notification Mailbox**.
3. Select **Enable**.
4. Configure these properties:
  - **Protocol** – choose between POP3 or IMAP, depending on the e-mail server used.
  - **Use SSL** – encrypt the connection between Unwired Server and the e-mail server in your environment.
  - **Server** and **Port** – configure these connection properties so Unwired Server can connect to the e-mail server in your environment. The defaults are localhost and port 110 (unencrypted) or 995 (encrypted).
  - **User name** and **Password** – configure these login properties so Unwired Server can log in with a valid e-mail user identity.
  - **Truncation limit** – specify the maximum number of characters taken from the body text of the original e-mail message, and downloaded to the client during synchronization. If the body exceeds this number of characters, the listener truncates

## Configuring the Mobile Workflow Package in Sybase Control Center

the body text to the number of specified characters before distributing it. The default is 5000 characters.

- **Poll seconds** – the number of seconds the listener sleeps between polls. During each poll, the listener checks the master inbox for new e-mail messages to process. The default is 60 seconds.
5. If you have added at least one distribution rule, you can click **Test** to test your configuration. If the test is successful, click **Save**.

# Troubleshoot

Use troubleshooting tips to isolate and resolve common issues.

See *Troubleshooting Sybase Unwired Platform* for information about troubleshooting issues for Workflow package and other Sybase Unwired Platform components.

## HTTP Error Codes

Unwired Server examines the EIS code received in a server response message and maps it to a logical HTTP error code, if a corresponding error code exists. If no corresponding code exists, the 500 code is assigned to signify either a Sybase Unwired Platform internal error, or an unrecognized EIS error. The EIS code and HTTP error code values are stored in log records.

The following is a list of recoverable and non-recoverable error codes. Beginning with Unwired Platform version 1.5.5, all error codes that are not explicitly considered recoverable are now considered unrecoverable.

**Table 7. Recoverable Error Codes**

| Error Code | Probable Cause                                    |
|------------|---|
| 409        | Backend EIS is deadlocked.                        |
| 503        | Backend EIS down or the connection is terminated. |

**Table 8. Non-recoverable Error Codes**

| Error Code | Probable Cause   | Manual Recovery Action                                       |
|------------|--|--|
| 401        | Backend EIS credentials wrong.   | Change the connection information, or backend user password. |
| 403        | User authorization failed on Unwired Server due to role constraints (applicable only for MBS). | N/A  |
| 404        | Resource (table/webservice/BA-PI) not found on Backend EIS.                                    | Restore the EIS configuration.                               |
| 405        | Invalid license for the client (applicable only for MBS).                                      | N/A  |
| 412        | Backend EIS threw a constraint exception.  | Delete the conflicting entry in the EIS.                     |

| Error Code | Probable Cause                                 | Manual Recovery Action |
|------------|--|------------------------|
| 500        | SUP internal error in modifying the CDB cache. | N/A                    |

Beginning with Unwired Platform version 1.5.5, error code 401 is no longer treated as a simple recoverable error. If the `SupThrowCredentialRequestOn401Error` context variable is set to true (which is the default), error code 401 throws a `CredentialRequestException`, which sends a credential request notification to the user's inbox. You can change this default behavior by modifying the value of the `SupThrowCredentialRequestOn401Error` context variable in Sybase Control Center. If `SupThrowCredentialRequestOn401Error` is set to false, error code 401 is treated as a normal recoverable exception.

## Recovering from EIS Errors

---

After sending a JSON request to Unwired Server, if you receive in the response log message an EIS code which is recoverable, the mobile workflow client throws a `TransformRetryException` or `ResponseRetryException`, as is appropriate.

A retry attempt is made after a retry time interval, which is set by default to 15 minutes. You can configure the retry time interval by setting the `SupRecoverableErrorRetryTimeout` (default: 15 minutes) and `SupUnrecoverableErrorRetryTimeout` context variables through the Sybase Control Center admin console.

Only certain error codes are considered to be recoverable.

**Table 9. Recoverable Error Codes**

| Error Code | Probable Cause                                    |
|------------|---|
| 409        | Backend EIS is deadlocked.                        |
| 503        | Backend EIS down or the connection is terminated. |

---

**Note:** If the problem with the EIS is not corrected, the retry process can continue indefinitely. Ensure that you set an appropriate retry time interval.

---

Other error codes are considered to be non-recoverable. No retry exception is thrown for these errors.



**Table 10. Non-recoverable Error Codes**

| Error Code | Probable Cause  | Manual Recovery Action                                       |
|------------|---|--|
| 401        | Backend EIS credentials wrong.  | Change the connection information, or backend user password. |
| 403        | User authorization failed on Un-wired Server due to role constraints (applicable only for MBS). | N/A  |
| 404        | Resource (table/webservice/BA-PI) not found on Backend EIS.                                     | Restore the EIS configuration.                               |
| 405        | Invalid license for the client (applicable only for MBS).                                       | N/A  |
| 412        | Backend EIS threw a constraint exception.   | Delete the conflicting entry in the EIS.                     |
| 500        | SUP internal error in modifying the CDB cache.  | N/A  |

## Mapping of EIS Codes to Logical HTTP Error Codes

The following is a list of SAP error codes mapped to HTTP error codes. SAP error codes which are not listed map by default to HTTP error code 500.

**Table 11. Mapping of SAP error codes to HTTP error codes**

| Constant                | Description  | HTTP Error Code |
|-------------------------|--|-----------------|
| JCO_ERROR_COMMUNICATION | Exception caused by network problems, such as connection breakdowns, gateway problems, or inavailability of the remote SAP system. | 503             |
| JCO_ERROR_LOGON_FAILURE | Authorization failures during the logon phase usually caused by unknown username, wrong password, or invalid certificates.         | 401             |
| JCO_ERROR_RESOURCE      | Indicates that JCO has run out of resources such as connections in a connection pool   | 503             |

| Constant             | Description                                    | HTTP Error Code |
|----------------------|--|-----------------|
| JCO_ERROR_STATE_BUSY | The remote SAP system is busy. Try again later | 503             |

## Credentials Are Lost after User Successfully Passes Activation Screen

---

User logs in successfully on Activation screen, but is no longer logged in at some point after that.

This happens when you do not execute a Save from the Activation screen, and then execute a Cancel on a subsequent screen, before a Save is executed.

Always execute a Save immediately after credentials are successfully validated on the Activation screen.

## Mobile Workflow Exception Handling

---

Describes how to handle a blocked mobile workflow..

If a mobile workflow is not received or processed on a device, this may indicate the mobile workflow is blocked in the message queue. By default, Unwired Server attempts to retry an action every fifteen minutes, up to three days.

This typically indicates that a workflow operation failed with a recoverable or unrecoverable error. To resolve the situation:

1. Check the Unwired Server and workflow client trace logs for information.
  - Unwired Server logs:
    - Aggregated Unwired Server logs – `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\UnwiredServer\logs`
    - Messaging service log details – `<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\MessagingServer\Data\Log`  
`<UnwiredPlatform_InstallDir>\UnwiredPlatform\Servers\MessagingServer\Trace`
  - Workflow client trace logs:
    - Single node – `<UnwiredPlatform_InstallerDir>\UnwiredPlatform\Servers\MessagingServer\Data\ClientTrace`
    - Cluster CDB node – `<UnwiredPlatform_InstallerDir>\Data\Messaging\ClientTrace`
2. Use information in the logs to resolve the problem.

3. Use Sybase Control Center to check message queue status, and to suspend, resume, or remove items in the queue. See:  
*Sybase Control Center for Sybase Unwired Platform > Manage > Managing Unwired Platform > Routine System Maintenance Tasks > Checking System Status > Checking Mobile Workflow Users and Queues*



# Index

## A

Activate starting point 29  
 activating devices 93  
 adding controls to the screen 50

## C

checkbox properties 60  
 choice item  
   adding 65  
   deleting 66  
   editing 65  
 client-initiated 9  
 Client-initiated starting point 29  
 communication style  
   asynchronous 57  
   synchronous 57  
 complex data type 49  
 configuring control properties 60  
 connection properties 72  
 Connections 29  
 Connections,  
   Goto 29  
   Listview details 29  
   Operation error 29  
   Operation success 29  
 controls 6, 51  
 Controls  
   listview  
     properties 69  
 controls, adding 50  
 creating keys 44  
 Credential request starting point 29  
 Credentials  
   dynamic 42  
   static 42

## D

Deploying  
   a Mobile Workflow package 82  
   to iPhone simulator 82  
 device platforms 5  
 device users  
   assigning mobile workflow packages 94

documentation roadmap  
   document descriptions 1  
 Dynamic authentication 43

## E

EIS error codes 97–99  
 Empty screen 29  
 encoding, UTF-8 74  
 error codes  
   EIS 97–99  
   HTTP 97–99  
   mapping of SAP error codes 99  
   non-recoverable 97, 98  
   recoverable 97, 98

## F

Flow Design  
   general properties 25  
   localization 25  
   matching rules 25  
   page properties 25  
   rulers and grid 25

## G

Goto connection 29

## H

HTTP error codes 97–99

## I

installing  
   App Store 81  
   iPhone 82

## K

keys 6  
 Keys 45  
 keys, creating 44

**L**

- listview
  - creating a field 70
- listview control 49
- Listview details connection 29
- Listview menu item 58
- Localization 25, 73
  - current locale 76
  - locale properties file, validating 77
  - updating the current locale 76
  - validating the locale properties file 77

**M**

- manually adding a starting point 36
- matching rules, adding 41
- MBOs
  - overview 10
- menu item, adding 51
- menu item, creating 55
- menu items 6
- messaging device registration 93
- messaging device setup 93
- mobile business object relationships 49
- Mobile Workflow Application Designer
  - checkbox 31
  - choice 31
  - editbox 31
  - htmlview 31
  - listview 31
  - menu item 31
    - Screen Design palette options
      - checkbox 31
      - choice 31
      - editbox 31
      - htmlview 31
      - listview 31
      - menu item 31
      - signature 31
      - slider 31
    - signature 31
    - slider 31
- Mobile Workflow Application Generation wizard 77
- Mobile Workflow form, designing 23
- Mobile Workflow Forms editor 23
- Mobile Workflow Forms Editor 23
  - Flow Design palette options 29

- Mobile Workflow package
  - overview 6
- mobile workflow packages
  - assigning device users 94
  - configuring notification mailbox 95
  - deploying 93

**N**

- navigations 6
- notification extraction rules 40
- notification extraction rules, editing 40
- notification mailbox 95
- Notification Mailbox 35
- notification, matching rules 41
- notification, sending a test 35

**O**

- Operation error connection 29
- Operation success connection 29

**P**

- parameter mappings 59
- properties
  - server-initiated 38
- Properties
  - Flow Design page 25

**R**

- registering messaging devices 93

**S**

- Screen Design
  - page properties 30
- Screen, empty 29
- screens 6
- sending a test notification 35
- server-initiated 9
- server-initiated properties 38
- server-initiated starting point 37
- Server-initiated starting point 29
- Settings
  - Connection info 82

- iPhone 82
- signature control 67
- starting point
  - adding manually 36
- starting points
  - activate 6
    - Activate 35
    - client-initiated 6
    - Client-initiated 35
    - credential request 6
    - Credential request 35
  - editing 41
  - server-initiated 6, 37, 41
  - Server-initiated 35
- Starting points
  - Activate 29
  - Client-initiated 29
  - Credential request 29
  - Server-initiated 29
- static authentication 42

- submit menu item 56
- Sybase Mobile Workflow Forms Editor
  - accessing 24
  - creating a new 24
- Sybase Unwired Platform
  - architectural overview 5

## T

- testing a notification 35

## U

- UTF-8 encoding 74

## W

- WorkflowContainer-iPhone.zip 82

