



**SAP Mobile WorkSpace: Hybrid App
Package Development**

SAP Mobile Platform 2.3

DOCUMENT ID: DC01941-01-0230-01

LAST REVISED: February 2013

Copyright © 2013 by Sybase, Inc. All rights reserved.

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

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

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

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

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

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

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

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

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

Contents

Developing a Hybrid App Package	1
Hybrid App Designer	1
Setting Hybrid App Designer Preferences	2
Hybrid App Designer Preferences	6
Creating a New Hybrid App Screen Flow	10
Creating a Screen From a Data Source	11
Adding Images to the Project	12
Hybrid App Designer Context Menus	14
Outline View	15
Flow Designs	16
Starting Points	16
Flow Design Application Properties	29
Flow Design Palette Options	35
Connections	36
Keys	38
Defining and Using Keys	38
Key Properties	39
Navigating Key References	40
Screen Designs	41
Screen Design Page Properties	42
Adding and Configuring Objects from the Palette	44
Previewing Screens	44
Hybrid App Controls	46
Checkbox Properties	46
EditBox Properties	47
Choice Control	50
Slider Properties	54
Signature Properties	55
HtmlView Properties	56
Listview Control	57

AttachmentViewer Control	62
Button Control	67
Label Properties	77
Link Properties	78
Image Properties	78
Collapsible Properties	79
Accordion Properties	80
Switch Properties	81
MAChartView Control	82
Checkbox Group Properties	90
Menu Items and Custom Actions	90
Creating a Save, Open, Close, or Cancel Menu Item	91
Creating an Online Request Menu Item or Custom Action	91
Creating a Submit Hybrid App Menu Item or Custom Action	92
Defining Code for a Custom Menu Item or Custom Action	92
Menu Item and Custom Action Properties	93
Localization and Internationalization	101
Localization Limitations	101
Localizing a Hybrid App Package	102
Changing the Encoding Type	103
Creating and Validating a New Locale Properties File	104
Editing the Locale Properties File	107
Deploying the Mobile Application Project	107
Configuring the Device for Localization	108
Removing a Locale	110
Updating the Current Locale	111
Hybrid App Package Internationalization	111
Internationalization on the Device	114
Sending a Test Notification	117

Generating Hybrid App Files and Deploying a Package	119
Hybrid App Generation Wizard Options	119
Deployment Modes	121
Generated Hybrid App Files	122
Default Look and Feel	123
Optimize for Performance Look and Feel	126
Using the BlackBerry Custom Look and Feel	127
Customizing Generated Code	130
Activating the Hybrid App	131
Generating Files for an Externally Developed Hybrid	
App	133
Server-initiated Notification Configuration	134
Troubleshoot	137
Index	139

Developing a Hybrid App Package

Hybrid App Designer lets you create or modify Hybrid App packages on different device platforms independently of native device development.

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

A Hybrid App requires an integration module on the server side, which processes Hybrid App-specific metadata to map keys to and from mobile business object attributes, personalization keys, and parameters. The 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 Hybrid App Designer allows you to create the forms that can be assigned to the devices that have the Hybrid Web Container installed and that are registered with SAP® Mobile Server through SAP Control Center.

When you use the Hybrid App Designer, 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 Hybrid App Designer allows you to develop Hybrid App screens that can call the create, update, delete, and other operations, as well as object queries of a mobile business object.

You can generate a Hybrid App that works on these platforms:

- Apple iOS
- Android
- BlackBerry
- Windows Mobile Professional

See *Supported Hardware and Software* for supported version levels.

When designing the user interface for Hybrid App screen flows for mobile devices, keep it simple. The Hybrid App screen flow should deliver clear, concise content with minimal navigation menus. The Hybrid App 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 Hybrid App screen flow.

Hybrid App Designer

The Hybrid App Designer allows you to create Hybrid App forms that can be deployed to devices and registered in SAP Control Center.

The Hybrid App Designer contains these tabbed pages:

- **Introduction** – by default, opens when you create a new Hybrid App or double-click an existing one. To hide this page, click **Window > Preferences**, navigate to **SAP AG > Mobile Development > Hybrid App Designer**, and unselect the option, **Show the Introduction page when opening the designer**.
- **Flow Design** – displays an empty canvas if you did not choose any of starting points in the New Hybrid App Designer. If you selected a starting point when creating the new Hybrid App 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 Hybrid App. The Palette view provides menu items and controls that you can add to your screen.
- **Preview screens** – optionally, display previews of how your screen design will look on specific devices. To see the preview tabs, enable their display in Preferences.

Separate from the Designer, the Outline view provides an expandable tree that you can use to quickly view and navigate the components of your Hybrid App. It also includes a thumbnail view of either the Screen Design or Flow Design.

See also

- *Creating a New Hybrid App Screen Flow* on page 10
- *Outline View* on page 15
- *Flow Designs* on page 16
- *Keys* on page 38
- *Screen Designs* on page 41
- *Previewing Screens* on page 44
- *Hybrid App Controls* on page 46
- *Menu Items and Custom Actions* on page 90

Setting Hybrid App Designer Preferences

Use the Preferences dialog to set the default appearance behavior of Hybrid App Designer elements.

Set preferences to specify default settings for Hybrid App Designer elements. By contrast, when you select an object in Designer and modify its properties in the Properties view, you override preferences only for that instance: preferences are unchanged for other instances.

1. In SAP Mobile WorkSpace, select **Window > Preferences**.
2. In the Preferences dialog, select **SAP AG > Mobile Development > Hybrid App Designer**.

- Set preferences for the Hybrid App Designer.
- Select **Screen Design Page** to set preferences for the Screen Design page.

3. Click **Apply**.

4. Click **OK**.

See also

- *Hybrid App Designer Preferences* on page 3

Hybrid App Designer Preferences

Settings for the default appearance and behavior of Hybrid App Designer elements.

Preferences set default properties for Hybrid App Designer pages. When you create a new Designer form, the default properties are inherited. During a session, you can modify the properties of any page. When you do so, you modify only that object instance — preferences are unchanged and default settings are still used for any new instance.

Hybrid App Designer Preferences

You can set the following preferences in the main Hybrid App Designer page:

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

See also

- *Setting Hybrid App Designer Preferences* on page 2

Flow Design Page Preferences

Settings for Flow Design pages of the Hybrid App Designer.

The page includes these global options, all selected by default:

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

- Show locale properties file validation dialog

Appearance Preferences

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

Connections Preferences

Select the default line style for new connections:

- Oblique (default) – the connection lines are not aligned with the horizontal or vertical axis and you can create breakpoints anywhere on the connection line. If you drag the connection line, a new breakpoint is created at the drag point and two lines are created.
- Rectilinear – the connection line always remains either horizontal or vertical.

Printing Preferences

Set default preferences for printing Designer pages:

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

Rulers and Grid Preferences

Rulers and grids help you position and size objects on the Flow Design page.

Section	Description
Ruler options	Default ruler settings for new design pages: • Show rulers for new diagram • Ruler units Show rulers for new diagram displays rulers in the design page, then select the ruler units.
Grid options	Default grid settings for new design pages: • Show grid for new diagrams • Snap to grid for new diagrams • Snap to shapes for new diagrams • Grid spacing (in inches)

Screen Design Page Preferences

Settings for Flow Design pages of the Hybrid App Designer.

The page includes these global options:

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

Appearance Preferences

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

Printing Preferences

Set default preferences for printing Designer pages:

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

Rulers and Grid Preferences

Rulers and grids help you position and size objects on the Screen Design page.

Section	Description
Ruler options	Default ruler settings for new design pages: • Show rulers for new diagram • Ruler units Show rulers for new diagram displays rulers in the design page, then select the ruler units.

Section	Description
Grid options	Default grid settings for new design pages: • Show grid for new diagrams • Snap to grid for new diagrams • Snap to shapes for new diagrams • Grid spacing (in inches)

Screen Preview Preferences

Settings to enable and configure Screen Preview pages of the Hybrid App Designer.

You can preview a Hybrid App screen during its design to see approximately how it will look on specific devices, without having to generate, deploy, and run the application.

By default, the Hybrid App Designer displays only the Flow Design and Screen Design tabs. To see previews, enable their display in Preferences.

You can enable one or more of these preview types, each based on a different look and feel:

- **jQuery** – emulates the look of WebKit-based browser engines. Available emulators include iOS, Android devices, and BlackBerry 6 and 7.
- **Non-jQuery** – emulates devices that are less sophisticated than Webkit, including the BlackBerry 5 devices.
- **Performance** – emulates simple devices like Windows Mobile, or the “performance” look of other devices.

For each preview type, you can select any of the available emulators and set the default orientation to either `Portrait` or `Landscape`.

To include ListViews in your previews, select **Enable the automatic generation of synthetic preview data**.

Hybrid App Designer Preferences

Settings for the default appearance and behavior of Hybrid App Designer elements.

Preferences set default properties for Hybrid App Designer pages. When you create a new Designer form, the default properties are inherited. During a session, you can modify the properties of any page. When you do so, you modify only that object instance — preferences are unchanged and default settings are still used for any new instance.

Hybrid App Designer Preferences

You can set the following preferences in the main Hybrid App Designer page:

- **Optimize for performance** provides a more basic look and feel. Raw HTML controls are used to the extent possible (for example, there is no signature or slider control in HTML),

with little to no CSS stylization or JavaScript enhancements to customize the look and feel. This option is always used for Windows Mobile Professional platforms.

- **Optimize for appearance** provides a more sophisticated, or native, look and feel, but may affect performance. This option is not available on Windows Mobile platforms.
- **Show the Introduction page when opening the designer** controls whether the Introduction page is visible or hidden. If this option is enabled, the Designer opens to the Introduction page, which provides basic information and help.

See also

- *Setting Hybrid App Designer Preferences* on page 2

Flow Design Page Preferences

Settings for Flow Design pages of the Hybrid App Designer.

The page includes these global options, all selected by default:

- Show connector handles
- Show pop-up bars
- Enable animated layout
- Enable animated zoom
- Enable anti-aliasing
- Show status line
- Show locale properties file validation dialog

Appearance Preferences

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

Connections Preferences

Select the default line style for new connections:

- Oblique (default) – the connection lines are not aligned with the horizontal or vertical axis and you can create breakpoints anywhere on the connection line. If you drag the connection line, a new breakpoint is created at the drag point and two lines are created.
- Rectilinear – the connection line always remains either horizontal or vertical.

Printing Preferences

Set default preferences for printing Designer pages:

Section	Description
Orientation	Either portrait or landscape.
Units	Either inches or millimeters.

Section	Description
Size	Choose one of the available print sizes. Optionally, modify the displayed width and height of the selected size.
Margins	Enter a value, in inches, for the margins.

Rulers and Grid Preferences

Rulers and grids help you position and size objects on the Flow Design page.

Section	Description
Ruler options	Default ruler settings for new design pages: • Show rulers for new diagram • Ruler units Show rulers for new diagram displays rulers in the design page, then select the ruler units.
Grid options	Default grid settings for new design pages: • Show grid for new diagrams • Snap to grid for new diagrams • Snap to shapes for new diagrams • Grid spacing (in inches)

Screen Design Page Preferences

Settings for Flow Design pages of the Hybrid App Designer.

The page includes these global options:

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

Appearance Preferences

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

Printing Preferences

Set default preferences for printing Designer pages:

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

Rulers and Grid Preferences

Rulers and grids help you position and size objects on the Screen Design page.

Section	Description
Ruler options	Default ruler settings for new design pages: • Show rulers for new diagram • Ruler units Show rulers for new diagram displays rulers in the design page, then select the ruler units.
Grid options	Default grid settings for new design pages: • Show grid for new diagrams • Snap to grid for new diagrams • Snap to shapes for new diagrams • Grid spacing (in inches)

Screen Preview Preferences

Settings to enable and configure Screen Preview pages of the Hybrid App Designer.

You can preview a Hybrid App screen during its design to see approximately how it will look on specific devices, without having to generate, deploy, and run the application.

By default, the Hybrid App Designer displays only the Flow Design and Screen Design tabs. To see previews, enable their display in Preferences.

You can enable one or more of these preview types, each based on a different look and feel:

- **jQuery** – emulates the look of WebKit-based browser engines. Available emulators include iOS, Android devices, and BlackBerry 6 and 7.
- **Non-jQuery** – emulates devices that are less sophisticated than Webkit, including the BlackBerry 5 devices.
- **Performance** – emulates simple devices like Windows Mobile, or the “performance” look of other devices.

For each preview type, you can select any of the available emulators and set the default orientation to either `Portrait` or `Landscape`.

To include `ListView`s in your previews, select **Enable the automatic generation of synthetic preview data**.

Creating a New Hybrid App Screen Flow

Use the Hybrid App Designer wizard to define a screen flow.

Prerequisites

You must have an existing project in which to create the Hybrid App screen flow.

Task

1. In the Mobile Development perspective, select **File > New > Hybrid App Designer**.
2. Follow the instructions in the Hybrid App Designer wizard:
 - **Enter or select the parent folder** – select the Hybrid App project in which to create the Hybrid App screen flow.
 - **File name** – enter a name for the Hybrid App screen flow. The extension for Hybrid App screen flows is `.xbw`.
 - **Advanced** – link the Hybrid App screen flow to an existing file in the file system.
 - **Link to file in the file system** – click **Browse** to locate the file to which to link the Hybrid App screen flow. Linked resources are files or folders that are stored in the file system outside of the project's location. If you link a resource to an editor, when you select the editor, the resource is selected in the WorkSpace Navigator. Conversely, when you select the resource in the WorkSpace Navigator, the editor is selected.

Click **Variables** to define a new path variable. Path variables specify locations on the file system.
3. Click **Next**.
4. In the Starting Points page, select the ways the Hybrid App starts:
 - **Can be started, on demand, from the client** – creates a Client-initiated starting point. This allows the user to start the Hybrid App on the device and submit data to the server.
 - **Responds to server-driven notifications** – creates a Server-initiated starting point. When a notification that matches the defined matching rules arrives in the Notification Mailbox, it is then sent as a notification to the devices that match those matching rules. When you select this option, you will set up the server-initiated starting point in the next screens of the wizard. Follow the procedure for *Creating a Server-initiated Starting Point*.

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

See also

- *Hybrid App Designer* on page 1
- *Outline View* on page 15
- *Flow Designs* on page 16
- *Keys* on page 38
- *Screen Designs* on page 41
- *Previewing Screens* on page 44
- *Hybrid App Controls* on page 46
- *Menu Items and Custom Actions* on page 90
- *Creating a Screen Flow with a Server-Initiated Starting Point* on page 17
- *Starting Point Properties* on page 22

Creating a Screen From a Data Source

Drag a data source, such as a mobile business object (MBO) or operation, onto the Flow Design to automatically create a screen with controls, keys, menus, and navigation.

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 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.

If the MBO you use 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.

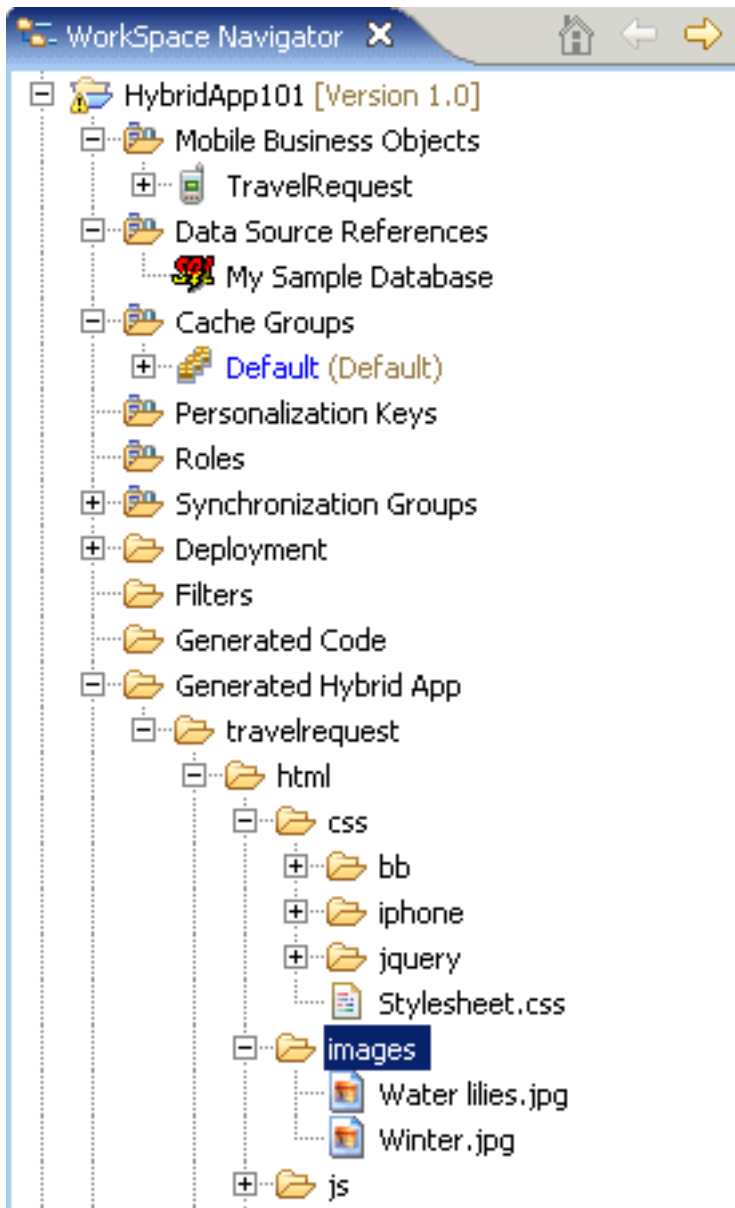
Note: Hybrid App do not support structured mobile business objects and complex parameters.

Adding Images to the Project

To use images in the Hybrid App, you must first add them to the project.

1. In WorkSpace Navigator, navigate to your project's *Generated Hybrid App* *hybrid-app-name*\html folder and create a subfolder for images.

For example:



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

See also

- *Screen Design Page Properties* on page 42

Hybrid App Designer Context Menus

Each Hybrid App Designer page except the Intro page provides quick access to appropriate actions through the right-click context menu.

Option	Description
Edit	Edit the selected object.
Delete	Delete the selected object.
Mobile Business Object References	Shows current mobile business object references and allows you to edit or delete existing mobile business object references. Note: This menu item appears only on the Flow Design page.
Generate Hybrid App Package	Generate the Hybrid App screen flow code.
Send a notification	Send an e-mail from the Hybrid App Designer to a device user. This can be used to quickly test e-mail triggered applications. The test e-mail bypasses the Notification Mailbox. Note: The Cc and Bcc fields are used only for matching rules. An e-mail is not sent to the users entered in these fields.
Select	Select: - All – selects all objects on the page. - All shapes – selects all shapes on the page. - All connections – selects all connections on the page.
Arrange all	Arranges the shapes on the page.
View	View: - Grid - Rulers - Page breaks - Recalculate page breaks - Snap to grid
Zoom	Select from the zoom options.
Show properties view	Shows the Properties view for the selected object.

Outline View

Provides an alternate way to view and navigate the components of your Hybrid App.

By default, the Outline view is located in a tab next to the Enterprise Explorer view.

The Outline view toolbar contains two icons that you can click to toggle between two view types:

- **Outline** – a tree representation of the components of your Hybrid App.
- **Overview** – a static thumbnail image of the current Flow Design or Screen Design page. The image changes when you open either page.

The Outline provides a dynamic tree that you can use not only to view your application components, but navigate to them. The tree is organized hierarchically:

- The root nodes represent starting points, listed by type (Activate, Credential Request, and so on).
- Under each starting point are the screens that flow from the point. (For this reason, a screen can appear more than once in the tree).
- Under each screen are component categories (Controls, Keys, Menu, and so on). Each category node shows a counter for the number of components in the screen.
- Under each component category are the individual components in the screen. Custom actions and menu item nodes show the action type.

Navigation actions from the Outline vary depending on which Designer page is currently open:

- Click an item in the Outline to select the corresponding component in the current Screen Design or Flow Design and open its properties view.
With the Screen Design page open, this action selects the component. With the Flow Design page open, selecting a component within a screen displays its parent screen.
For example, to navigate to a particular key, you can select the key in the Outline with the Screen Designer open. The Properties view opens to the Keys tab and selects the key in the list.
- Double-click an item in the Outline to select the corresponding component in the Screen Design. If the Flow Design page is currently open, the Designer switches to the Screen Design page.
This can be useful if the Flow Design page is currently open and you want to navigate to a component within a screen.
- To see a key's locations in the application, right-click the key in the outline and choose **Usage** to open Key References browser.

See also

- *Hybrid App Designer* on page 1

- *Creating a New Hybrid App Screen Flow* on page 10
- *Flow Designs* on page 16
- *Keys* on page 38
- *Screen Designs* on page 41
- *Previewing Screens* on page 44
- *Hybrid App Controls* on page 46
- *Menu Items and Custom Actions* on page 90

Flow Designs

The flow design of your Hybrid App describes starting points and screens and relationships between them.

The Flow Design view in the Hybrid App Designer provides a graphical interface for defining the logical flow of your Hybrid App.

See also

- *Hybrid App Designer* on page 1
- *Creating a New Hybrid App Screen Flow* on page 10
- *Outline View* on page 15
- *Keys* on page 38
- *Screen Designs* on page 41
- *Previewing Screens* on page 44
- *Hybrid App Controls* on page 46
- *Menu Items and Custom Actions* on page 90

Starting Points

Starting points specify which screen the user sees when they perform a certain action. Every screen must be part of a flow that connects to a starting point.

Starting points include:

- Client-initiated – this starting point is activated when the user opens the Hybrid App on the device.
- Activate – this is the starting point that is activated when the user first opens the Hybrid App. It does not activate on the second and subsequent times the application is started. Multiple Hybrid Apps can share the same Activate starting point. Different Hybrid Apps can specify an activation key, and Hybrid Apps with the same activation key share their activation status. For example, if Hybrid App A and Hybrid App B both specify an activation key of AB (via the key attribute on the RequiresActivation element), when Hybrid App A gets activated, it also activates Hybrid App B such that when Hybrid App B is invoked for the very first time, its activation screen will not be displayed.

Note: A Hybrid App with an Activate starting point is considered to be successfully activated only if it is closed with a Submit Hybrid App menu item.

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

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

Creating a Screen Flow with a Server-Initiated Starting Point

Use the Hybrid App Designer wizard to define a screen flow with a starting point that is activated when a notification message is sent to a device identified by matching rules in the Hybrid App.

Note: Each Hybrid App package can have only one server-initiated starting point.

1. In the Mobile Development perspective, select **File > New > Hybrid App Designer**.
2. Follow the instructions in the Hybrid App Designer wizard:
 - **Enter or select the parent folder** – select the Hybrid App project in which to create the Hybrid App screen flow.
 - **File name** – enter a name for the Hybrid App screen flow. The extension for Hybrid App screen flows is .xbw.
 - **Advanced** – link the Hybrid App screen flow to an existing file in the file system.
 - **Link to file in the file system** – click **Browse** to locate the file to which to link the Hybrid App screen flow. Linked resources are files or folders that are stored in the file system outside of the project's location. If you link a resource to an editor, when you select the editor, the resource is selected in the WorkSpace Navigator. Conversely, when you select the resource in the WorkSpace Navigator, the editor is selected.

Click **Variables** to define a new path variable. Path variables specify locations on the file system.
3. Click **Next**.
4. In the Starting Points page, select **Responds to server-driven notifications** and click **Next**.
5. In the Select a Mobile Business Object and Object Query page, click **Search**.
6. In the Search for Mobile Business Object page:
 - a) Click **Search**.
 - b) Accept the current project or choose a different one.

- c) Click **Search**.
 - d) Select a Mobile Application Project from the list and click **OK**.
7. (Optional) In the Select a Mobile Business Object and Object Query page, select an object query and click **Next**.
- Consider using an object query if you want the Hybrid App message to use data from the back end or from the SAP Mobile Platform cache that is not available in the originating notification.
In this scenario, you can use extraction rules to get input parameters for the object query from the originating notification. Then, you can get the data values for the workflow message from the query result set.
 - You do not need an object query if all of the data values for the workflow message are available in the originating notification. In this scenario, you can use extraction rules to obtain the data values directly from the originating notification.

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

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

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

- **Field** – select the field from the notification message from which the parameter receives data, for example, Subject. Selecting Subject indicates that the parameter receives data from the subject line of the notification message.
- **Start tag** – enter or select the text for the start tag regular expression, for example, `"Approval Request\"`.
- **End tag** – enter or select the text for the end tag regular expression, for example, `"\) was sent"`.
- **Format** – specify a format that supports locale-specific parameter extraction. For example, you can indicate that a given date parameter will be in the form of yyyy-MM-dd in the e-mail, and another date parameter will be in the form of yyyy-dd-MM. This applies to the DateTime type.
- **Sample notification field contents** – enter the value of the field previously specified on the Specify Sample Notification page of the Notification Processing wizard.

- **Value extracted from sample notification field contents** – specify the value the start tag and end tag matching rules, given the sample field value.

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

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

See also

- *Creating a New Hybrid App Screen Flow* on page 10
- *Starting Point Properties* on page 22

Notification Extraction Rules

Notification extractions rules are used to extract parameter values to keys, so you can then map the parameters to objects, such as keys.

This means that you can extract values from the notification without using 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 Hybrid App to submit a create/read/update/delete (CRUD) operation or object query.

Define the notification extraction rules when you create a server initiated starting point. 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.

Some e-mail clients allow you to specify the **To:** and **From:** fields without quotes (for example, John Doe <john.doe@sap.com>). The actual value received by the SAP Mobile Server may have quotes in it (for example, "John Doe" <john.doe@sap.com>), which any extraction rules that act on those fields must consider.

Adding a Starting Point to an Existing Screen Flow

Drop a starting point from the palette into your Flow Design page.

This topic describes how to create a starting point manually (for example, if you did not use the Hybrid App Designer wizard to automatically create it.)

1. From the palette on the Flow Design page, select a starting point, then click on the Flow Design canvas.
2. If you select:
 - the activate, credential request, or client-initiated type, go to the next step.
 - the server-initiated type, the Notification Processing wizard opens. Optionally, complete the steps to configure the object query and notification message to be used. Click **Finish** to create the starting point.

Note: When you manually create a server-initiated starting point that invokes an object query, the list key has no children, so you must manually create them.

Create the server-initiated starting point using the Hybrid App Designer wizard, or by dragging and dropping the mobile business object onto the Flow Design page.

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

Implementing a Conditional Start Screen

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

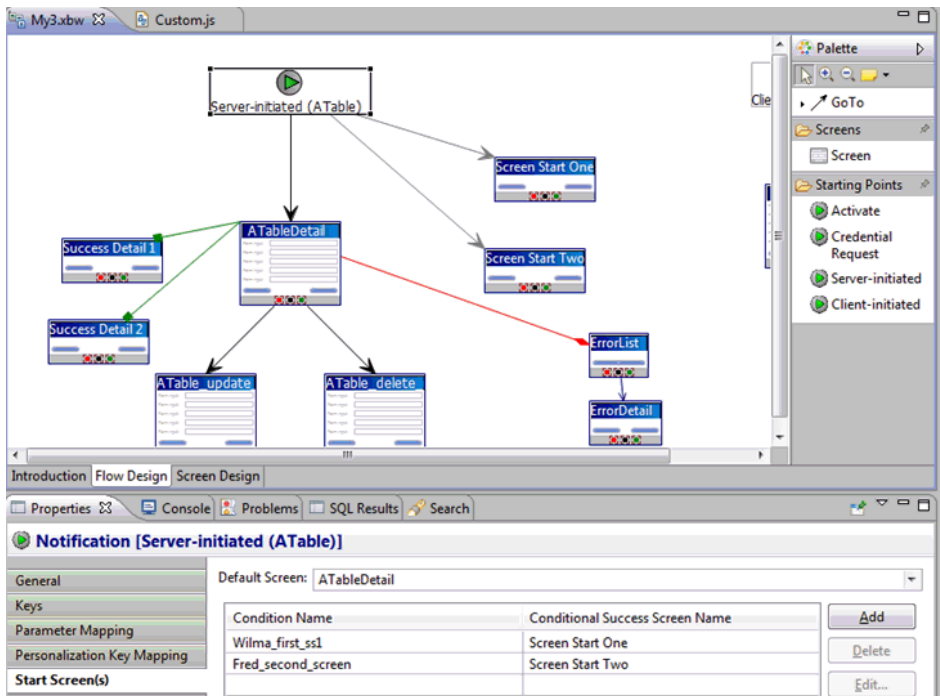
Like the conditional success navigation feature, there is a table of condition names with the matching Start screen. If all of the conditions are evaluated as false (or if they are absent), the default navigation is executed.

1. In the Flow Design page, select the server-initiated starting point to see the Properties.
2. In the Properties view, click **Start Screen(s)**.
3. Click **Add** to add a condition.
4. In the dialog, enter the condition name, select the target screen with which to associate the condition, and click **OK**.

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

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

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



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

```
function customConditionalNavigation( currentScreenKey,
    actionName,
    defaultNextScreen,  conditionName,
    workflowMessage ) {
    if((currentScreenKey === SERVERINITIATEDFLAG) && (actionName
    === '')) {
        // conditional start screen uses this magic screen key and
        the empty action name.
        if( conditionName === 'Wilma_first_ss1') {
            // custom logic
            return true;
        }
        else if(conditionName === 'Fred_second_screen'){
            // custom logic
            // return true or false
            return false;
        }
    }
    // default case is to NOT change the flow
    return false;
}
```

6. Regenerate the Hybrid App package.

When you regenerate the Hybrid App package, the hybridapp.js file is regenerated. The conditional start screen method is shown in the hybridapp.js file similar to this:

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

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

Starting Point Properties

Properties for starting points.

The Properties view for all starting points includes the Keys page. Server-initiated starting points have the additional properties pages described in this topic.

General

Property	Description
Mobile business object	(Optional) A mobile business object with which the Server-initiated starting point is associated.
Object query	(Optional) When you define an object query in the mobile business object associated with the Hybrid App screen flow, the object query returns the specified instances of the mobile business object. For example, if a Hybrid App user submits a travel request to Hawaii, 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. You can use object queries that return multiple instances, as well as object queries that return only one instance.

Property	Description
Generate old value keys	Send two copies of data—one with the key names you select at design time, and one with the same names prefixed with “_old” (default). The application might update the keys you select at design time, but the old values are not updated unless this option is selected. The “_old” value keys can be critical when an update or parent update operation is executed later.
Jumpstart	Launch the Notification Processing wizard, where you can compose a sample notification message that specifies parameter value extraction rules and matching rules.
Extraction rules	Shows the notification extraction rules for the Server-initiated starting point. Select a rule and click Edit to change the transformation rule.
Error screen	The screen to which the user navigates if an asynchronous operation invocation fails.
Subject	If an error occurs, a notification is sent to the device with the appropriate error log keys and the specified subject. The value you specify here appears in the Subject field of the notification.
From	Text that appears in the From field of error notifications.

Keys

The Keys page lets you view and configure properties for keys that are associated with a starting point. You can sort on any property in the table by clicking its column heading.

Property	Description
Key name	The name of the associated key.
Type	The data type of the associated key, for example, string.
Data binding	The data source (if any) to which the key is bound.

The view provides buttons to perform these operations on keys:

- **New** – create a new key.
- **Edit** – edit the properties of information a selected key.
- **Remove** – delete a selected key.
- **Usage** – view and navigate to locations in the application where a selected key is used.

Parameter Mappings

Online requests and custom actions can bind parameters in Mobile Business Object (MBO) operations and object queries to keys or context data that are used in the submitted messages.

The Parameter Mappings properties are generated as metadata in `WorkflowClient.xml`, which defines all the application's action mappings.

The Parameter Mappings table shows the available parameters in MBO operations and object queries, and their current mappings. The **Edit** button opens the Parameter Mapping dialog, where you can modify the properties of a selected mapping.

For information about parameter mapping, see *WorkflowClient.xml File* in *Developer Guide: Hybrid Apps*.

Property	Generated <i>WorkflowClient.xml</i> File Metadata
Parameter name	The name of the parameter in the operation or query that contains the data.
Parameter type	The type of parameter in MBO terms.
Mapping type	The source of the data, one of these types: • Key • BackEndPassword • BackEndUser • DeviceId • DeviceName • DeviceType • UserName • MessageId • ModuleName • ModuleVersion • QueueId • ContextVariable
Key	The key whose corresponding value is used as the value for this parameter. Used only when the Mapping type is Key.
Return data to client	Includes the key data in the return message to the client. If not selected, key data from the mapping is omitted.

Personalization Key Mappings

This page lists any personalization keys. To define the value of a personalization key, select its row in the list and click **Edit**.

Personalization keys allow the mobile user to personalize specific input field values by associating a key with a value. If your mobile business object has load parameters mapped to transient personalization keys, you can specify the key values that are loaded when an operation or object query is invoked from the Hybrid App package. This can be useful, for example, in loading specific data from large data sources into a consolidated database.

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

For more information, see *Managing Personalization Keys* in *SAP Mobile WorkSpace - Mobile Business Object Development*.

Property	Description
Personalization key	The name of the personalization key.
Key	An operation argument, load argument, or synchronization parameter that provides the value of the personalization key.
Project	The project in which this personalization key is created. The personalization key appears in the <code>Personalization Key</code> folder of the selected Mobile Application project.

Start Screen(s)

This page contains a table of condition names with the matching Start screen. If there are no conditions or if all of the conditions are evaluated as false, the default navigation is executed.

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

Edit Extraction Rule Dialog

Property	Description
Field type	The field in the notification from which the parameter value is extracted, for example, <code>Subject</code> , indicating a value in the subject line of the notification message.
Tag before parameter	Text indicating what to extract from the beginning of the field's contents, for example, <code>"Approval Request\("</code> .
Tag after parameter	Text indicating what to extract from the end of the field's contents, for example, <code>"\)was sent"</code> .

Property	Description
Format	Format uses the C# <code>ParseExact</code> syntax, and applies only to <code>DateTime</code> . It supports locale-specific parameter extraction. For example, you can indicate that a given date parameter is in the form <code>yyyy-MM-dd</code> in the e-mail, and another date parameter is in the form <code>yyyy-dd-MM</code> .
Key	The key to extract from the notification value.

Note: Regular expressions support the syntax used by the C# `Match()` methods of the class `System.Text.RegularExpressions.Regex`.

See also

- *Creating a New Hybrid App Screen Flow* on page 10
- *Creating a Screen Flow with a Server-Initiated Starting Point* on page 17

Credentials

You can use either dynamic or static credentials in a Hybrid App screen flow.

See *Security* and *System Administration* for more detailed information about implementing security and certificates.

The user name and password values are required when the Hybrid App invokes a mobile business object operation. These authentication values can be provided statically (at design time), or dynamically (by the user at runtime). For requests sent by the client with a credential screen specified, requests are always invoked on the server using the credentials specified by the user, regardless of whether static or dynamic authentication is specified.

The choice of static versus dynamic authentication applies only to requests that must be executed on the server that do not have any credentials, or that do not have valid credentials. This happens when an object query needs to be run by a server-initiated notification, for example, or if the client provides incorrect credentials. In that scenario, the decision between static and dynamic becomes important. If static was chosen, it silently uses those hard-coded credentials. If dynamic was chosen, it sends a notification to the client and asks the user to supply the credentials.

For example, you might define a server-initiated Hybrid App with a credential screen and static authentication. When the notification first comes in, it runs an object query using the hard-coded credentials. This is then sent to the user, who opens the notification and then makes an online request. This online request, be it an operation or an object query, will be made using the credentials supplied by the user.

Dynamic credentials require the user to enter the user name and password on a screen that the credential request starting point references. Select **Credential Cache User Name** and **Password** to indicate the user name and password to be required on the client. When the user logs in, the credentials are authenticated using the stored credentials.

Note: If an e-mail triggered Hybrid App has dynamic cached credentials, the cached credentials are not cached between invocations of the Hybrid App form through an e-mail 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 Hybrid App can be extracted from the selected SAP Mobile Platform profile user name and password when the Hybrid App is generated, or they can be hard-coded using the Properties view. After deployment, you can change static credentials in SAP Control Center.

The application can also have a credential screen (Credential Request) that appears if the Hybrid App 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 SAP Mobile Server connection profile authentication** – specifies that the user name and password associated with the connection profile are used when code is generated for the Hybrid App. Selected by default.
 - **Use hard-coded credentials** – sets the user name and password. When you select this option, the User name and Password fields are activated.
 - **Use certificate-based credentials** – enables you to use a certificate to generate authentication credentials.
4. (Optional) If you select **Use hard-coded credentials** in the previous step, enter the **User name** and **Password** that are to be used for authentication.
5. Select **File > Save**.

Setting Up Static Authentication Using a Certificate

Set up static authentication credentials generated from a certificate.

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

5. Enter a password and select an alias, then click **OK**.

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

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

6. Select **File > Save**.

Setting Up Dynamic Authentication

Use dynamic authentication to enable 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 Hybrid App, 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 Hybrid App.

1. In the Mobile Development perspective, select **File > New > Hybrid App Designer**.
2. Follow the instructions in the Hybrid App Designer wizard:
 - **Enter or select the parent folder** – select the Hybrid App project in which to create the Hybrid App screen flow.
 - **File name** – enter a name for the Hybrid App screen flow. The extension for Hybrid App screen flows is `.xbw`.
 - **Advanced** – link the Hybrid App screen flow to an existing file in the file system.
 - **Link to file in the file system** – click **Browse** to locate the file to which to link the Hybrid App screen flow. Linked resources are files or folders that are stored in the file system outside of the project's location. If you link a resource to an editor, when you select the editor, the resource is selected in the WorkSpace Navigator. Conversely, when you select the resource in the WorkSpace Navigator, the editor is selected.

Click **Variables** to define a new path variable. Path variables specify locations on the file system.

3. In the Starting Points page, select **Credentials (authentication) may be requested dynamically from the client application**.
4. Follow the steps to create the type of Hybrid App you want. Click **Finish**.
5. In the Hybrid App Designer, open the **Flow Design** to see the Credential Request starting point and its associated Credential Request screen.

To see the two pre-defined keys, `cc_username` and `cc_password` in the Properties view, click the Credential Request starting point.
6. Double-click the **Credential Request** screen to open the Screen Design page.

The two editbox controls on the screen are bound to the pre-defined keys, `cc_username` and `cc_password`.

7. Select **Username**. In the Properties view, open the **Advanced** page.

On the Username editbox, **Credential cache username** is selected by default. Click the **Password** editbox; the associated **Credential cache password** checkbox is selected.

Note: If you create a Credential Request starting point and screen manually, you must add the editbox controls, create the keys for the username and password, and check the corresponding Credential cache username or password box.

8. (Optional) To use certificate-based authentication instead of the user name and password:
 - a) Add a **MenuItem** to the Menu box.
 - b) Select the MenuItem to see its Properties.
 - c) In the Properties view, from Type, choose **Select Certificate**.

When the user selects the menu item on the device, a dialog opens to select a certificate for credentials.

9. Select **File > Save**.

The first time the Hybrid App is started following deployment, the credential screen opens. The username and password values are cached in the credential cache.

Note: If an e-mail-triggered screen flow has dynamic cached credentials, the cached credentials are not cached between invocations of the screen flow through an e-mail trigger.

Flow Design Application Properties

The Properties view shows application properties when no objects in the Flow Design page are selected.

General

Property	Description
Module name	The name of the module.
Module version	The version of the module.
Module description	Enter a description for the module.
Module display name	Enter the name that appears on the Hybrid App when it is viewed on the client.

Property	Description
Delete message after processing	If this option is selected, server-initiated Hybrid App messages are removed from the device Inbox after they have been processed. A message is considered to have been processed if the corresponding Hybrid App 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 Hybrid App 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 Hybrid App 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 Hybrid App will have a different credentials key. If there is more than one Hybrid App with the same credentials key, then all of those Hybrid Apps can share the same credentials so that authentication for each Hybrid App does not have to be separately defined.
Requires activation key	Specify a key to use for activation. If you do not enter anything, each Hybrid App will have a different activation key, which means that each Hybrid App will have to be activated separately. If you specify an activation key, then when you activate one Hybrid App, all the other Hybrid Apps sharing the specified activation key are automatically activated.
Shared storage key	Specify a storage key to share storage between Hybrid App packages. This storage key is passed to the <code>manifest.xml</code> file used by the container.
Maximum message length	Specify in bytes. Default: 1048576.
Recoverable error retry timeout	Maximum time, in seconds, to retry sending a request after a recoverable error. A retry attempt is made after a retry time interval, which is set by default to 900 seconds. The only recoverable HTTP client error codes are 409 and 503.
Unrecoverable error retry timeout	Maximum time, in seconds, to retry sending a request after an unrecoverable error. A retry attempt is made after a retry time interval, which is set by default to 259200 seconds (three days).
Request new HTTP headers on 412 response error	Specifies that when the back end throws a 412 response error, the client send new HTTP headers.

Property	Description
Request new credentials on 401 error code	Specifies that when authentication fails, the application prompt the user to reenter credentials.

Keys

The Keys page lets you view and configure properties for keys that are associated with an application. You can sort on any property in the table by clicking its column heading.

Property	Description
Key name	The name of the associated key.
Type	The data type of the associated key, for example, string.
Data binding	The data source (if any) to which the key is bound.

The view provides buttons to perform these operations on keys:

- **Edit** – edit the properties of information a selected key.
- **Remove** – delete a selected key.
- **Usage** – view and navigate to locations in the application where a selected key is used.

Custom Icons

Device users choose your deployed Hybrid App by clicking its icon. Instead of the built-in Hybrid App icons that are provided by default, you can provide custom icons. Typically an icon includes a pair of images, representing an initial and a processed workflow status.

Custom icon images should be 32x32 pixels, and one of these types: BMP, PNG, JPG, or GIF.

Any custom icons that you include in your generated package are used instead of the built-in icons. However, if the container cannot load the icon file based on the custom icon settings (for example, if both the normal and processed icon files are not found in the package), it displays a built-in icon.

The **Add** and **Edit** buttons open the Custom Icons dialog, where you define your custom icons. Specify each custom icon using a unique combination of name, width, height, and image type.

Property	Description
Icon name	An identifier for a pair of custom icons.

Property	Description
Image files	The initial (unprocessed) and processed images. The Custom Icons dialog requires you specify an existing image file, of a supported type, from the current project.
Type	The image type, based on the extension of the specified image file name.
Height	The height of the icon image, in pixels.
Width	The width of the icon image, in pixels.

Rulers & Grid

These properties enable you to customize the default layout of the Flow Design page in the Designer.

Property	Description
Display	• Show Ruler – Displays ruler units above and to the left of the page. Default: not shown. • Show Grid – Fills the page with a grid. Default: not shown. • Grid in Front – Displays the grid in front of screen objects. Default: the grid is in front. If unselected, the grid is behind objects.
Grid line	• Color – A selected grid color, or a specified color code. Default: grey. • Style – A line style selected from the list. Default: Spaced Dot.
Measurement	• Ruler Units – Available units: Inches (default), Centimeters, Pixels. • Grid Spacing – The space between grid lines, using your selected ruler units. Default: 0.125 inches, 0.318 cm, or 12 px.
Snap	Automatically aligns objects that you relocate to specified locations on the screen. • Snap to Grid – Snaps objects to the grid lines on the canvas. Default: selected. • Snap to Shapes – Snaps objects to vertical and horizontal coordinates of the the edges of other objects.
Restore Defaults	Sets the value of all ruler and grid properties to their default values, overwriting any customizations.

Localization

Property	Description
New	Launch the New Locale wizard and create a new locale file for the Hybrid App.

Property	Description
Update	Update the currently loaded locale file.
Load	Load a locale properties file to the Hybrid App Designer.
Validate	Select a locale properties file and click to validate.
Edit	Select the locale properties file, then click to edit the locale properties file.
Remove	Select the locale properties file to remove, then click to remove.

Matching Rules

Any existing matching rules appear in the matching rules Properties view. Matching rules determine how notifications are redirected at runtime.

Property	Description
New	Create a new matching rule.
Delete	Select a matching rule and click to delete.
Edit	Select a matching rule and click to edit.

Authentication

Property	Description
Use static credentials	Select when the authentication field is the same for all users accessing the resource.
Use SAP Mobile 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 Hybrid App Designer.
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.

Property	Description
Use certificate-based credentials	Click Generate from Certificate to select a certificate file from which to generate authentication. Once you choose the certificate to use, these fields are filled in automatically. • Issuer – the issuer of the certificate • Subject – the value of the subject field in the metadata of the certificate as defined in the X.509 standard • Valid from – the date the certificate is valid from • Valid to – the date the certificate is valid to
Mobile Server domain	Optionally specify the SAP Mobile 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, <i>@domain</i> is added to the user name, thus allowing the user access to that domain. Note: When you create a new security configuration that includes the SAPSSO-TokenLoginModule, and deploy it to a new domain, if the Hybrid App uses the MBOs associated with the new security configuration, you must specify an SAP Mobile Server domain that corresponds to the domain using the security configuration. See <i>Security</i> for more information about security configurations.

File Order

This page lists the files to be generated, in the order they are loaded in the running application. File types are grouped into three tabs: jQuery Mobile, Custom, and Performance. These groups correspond to the three look and feel categories that are available for screen previews (described in *Previewing Screens*).

The control buttons enable you to add and remove files, as well as change their load order.

Note: By default, the generated package includes the `makit` folder for MAKit API support, and `datajs-version.js` OData support. If your application does not include an MChartview control or customizations that call OData library functions, you can delete these files to reduce the size of the package.

Client Variables

Client variables store data on the server that must be saved between user sessions. They are retrieved by clients using either the native API or JavaScript API (see *Manifest.xml File* in *Developer Guide: Hybrid Apps*).

The **Add**, **Delete**, and **Edit** buttons enable you to create and manage client variables. The table shows the name and value of each client variable defined for your application.

A client variable is defined as a name/value pair. Both name and string are string types, and the name is case-sensitive. However, names must be unique even when they differ by case. The

maximum name length in ANSI code (not Unicode) is 256. Due to a limitation on Windows Mobile platforms, the total length of all the client variables (keys and values) cannot exceed 2000 characters.

Flow Design Palette Options

The Flow Design Palette lists connection types, screens, and starting points that can be used in a Hybrid App.

Connections

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

Connection	Description
GoTo	Specifies which screen the user goes to upon completing an action on the current screen, such as clicking OK or Next .
Operation Success	Specifies which screen the user automatically goes to upon successful completion of an operation.
Operation Error	Specifies the screen that the user goes to automatically when an error occurs while attempting to complete an operation.
Listview Details	Specifies which screen to open when a row in a listview is clicked. Once on the details screen, you can add, update, or delete rows from the listview.
Listview Add	Specifies a screen that can be directly navigated to in order to add a new row to the listview.
Conditional Operation Success	This connection implies a code generation callout to the customConditionalNavigation function, which returns a boolean. When a user navigates through the application, each piece of custom logic is evaluated individually, in order. Upon finding the first true condition, the matching screen is shown on the user's device.
Async Request Error	Indicates that if an asynchronous operation fails, a new notification is sent to the device with the appropriate error log keys, along with user-specified e-mail field values (such as Subject and From).

Screen

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

Starting points

Starting point	Description
Activate	Specifies a screen that is displayed only the first time the user opens the application. It is not displayed subsequently. For example, you might use an Activate starting point to show instructions for using the application when the user first opens the application.
Credential Request	Provide the credentials for authentication. During execution, authentication credentials are evaluated in this order: 1. User-defined user name and password keys 2. Credential cache 3. Specified generic credential pair
Server-initiated	Defines the matching rules for the Hybrid App form, and specifies 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.

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.

Adding GoTo Connections to Screens

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

1. In the Flow Design Palette, click **GoTo**.

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

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

Connection Properties

Properties that define actions between connected objects in a Flow Design.

Properties of a connection selected on the Flow Design page:

Property	Description
Routing	New connections use the default style defined in the preferences settings. You can change the style of the connection line individually in the Properties view. • Styles: • Oblique – connection lines are not aligned with the horizontal or vertical axis, and you can create breakpoints anywhere on the connection line. If you drag the connection line, a new breakpoint is created at the drag point and two lines are created. • Rectilinear – the connection line always remains either horizontal or vertical. • Tree – arranges the connection lines on the Flow Design in a tree structure. • Avoid obstructions – connection lines go around objects. • Closest distance – connection lines pass through objects to shorten the distance of the connection line.
Lines and arrows	Line thickness and style, and arrow types.
Smoothness	• None • Less • Normal • More

Property	Description
Jump links	- Status: - None - Below - All - Above - Type: - Semicircle - Chamfered - Square - Reverse jump links

Keys

A key is a unique identifier that is bound to some data. Keys provide a flexible way for your Hybrid App to retrieve or send data.

See also

- *Hybrid App Designer* on page 1
- *Creating a New Hybrid App Screen Flow* on page 10
- *Outline View* on page 15
- *Flow Designs* on page 16
- *Screen Designs* on page 41
- *Previewing Screens* on page 44
- *Hybrid App Controls* on page 46
- *Menu Items and Custom Actions* on page 90

Defining and Using Keys

Create and reference keys to access data in your Hybrid App.

You can bind a key to various kinds of data, such as an MBO attribute, an MBO parameter, an MBO operation parameter, a subscription parameter, an object query, or an MBO relationship.

After you define a key, its data can be accessed by objects in any part of the application flow where the key is in scope. Many types of objects can use keys, including screens, controls, starting points, items in lists or menus, and parameters of an MBO operation or query.

- To define a key:

- a) In the Designer, select the object (such as a screen or control) in which you want to create the new key.
- b) In General Properties, in the Input Data Binding section, click **New key**.
- c) In the Key dialog, enter a unique name and specify its data properties to define the data that it binds.
- d) Click **OK**.
- To use a key:
 - a) In Designer, select the object in which you want to use the key.
 - b) In General Properties, in the Input Data Binding section, select an existing key.
 - c) (Optional) Enter a default value for the key.
 - d) Select **File > Save**.

Key Properties

Properties of keys vary with the objects that use them and the operations they perform.

Property	Description
Name	The name of the key.
Type	The datatype of the key, one of: • string • int • double • decimal Note: NaN (not a number) values are not supported. • bool • DateTime Note: Null values in the DateTime type are not supported for BlackBerry 5 devices. • list: can be bound to relationships or to 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.
Sent by server	Enables the key to be sent by the server as specified in the Input Data Binding properties below. By default, the key is sent by the device instead.

Property	Description
Mobile business object	The MBO associated with the key (for example, an MBO referenced in an object query in the application).
Input Data Binding	The data that is bound to the key. One of: • Mobile business object attribute – name of an attribute defined in the MBO. Optionally (if available), convert the attribute value to UTC. • Mobile business object relationship – relationship currently defined for the MBO (list type keys only). • Mobile business object query results – result of an MBO object query. • Hard-coded value – fixed value for the message sent from the server to the client. When the message is sent by the server to the client, rather than getting the value of the key from a source like an MBO attribute, it uses the fixed value you enter here. Note: You can return hard-coded key values only from server-initiated starting points, not from online requests. • User-defined – data that the user provides in the current control. • Extraction rule – name of a defined extraction rule.

Navigating Key References

Use the Key References browser to see the locations in a Hybrid App where a selected key is referenced, and optionally open the location.

The key usage browser can help you understand where data is used in an application.

- To open Key References for a key:
 - From a Properties view:
 1. Find an instance of the key in the Properties view of an object that references the key.

For example, to find a key that is used in a Menu Item, open the Menu Item's Output Keys properties page.

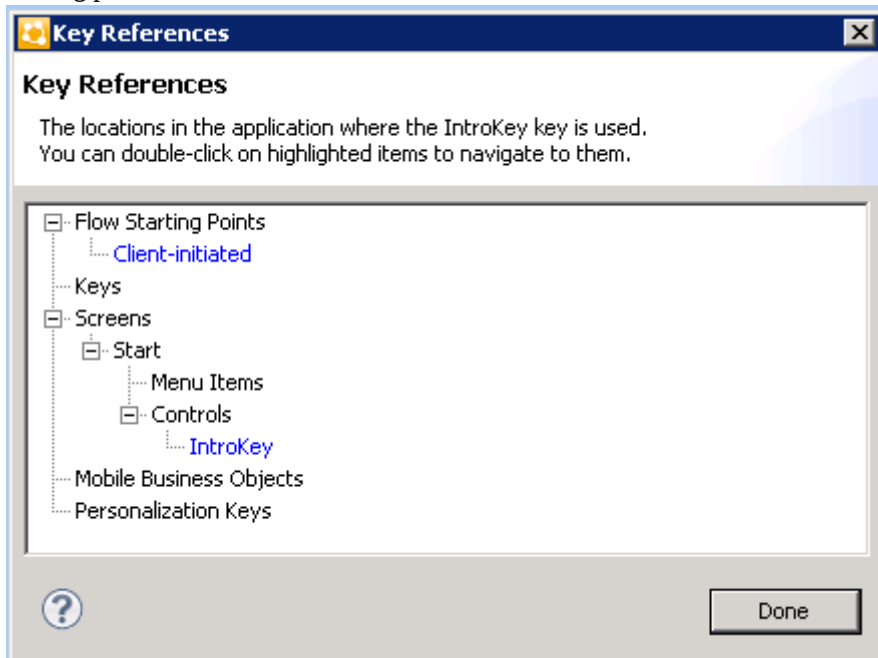
2. Select the key and click **Usage**.

- From Outline view, right-click the key and choose **Usage**..

The Key References browser lists the kinds of objects in the Hybrid App that can reference the selected key, such as starting points, screens, MBOs, and personalization keys.

- To view the key locations in the Key References browser, expand the object types to show the application objects that contain instances of the key.

In this example, the selected key (named IntroKey) is referenced in the client-initiated starting point and in the Start screen:



- To navigate to a key location, double-click a link in the list.

The selected object opens in Designer.

For example, clicking Client-initiated in the preceding example opens the client-initiated starting point, which references the IntroKey key.

Screen Designs

The screen design of your Hybrid App defines how each screen looks and behaves.

The Screen Design view enables you to quickly add menu items and controls from the Palette view, as well as your own custom actions, and to bind objects on the screen to functions and data using keys.

See also

- *Hybrid App Designer* on page 1
- *Creating a New Hybrid App Screen Flow* on page 10
- *Outline View* on page 15
- *Flow Designs* on page 16
- *Keys* on page 38
- *Previewing Screens* on page 44
- *Hybrid App Controls* on page 46
- *Menu Items and Custom Actions* on page 90

Screen Design Page Properties

The Properties view shows properties for the main Hybrid App screen.

To display Screen Design page properties, click an empty area of the screen, so that no screen objects are selected.

General

Property	Description
Name	The name of the screen.
Key	The key associated with the screen.
Background color	The color value of the control background. You can use the color picker, or manually enter the value. Valid values are color names (black, blue, brown, gray, green, orange, pink, purple, red, white, yellow) or strings in the format #<i>rrggbb</i>, where <i>rr</i>, <i>gg</i> and <i>bb</i> are the hexadecimal code values, in the range 0 to ff(255), for red, green, and blue.
Background image	An optional image, selected using the Browse button. Default: no image. Note: The image must be located in the project.
Background image repeat	How the background image repeats within the control area. One of these values: • repeat – the image repeats. • no repeat – the image does not repeat. • inherit – the background image is inherited from the parent screen. • repeat-x – the background image repeats horizontally. • repeat-y – the background image repeats vertically.
OK action	Select the action to perform when the user clicks OK on the screen.

Property	Description
Auto-arrange menus on iOS	On iOS devices, automatically sorts menu items in the order specified in the screen design. If not selected, menu items are listed in alphabetical order.

Rulers & Grid

These properties enable you to customize the default layout of a screen design page in the Designer (when no objects on the Screen Design page are selected).

Property	Description
Display	• Show Ruler – Displays ruler units above and to the left of the page. Default: not shown. • Show Grid – Fills the page with a grid. Default: not shown. • Grid in Front – Displays the grid in front of screen objects. Default: the grid is in front. If unselected, the grid is behind objects.
Grid line	• Color – A selected grid color, or a specified color code. Default: grey. • Style – A line style selected from the list. Default: Spaced Dot.
Measurement	• Ruler Units – Available units: Inches (default), Centimeters, Pixels. • Grid Spacing – The space between grid lines, using your selected ruler units. Default: 0.125 inches, 0.318 cm, or 12 px.
Snap	Automatically aligns objects that you relocate to specified locations on the screen. • Snap to Grid – Snaps objects to the grid lines on the canvas. Default: selected. • Snap to Shapes – Snaps objects to vertical and horizontal coordinates of the the edges of other objects.
Restore Defaults	Sets the value of all ruler and grid properties to their default values, overwriting any customizations.

Keys

The Keys page lets you view and configure properties for keys that are associated with a screen. You can sort on any property in the table by clicking its column heading.

Property	Description
Key name	The name of the associated key.
Type	The data type of the associated key, for example, string.
Data binding	The data source (if any) to which the key is bound.

The view provides buttons to perform these operations on keys:

- **New** – create a new key.
- **Edit** – edit the properties of information a selected key.
- **Remove** – delete a selected key.
- **Usage** – view and navigate to locations in the application where a selected key is used.

See also

- *Adding Images to the Project* on page 12

Adding and Configuring Objects from the Palette

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

1. In the Hybrid App Designer, select **Screen Design > Palette > Controls**.
2. In the Palette, select the object you want to add to the screen design.
3. Click the appropriate area in the Screen Design page to add the selected item:
 - Add a menu item to the **Menu** box.
 - Add a custom action to the **Custom Actions** box.
 - Add a control to the screen display box.
4. If the object has a name or label, you can enter the text directly in the new object. Otherwise, specify this value later, when you configure the object's properties.
5. Use the Properties page to configure the new object's properties.
If the Properties view is not open, right-click the object and select **Show Properties View**.
6. Save your changes (for example, press **Ctrl+S**).

Previewing Screens

View screen emulations for one or more devices based on your current design.

Prerequisites

Before previewing iOS, Android, and some BlackBerry devices, install Apple's Safari browser. These emulators require the WebKit rendering services distributed with Safari.

Task

The graphical views in the Flow Design and Screen Design tabs show the structure and function of screens, but not necessarily how they look when deployed on devices. For example, the Screen Design page includes invisible components, like custom actions, that can obscure the physical design. Previews can give you a better (although not exact) idea how a screen will look on a device, without having to generate, deploy, and run the application.

By default, the Hybrid App Designer displays only the Flow Design and Screen Design tabs. To see previews, enable their display in Preferences.

1. To enable previews and configure or change preview settings:
 - a) Open the Preferences dialog.
For example, click **Window > Preferences**.
 - b) In Preferences, open **SAP AG > Mobile Development > Hybrid App Designer > Screen Preview**.
 - c) Choose the preview types you want to display, and configure preview settings.
2. In the Hybrid App Designer, open the Flow Design page and select the screen to preview. Previews are not based on your screen selection in the Screen Design page.
3. Open any of the preview tabs that you configured in Preferences — one of these types:
 - **jQuery** – emulates the look of WebKit-based browser engines. Available emulators include iOS, Android devices, and BlackBerry 6 and 7.
 - **Non-jQuery** – emulates devices that are less sophisticated than Webkit, including the BlackBerry 5 devices.
 - **Performance** – emulates simple devices like Windows Mobile, or the “performance” look of other devices.

The preview page displays the name of the selected screen, and a list of available emulators.

4. Select one of the available emulators.

Tip: If a screen does not display, check for error messages in the Console or Problems view. For some emulators, you might need to upgrade Microsoft Internet Explorer or install Safari.

If the application is not legal (for example, it is missing keys), preview generation fails with this message in the Preview tab of the Console window:

```
An error occurred while generating preview.
```

This error condition has the same root cause as the application generation error, An error occurred while generating code.

5. After changing your Hybrid App (for example, adding a control or modifying properties), you can update your preview to see the effects:
 - a) Save your changes. Saving the `hybrid-app.xbw` file regenerates the previews.
Previews are not automatically updated when you change your Hybrid App in Designer.
 - b) Select a screen, then open a preview tab to see the updated preview.

Notes:

- Buttons in screen previews not functional, except for effects like hover text or highlighting.

- The page title is not displayed on some devices.
 - To clear and reload a preview at any time, right-click in the preview screen and choose **Force Refresh**.
 - To change the orientation of a preview, right-click in the screen and choose **Rotate to Landscape** or **Rotate to Portrait**.
-

See also

- *Hybrid App Designer* on page 1
- *Creating a New Hybrid App Screen Flow* on page 10
- *Outline View* on page 15
- *Flow Designs* on page 16
- *Keys* on page 38
- *Screen Designs* on page 41
- *Hybrid App Controls* on page 46
- *Menu Items and Custom Actions* on page 90

Hybrid App Controls

Controls provide visible objects on the main part of a screen that enable users to interact with the Hybrid App.

See also

- *Hybrid App Designer* on page 1
- *Creating a New Hybrid App Screen Flow* on page 10
- *Outline View* on page 15
- *Flow Designs* on page 16
- *Keys* on page 38
- *Screen Designs* on page 41
- *Previewing Screens* on page 44
- *Menu Items and Custom Actions* on page 90

Checkbox Properties

Properties for a checkbox that selects or unselects an option.

A checkbox is a small rectangle that is either selected (checked) or unselected (unchecked). The Checkbox control is supported for these platforms:

- Apple iOS
- Android

- BlackBerry
- Windows Mobile Professional

General

Property	Description
Label	• Label – text identifying the control on the user interface. • Label position – position of the label in relation to the control it describes.
Input data binding	• Key – unique identifier within the current scope, used when setting or retrieving a value. Click New key to create a new key for the control. Note: If the value is not unique, the Problems view reports a missing key error. • Default value – default value for the control.
Read only	Specifies that the control is read-only. Default: unselected. Note: The read-only property is not supported on the Windows Mobile platform for the Checkbox, Slider, and Choice controls.

Advanced

Property	Description
Validation	• Required – specifies that user input is required. • Validation message – the error message displayed to the user if the input validation criteria do not match.

EditBox Properties

Properties of a region of text.

An EditBox can be used to display and edit text, numeric, or time-related types of data on the application screen. The EditBox control is supported for these platforms:

- Apple iOS
- Android
- BlackBerry
- Windows Mobile Professional

General

Property	Description
Label	• Label – text identifying the control on the user interface. • Label position – position of the label in relation to the control it describes.
Input data binding	• Key – unique identifier within the current scope, used when setting or retrieving a value. Click New key to create a new key for the control. Note: If the value is not unique, the Problems view reports a missing key error. • Default value – default value for the control.
Read only	Specifies that the control is read-only. Default: unselected. Note: The read-only property is not supported on the Windows Mobile platform for the Checkbox, Slider, and Choice controls.
Password	If selected (set to true), text entered in the EditBox is replaced with asterisks. The default is false.
Locale specific	This option is enabled only with DATE, DATETIME, and TIME logical types. If selected, the content of the cell is rendered in Locale Display format.
Allows null	Determines whether the generated HTML definition of the control identifies the field as supporting NULL: • If selected, enables application users to change the field value between NULL and non-NULL and save it. • If not selected, prevents users from modifying the value datatype. Note: The <code>Allows null</code> property is supported only in the jQueryMobile look and feel. It is not supported for Windows Mobile devices.

Property	Description
Logical type	Select the logical type for the EditText: TEXT – string data NUMERIC – numeric data DATE – 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. DATE TIME – 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 +<i>nnn</i>," the <i>nnn</i> means seconds and "today" is interpreted as the current date time, 12:00PM. Note: Null values in the DateTime type are not supported for BlackBerry 5 devices. TIME – only absolute values, for example, "14:30" or "14:30:15." Valid values are 00:00:00 to 23:59:59. Note: Available properties vary for each logical type. For example, some properties might apply only to the numeric type, and some apply only to the text type.
Input type	Specifies the type attribute of an HTML5 input element to add one of these labeled input boxes to a jQuery Mobile-style form: - OTHER - EMAIL - NUMBER - SEARCH Note: The search box is not supported for BlackBerry 5 devices. - TEL (Telephone) - URL

Advanced

Property	Description
Content size	Maximum length – maximum number of characters allowed for the control. Number of lines – number of lines for the control. Number of decimals – number of digits after the decimal point—valid range is 0 – 28. Valid only for numeric attributes. The default is zero.

Property	Description
Validation	• Minimum value – smallest allowed value for numeric input. Valid only in with numeric attributes. • Maximum value – largest allowed value for numeric input. Valid only with numeric attributes. • Required – specifies that user input is required. • Validation regular expression – if the user enters text in the control, the input has to comply with the provided regular expression. • Validation message – the error message displayed to the user if the input validation criteria do not match.
Time precision	Specifies the precision of the time portion displayed to the user and calculations. Note: This property is used only if the EditText logical type is datetime or time.
Credential cache username	If the widget supports cached credentials, select to specify that the EditText be treated as a user name.
Credential cache password	If the widget supports cached credentials, select to specify that the EditText be treated as a password.

Choice Control

A drop-down list from which the user can choose an item.

The Choice control is supported for these platforms:

- Apple iOS
- Android
- BlackBerry
- Windows Mobile Professional

Adding, Editing, and Deleting Choice Items

Add and manage items in a Choice control.

In the General page of the Choice control Properties view:

- To add a Choice item:
 - a) Click **Add**.
 - b) Enter a unique **Display Name** and **Value** in the Choice Option Item dialog, and click **OK**.
- To edit an existing Choice item:
 - a) Select the item and click **Edit**.
 - b) Make the changes in the Choice Option Item dialog and click **OK**.

- To delete a Choice item, select the item and click **Delete**.

Creating Dynamic Contents for the Choice Control

Specify contents of the Choice control dynamically.

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

Notes:

- Each key must be unique; otherwise the names or value returned in the form are unpredictable. Because dynamic content is not resolved until runtime, the Designer cannot prevent you from entering duplicate value keys and display name keys.
 - The display name key and its associated value name key can be the same or different, but both must be children of a key that is a list type.
-

Configuring a Choice Control to Use Data from an MBO

Configuring a Choice control whose items come from a data source

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

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

Name: findByName

Return type: Multiple objects

Create an index: selected

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

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

3. Deploy the project.

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

Example: the sample application uses the key,

PurchaseOrderApprover_findByName_resultSetKey, for the Choice control.

6. In your MBO screen design, add a menu item from the Palette to the menu box.
7. Select the new menu item and configure its properties:
 - a) In the General properties page, specify the menu item name.
 - b) Choose the operation type.

These operations can invoke an online query:

Online Request
Submit Hybrid App
Add Key Collection
Update Key Collection
Delete Key Collection

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

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

8. Add a Choice control from the Palette to the MBO screen design. Select the new control and edit its properties to define data bindings:
 - a) Specify the label name and position.
 - b) Under Input Data Binding, click the **Key** control. Then select one of the available keys or specify a new key.

Example: the sample application binds
PurchaseOrder_create_purchaseOrderApproverID_paramKey.
 - c) Enable the **Dynamic** option. Bind each field to the key that you created in the screen's flow design.

Example: the sample application uses these keys:

- Value key -
PurchaseOrderApprover_findByName_resultSetKey.PurchaseOrderApprover_ID_attrbKey

- Display name key - **PurchaseOrderApprover_findByName_resultSetKey.PurchaseOrderApprover_name_attribKey**

9. Save your form.

Choice Properties

Properties of a control that displays a list from which the user selects an item.

General

Property	Description
Label	• Label – text identifying the control on the user interface. • Label position – position of the label in relation to the control it describes.
Input data binding	• Key – unique identifier within the current scope, used when setting or retrieving a value. Click New key to create a new key for the control. Note: If the value is not unique, the Problems view reports a missing key error. • Default value – default value for the control.
Read only	Specifies that the control is read-only. Default: unselected. Note: The read-only property is not supported on the Windows Mobile platform for the Checkbox, Slider, and Choice controls.
Logical type	Logical types include: • TEXT • NUMERIC Note: The availability of some properties depends on the logical type: some properties apply only to the numeric type and some apply only to the text type.
Options	• Static – the Choice contents are static. Click Add to assign a display name and value pair to the Choice item. • Dynamic – the Choice contents are dynamic.

Advanced

Property	Description
Content size	The number of digits after the decimal point. Valid range is 0 to 28. Valid only in conjunction with numeric attributes. The default is zero. Note: This property is used only if the logical type for the Choice control is set to numeric.
Validation	• Required – specifies that user input is required. • Validation message – the error message displayed to the user if the input validation criteria do not match.

Slider Properties

Properties of a control that lets a user set a numeric value by moving a slider across a range of values.

The Slider lets the user set a numeric value by moving a slider across a range of values. The Slider control is supported for these platforms:

- Apple iOS
- Android
- BlackBerry

General

Property	Description
Label	• Label – text identifying the control on the user interface. • Label position – position of the label in relation to the control it describes.
Input data binding	• Key – unique identifier within the current scope, used when setting or retrieving a value. Click New key to create a new key for the control. Note: If the value is not unique, the Problems view reports a missing key error. • Default value – default value for the control.
Read only	Specifies that the control is read-only. Default: unselected. Note: The read-only property is not supported on the Windows Mobile platform for the Checkbox, Slider, and Choice controls.

Property	Description
Value Constraints	• Minimum value • Maximum value

Advanced

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

Signature Properties

Properties of a control that displays a signature image.

The Signature control is supported for these platforms:

- Apple iOS
- Android
- BlackBerry

Note: The Signature control is not supported when the **Optimize for Performance** option is set in Preferences.

General

Property	Description
Label	• Label – text identifying the control on the user interface. • Label position – position of the label in relation to the control it describes.
Input data binding	• Key – unique identifier within the current scope, used when setting or retrieving a value. Click New key to create a new key for the control. Note: If the value is not unique, the Problems view reports a missing key error. • Default value – default value for the control. The signature value is a list of coordinates that represent the image. For example, $-1 \times -1, 5 \times 7, 5 \times 8, -1 \times -1$, where -1×-1 is interpreted as an interruption of a line drawn to the screen.

Property	Description
Read only	Specifies that the control is read-only. Default: unselected. Note: The read-only property is not supported on the Windows Mobile platform for the Checkbox, Slider, and Choice controls.

Advanced

Property	Description
Validation	• Required – specifies that user input is required. • Validation message – the error message displayed to the user if the input validation criteria do not match.

HtmlView Properties

Properties of a control that defines a block of HTML markup that you want to add to the application screen.

The HtmlView control is supported for these platforms:

- Apple iOS
- Android
- BlackBerry
- Windows Mobile Professional

Property	Description
Input data binding	• Key – unique identifier within the current scope, used when setting or retrieving a value. Click New key to create a new key for the control. Note: If the value is not unique, the Problems view reports a missing key error. • Default value – default value for the control.
Embedded HTML	The path of an HTML file that you want to include in the HtmlView control. Specify the path and configure the embedded file by clicking Browse and completing the Embedded Html Wizard. The embedded HTML overrides any value in the <code>Default value</code> property. If the HTML file being imported contains references to JS and CSS files using ". . \" directory navigation, that file reference is ignored. This is to prevent including files outside the expected HTML hierarchy.

To bind an HTMLView control to a chart that you define using the MAKit HTML5 Library in SAP Mobile WorkSpace, see *Mobile Analytics Kit Developer Guide for HTML5*. Alternatively, add an MAChartView control and define a chart directly in Designer.

Embedding an HTML File in an HtmlView Control

Include an external HTML file in an HtmlView control.

When the Hybrid App is generated, the content of the linked HTML file is inserted into the HTMLView control. This feature can be useful for incorporating screen designs that you develop using third-party tools.

1. In the Properties view of the HtmlView control, click **Browse**.
2. In the Embedded HTML Wizard, click **Import** and choose the HTML file to embed.
3. In **Target Path**, specify a root path on the target system, under which the referenced JavaScript and CSS files will be deployed.

Use a relative path from `Generated Hybrid App\hybrid-app-name\html` in the target file hierarchy. This is the default target path, with no value specified.

For example, assume this reference in the embedded html:

```
<script src="js/utils.js"></script>
```

- If the target path is empty, the reference resolves to `Generated Hybrid App\hybrid-app-name\html\js\utils.js`.
 - If the target path is `myjs`, the same reference resolves to `Generated Hybrid App\hybrid-app-name\html\myjs\js\utils.js`.
4. In the Reference File Name table, select the referenced files to deploy.
By default, all the files detected in the imported HTML file are selected. Any files that you unselect are not deployed.
 5. Click **OK** to close the Embedded HTML Wizard.
The embedded HTML file path is displayed in the HtmlView Properties view.

Listview Control

A control with a list of items that application users can click to perform the indicated actions.

For example, you might create a listview containing the items of a sales order, and the user might click **Order Details** in the list to open a details screen for the sales order.

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

The Listview control is supported for these platforms:

- Apple iOS
- Android
- BlackBerry
- Windows Mobile Professional

Adding or Editing a Listview Field

In the Properties view, add or modify Listview fields.

Prerequisites

- In the Flow Design for your screen, define any keys that you want to bind to your fields. See *Screen Design Page Properties*.
- If you are adding any static images, include the images in your project. See *Adding Images to the Project*.
- In the Screen Design page, add a ListView control to your screen from the Palette.

Task

1. Select the Listview control to display its properties.
2. In the General tab, choose a key to bind to the input, and configure any other properties you require.
3. In the left pane of the Properties view, click **Cell**.
4. To add a new cell line in the Listview, click **Add**.
5. In the Cell Lines table, select a line where you want to add or edit a field.
6. Under Fields for cell line, click **Add** to add a new field to the cell, or **Edit** to modify an existing field.
7. In the Listview Field dialog, define the field properties.
8. Click **OK**.

Generated Keys

Listview controls are bound to a list key, but the Listview displays only a subset of that list key's child keys.

For example, when you have created a Listview screen by dragging and dropping a mobile business object (MBO), the list key contains both child keys that are bound to attributes and child keys that are bound to the create, update and/or delete operations' parameters. However, the Listview displays only the keys bound to attributes.

If you create a row using parameters bound to attribute keys, you get a new row with nonempty values visible in the Listview. If you create a row using parameters bound to parameter keys, you get a new row with empty values visible in the Listview, the nonempty values are in keys not displayed in the Listview. For some MBO operations it would not be advisable to bind parameters to attribute keys (for example, an MBO with first_name and last_name attributes

and a create operation that had one `full_name` parameter); but it is appropriate for many MBO operations, specifically those automatically generated for MBOs with database back ends.

Mobile Business Object Relationships

In a Hybrid App with two mobile business objects that have a one-to-many or one-to-one composite relationship, the Listview control shows the list of related objects.

When you create the relationship between the two mobile business objects, choose a Composite relationship, and then either the **One to many** or **One to one** option.

The listview control allows you to navigate these relationships. The one-to-one cascade relationships use regular controls or keys that bind to the related mobile business object's attributes.

Note: Structured mobile business objects and complex parameters that have lists as child elements are not supported in Hybrid Apps.

Listview Properties

Properties for a list of items from a data source.

General

Property	Description
Input data binding	A unique identifier within the current scope, used when setting or retrieving a value. Click New key to create a new key for the control. Note: If the value is not unique, the Problems view reports a missing key error.
Alternate row color	A color that you enter or select using the Color Picker, for the alternate rows in the list. Available values are black, blue, brown, gray, green, orange, pink, purple, red, white, yellow, and any string with the format <code>#rrggbb</code>, where <i>rr</i>, <i>gg</i> and <i>bb</i> are the hexadecimal values for the red, green and blue, in the range 0 to ff(255).
On empty list	The text that appears on the screen when the Listview is empty.
Show header	Specifies that column titles appear in the Listview control.

Property	Description
Wrap data in PRE and CODE elements	If the text to be displayed on the device contains HTML, HTML tags, or HTML or XML escape characters, or is enclosed inside “<” or “>” (less than and greater than characters), displays the data as text on the device. HTML markup and tags, and other characters, are filtered out. For example, if this is the input: Acme <a green company> Doors and you do not select this option, the output looks like: Acme Doors on the device. The text inside the <> characters is not displayed. To display the text, select this option. For example: &apos; &amp; &lt; &gt; &#955; &#X03bb; • \] • \xA9 • \251 • \u00AE • \uQA1029 • &quot;
Locale-specific display	This option is enabled only with DATE, DATETIME, and TIME logical types. If selected, the content of the cell is rendered in “Locale Display” format.
Listview Details Screen	The screen that opens when the user clicks a row.

Cell

Property	Description
Cell lines	Displays rows in the control. Associated buttons enable you to add, delete, and move lines.
Fields for cell line	Displays fields in a selected line. Associated buttons enable you to add, delete, and move fields. For field descriptions, see <i>Listview Field Properties</i> .

Advanced

Property	Description
Numbered list	A list with numbered items. Note: Numbered lists are not supported in BlackBerry 5 devices.
Inset list	Adds a margin and rounded corners to the Listview.

Property	Description
Use page theme	Specifies that the divider uses a theme compatible with the current page theme. Default: unselected, the divider uses the default Theme B.
Filter Bar Options	• Filter bar – Enables the user to limit searches based on text in the filter bar. • Default placeholder text – If Filter bar is selected, specifies text initially displayed in the filter bar. The default text is “Filter items...”.
Dividers and Counts	• Generate divider – Provides a divider between groups of items. Default: unselected. • Case-sensitive – If Generate divider is selected, considers case when breaking for divider. • Divider Position – If Generate divider is selected and the Custom option is not selected for Generate how?, specifies one of these positions in the group: • Place divider at beginning of group • Place divider at end of group (default) • Generate how? – Choose one of these options to generate dividers: • Default – divides on the first letter of the first field. • Custom – uses callout code in <code>Custom.js</code>, where you should use the <code>customShouldGenerateListViewDivider</code> method to define the display value for the divider and to specify when to divide. This option disables the Divider Position options in the Properties view. • Key field – generates the divider on the specified key field. • Generate count bubble – If Generate divider is selected, adds a count to the end of each list item, rendered as a bubble. Note: Sort your query to match the definition of the divider. For example, if you set a field divider for <code>Dept__Name</code>, then sort your query results by <code>Dept__Name</code>.

Listview Field Properties

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

Option	Description
Key	The key with which to associate the field, selected from keys defined in the screen's Flow Design properties.
Field Width	A percentage value between 1 – 100. The widths of all fields should add up to 100.

Option	Description
Font options	One of these font types: • Plain • Bold • Italic
Field header title	A text label for the cell line.
Image	Specifies that the field contains an image instead of text. Default: unselected.
Image Height	If Image is selected, specifies a dimension in pixels. Note: Image scaling does not work correctly on BlackBerry 9700 devices.
Static Image	If Image is selected, specifies whether the image is static or dynamic: • If selected, the image must be specified by the application back-end database and must be included in the application. • If unselected (default), the image is not specified in the application, but in server or browser code at runtime.

AttachmentViewer Control

A control for downloading and viewing attachments on the device.

A file viewer must be available on the device for the associated file type, such as `.pdf`, `doc.`, or `.xls`. The AttachmentViewer control is supported for these platforms:

- Apple iOS
- Android
- BlackBerry
- Windows Mobile Professional

Adding and Configuring the AttachmentViewer

The AttachmentViewer control allows you to download and view attachments.

1. In the Hybrid App Designer, select **Screen Design > Palette > Controls**.
2. Click **AttachmentViewer** and then click in the screen.
3. From the Properties view, configure the Attachment Viewer:
 - a) Select the **AttachmentViewer** control.
 - b) Click the **Properties** tab.

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

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

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

- Keys, or
- Object queries

Note: MBO attributes displayed as attachments in a Hybrid App must be binary type, no matter what the corresponding backend datatype is.

4. (Optional) Click **Parameter Mappings** to bind the parameters (if any) of an operation or object query to keys or context data, such as DeviceName, BackEndPassword, BackEndUser, and so on

For information about the parameters and how they map to MBO operations and queries, see *WorkflowClient.xml File* in *Developer Guide: Hybrid Apps*.

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

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

6. Select **File > Save**.

AttachmentViewer Properties

Properties for a control for downloading and viewing attachments to a viewer on the device.

General

Property	Description
Label	• Label – text identifying the control on the user interface. • Label position – position of the label in relation to the control it describes.
Input data binding	• Key – unique identifier within the current scope, used when setting or retrieving a value. Click New key to create a new key for the control, then save the screen. Important: Saving the screen is required because the key is not automatically updated when you enter or change its value. Note: If the value is not unique, the Problems view reports a missing key error. • Default value – default value for the control.

Property	Description
Read-only	Specifies that the control is read-only. Default: unselected. Note: The read-only property is not supported on the Windows Mobile platform for the Checkbox, Slider, and Choice controls.
Content	Enables you to bind the AttachmentViewer to data by using an object query and mapping the parameters of the object query to keys. You indicate which binary attribute of the object query corresponds to the contents of the attachment. • Use object query – Specifies that you will bind the AttachmentViewer to data by using an object query and mapping the parameters of the object query to keys. • Mobile business object – the MBO that contains the object query you want to use. • Object query – the object query to invoke. • Attribute – the binary value that corresponds to the attachment contents. Note: The MBO binary value must be big enough to support the size of the attachment when developing the MBO. To do this, you must change the MBO's binary attribute from BINARY to BINARY(%) where % is larger than the largest binary content you want to store. See <i>Datatype Support in SAP Mobile WorkSpace - Mobile Business Object Development</i> for information about setting the binary attribute size. • MIME type key – A mime type identifies an object type recognized by the attachment viewer (see http://en.wikipedia.org/wiki/Mime-type#Type_application). Specify either a mime-type or a file name, such as <code>report.pdf</code>. For example, for an attachment named <code>doc1.pdf</code>, you can enter either the file name or its mime-type, <code>application/pdf</code>. Note: The device must have a viewer installed that can view the attachment. • File name – a key that is bound to data that defines the name of the file to be viewed. • File size – a key that is bound to data that defines the size of the file to be viewed. Note: Attachments requested on demand are limited to 5MB after Base64 encoding.

Parameter Mappings

Online requests and custom actions can bind parameters in Mobile Business Object (MBO) operations and object queries to keys or context data that are used in the submitted messages. The Parameter Mappings properties are generated as metadata in `WorkflowClient.xml`, which defines all the application's action mappings.

The Parameter Mappings table shows the available parameters in MBO operations and object queries, and their current mappings. The **Edit** button opens the Parameter Mapping dialog, where you can modify the properties of a selected mapping.

For information about parameter mapping, see *WorkflowClient.xml File* in *Developer Guide: Hybrid Apps*.

Property	Generated <i>WorkflowClient.xml</i> File Metadata
Parameter name	The name of the parameter in the operation or query that contains the data.
Parameter type	The type of parameter in MBO terms.
Mapping type	The source of the data, one of these types: • Key • BackEndPassword • BackEndUser • DeviceId • DeviceName • DeviceType • UserName • MessageId • ModuleName • ModuleVersion • QueueId • ContextVariable
Key	The key whose corresponding value is used as the value for this parameter. Used only when the Mapping type is Key.
Return data to client	Includes the key data in the return message to the client. If not selected, key data from the mapping is omitted.

Personalization Key Mappings

This page lists any personalization keys. To define the value of a personalization key, select its row in the list and click **Edit**.

Personalization keys allow the mobile user to personalize specific input field values by associating a key with a value. If your mobile business object has load parameters mapped to transient personalization keys, you can specify the key values that are loaded when an operation or object query is invoked from the Hybrid App package. This can be useful, for example, in loading specific data from large data sources into a consolidated database.

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

For more information, see *Managing Personalization Keys* in *SAP Mobile WorkSpace - Mobile Business Object Development*.

Property	Description
Personalization key	The name of the personalization key.
Key	An operation argument, load argument, or synchronization parameter that provides the value of the personalization key.
Project	The project in which this personalization key is created. The personalization key appears in the <code>Personalization Key</code> folder of the selected Mobile Application project.

Supported Attachment Types

This topic shows the attachment file types supported by each device platform.

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

AttachmentViewer and Image Limitations

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

These limitations vary by platform.

Platform	Size Limit
iOS	Large attachments can produce longer processing times.
Android	Large attachments can produce longer processing times. There is a 1MB limit for attachments on Android devices.
Windows Mobile	The maximum size of a JavaScript variable for Windows Mobile is 2MB, which allows for more memory. Warning messages are shown if the script continues for a long time, which can cause the memory to run out.
BlackBerry 5.0 and BlackBerry 6.0	On BlackBerry 5.0, the maximum size of a JavaScript variable is 500KB and on BlackBerry 6.0 and later, the maximum size of a JavaScript variable is 2MB. The maximum size must be larger than the attachment and the rest of the Hybrid App message. If the attachment is Base64-encoded, also allow for an increase in the attachment size.

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

See also

- *Image Properties* on page 78

Button Control

A control, identified by text or an image, that users click to perform an action.

The button control is supported for these platforms:

- Apple iOS
- Android
- BlackBerry

- Windows Mobile Professional

Creating an Online Request Button

Create a button that performs an online request for synchronous calls to SAP Mobile Server, which means the device client waits for a response from the server.

Prerequisites

In the Screen Design Palette, click **Button**, then click in an appropriate container (for example, an HtmlView control) to create a new button.

Task

1. Select the new item to display its properties.
2. Configure the properties in the General tab. If the operation is associated with a mobile business object (MBO), specify the MBO and configure the remaining properties in the Details section.
3. In the Parameter Mappings tab, optionally click **Edit** to modify any mappings that bind the MBO operation or object query to keys.
For information about how parameters map to MBO operations and queries, see *WorkflowClient.xml File in Developer Guide: Hybrid Apps*.
4. In the Personalization Key Mappings tab, optionally map a personalization key to a key.
You can also fill the personalization key values from values extracted from the notification, depending on where you invoke the object query from.
5. In the Output Keys tab, optionally specify keys that are filled from the server and returned to the client by the selected action.
6. Click **File > Save**.

Creating a Submit Hybrid App Button

Submitting a Hybrid App enables the user to update, delete, or create data even when not connected to the network. When the user connects, data is automatically sent and received.

Prerequisites

In the Hybrid App Designer Screen Design page, add a Button control.

Task

The Submit Hybrid App button of the first screen in Hybrid App Designer must be associated with a mobile business object (MBO) operation or an invoke parent update.

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

1. Select the new item to display its properties.
2. Configure the properties in the General tab. If the operation is associated with a mobile business object (MBO), specify the MBO and configure the remaining properties in the Details section.
3. In the Parameter Mappings tab, optionally click **Edit** to modify any mappings that bind the MBO operation or object query to keys.

For information about how parameters map to MBO operations and queries, see *WorkflowClient.xml File in Developer Guide: Hybrid Apps*.

4. In the Personalization Key Mappings tab, optionally map a personalization key to a key.
You can also fill the personalization key values from values extracted from the notification, depending on where you invoke the object query from.
5. In the Output Keys tab, optionally specify keys that are filled from the server and returned to the client by the selected action.
6. Click **File > Save**.

Button Properties

Properties for a control that users click to perform an action.

General

Property	Description
Input Label Source Binding	• Key – unique identifier within the current scope, used when setting or retrieving a value. Click New key to create a new key for the control. Note: If the value is not unique, the Problems view reports a missing key error. • Default value – the text displayed in the label by default (optional). • Label position – position of the label in relation to the control it describes.
Icon Data	• Icon – a list of standard icons such as Alert, Delete, and Plus, from which you can choose one. • Icon position – Left, Right, Top, Bottom, or NoText.
Image Data	• Has Image – indicates that the button uses an image. • Key – unique identifier for the image. Click New key to create a new key for the image. • Default value – optional path to a default image. Click Browse to navigate to the location.

Property	Description
Image Dimension	The width and height, in pixels, used to display the button image.
Options	One of these: • Static image source – value for the image is the file path to the image file, which may come from the client or server side. • Dynamic image source (binary data) – value for the image is binary data. This can come from the client or server side.
Key	A key, unique within the current scope, that identifies the item. Note: If the value is not unique, the Problems view reports a missing key error.

Property	Description
Type	The type of action the item performs: • Save – closes the current open screen, validates the current input, and saves the the collection of response values to be sent to the server. • 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. • Close Hybrid App – allows the user to leave the client-side component no matter which screen the client-side component currently shows. • Cancel – allows the user to close the current open screen without performing any data validation, and returns the user to the underlying screen. • Online Request – creates a synchronous call to the server, validates the data input on all open screens, saves it, sends it to the server and waits for a response from the server before continuing. • Submit Hybrid App – creates an asynchronous call to the server, validates the data input on all open screens, closes the application, and sends the data to the server. You cannot invoke Submit Hybrid App on a nested or child, element. A Submit Hybrid App menu item allows the user to update, delete, or create data even when not connected to the network. When the user connects, data is automatically sent and received. The Submit Hybrid App menu item of the first screen in the Designer must be associated with an MBO operation or an invoke parent update. If the user has multiple screens open at the time of the submit execution, and one of the underlying screens fails validation, the application displays the configured warning message and the submit process stops. • Select Certificate – opens a dialog where the user can choose a certificate to use for the credentials. • Add/Update/Delete Key Collection – adds, updates, or deletes a key from a key collection from a list key that is bound to a Listview control. These operations do not run immediately, but during a Submit operation on the parent MBO. You can add a key collection only to screens that users access by clicking a row in a Listview. • Reset Credentials – enables the user to authenticate using different credentials. • Reset Activation – enables the user to request that an activation sequence (such as a maximum number of login attempts) be restarted. • Custom – enables you to enter custom code to define the action.

Table 1. Details

Property	Action type	Description
The properties in the Details section vary depending on the action type.		
Mobile business object	Online Request, Submit Hybrid App	The MBO with which the menu item is associated. Click Search to locate the MBO.
Invoke parent update	Online Request, Submit Hybrid App	No changes are made to the parent MBO, but the appropriate operations (create, update, delete) are performed on its descendants.
Invoke operation	Online Request, Submit Hybrid App	Runs the selected operation for a selected MBO.
Invoke object query	Online Request, Submit Hybrid App	Runs the specified object query from the item.
Generate old value keys	Online Request, Submit Hybrid App	Specifies that old value keys are generated (default). If unselected, old value keys are not generated, resulting in a much smaller message and improved performance. However, if you need the old value keys for update or parent update operations, you should leave the property selected.
Show credential screen on authentication failure	Online Request, Submit Hybrid App	The credential screen displayed when authentication fails. If unselected, authentication failures are treated the same as other types of failures.
Execute asynchronously	Online Request, Submit Hybrid App	Specifies that the call does not wait until the current operation completes, but starts immediately. By default, the call is synchronous.
Timeout	Online Request	The maximum amount of time, in seconds, to wait for the server to respond before timing out and giving an error message.

Property	Action type	Description
Store data in on-device cache	Online Request	Enables query results to be stored in the device cache as defined in the on-device cache properties below. When unselected (default), query results are stored only in server cache.
On-device Cache Properties	Online Request	If Store data in on-device cache is selected, these properties control parameters for caching query results. - Timeout: Never Expires – Specifies that query results are cached on the device indefinitely. Expiration interval – The time, in seconds, that query results are cached on the device. If the same query is made with the same parameter values within the specified timeout, the results are fetched from the device cache and no call to the server is made. - Cache Lookup Policy: Get results from on-device cache first – Sets policy to look up results on the device before the server. Get results from server first – Sets policy to look up results on the server before the device.
Screen	Open	The screen that opens when the user clicks the item, or as the result of a GoTo connection.

Parameter Mappings

Online requests and custom actions can bind parameters in Mobile Business Object (MBO) operations and object queries to keys or context data that are used in the submitted messages. The Parameter Mappings properties are generated as metadata in `WorkflowClient.xml`, which defines all the application's action mappings.

The Parameter Mappings table shows the available parameters in MBO operations and object queries, and their current mappings. The **Edit** button opens the Parameter Mapping dialog, where you can modify the properties of a selected mapping.

For information about parameter mapping, see *WorkflowClient.xml File* in *Developer Guide: Hybrid Apps*.

Property	Generated <i>WorkflowClient.xml</i> File Metadata
Parameter name	The name of the parameter in the operation or query that contains the data.
Parameter type	The type of parameter in MBO terms.
Mapping type	The source of the data, one of these types: • Key • BackEndPassword • BackEndUser • DeviceId • DeviceName • DeviceType • UserName • MessageId • ModuleName • ModuleVersion • QueueId • ContextVariable
Key	The key whose corresponding value is used as the value for this parameter. Used only when the Mapping type is Key.
Return data to client	Includes the key data in the return message to the client. If not selected, key data from the mapping is omitted.

Personalization Key Mappings

This page lists any personalization keys. To define the value of a personalization key, select its row in the list and click **Edit**.

Personalization keys allow the mobile user to personalize specific input field values by associating a key with a value. If your mobile business object has load parameters mapped to transient personalization keys, you can specify the key values that are loaded when an operation or object query is invoked from the Hybrid App package. This can be useful, for example, in loading specific data from large data sources into a consolidated database.

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

For more information, see *Managing Personalization Keys* in *SAP Mobile WorkSpace - Mobile Business Object Development*.

Property	Description
Personalization key	The name of the personalization key.
Key	An operation argument, load argument, or synchronization parameter that provides the value of the personalization key.
Project	The project in which this personalization key is created. The personalization key appears in the <code>Personalization Key</code> folder of the selected Mobile Application project.

Output Keys

Lists keys you want an Online Request menu item or custom action to load from the server and return to the client.

Property	Description
Key Name	A key whose value is resolved by the server and returned to the client.
Type	The type of the selected key (read-only).
Data Binding	The data binding, if any, associated with the key (read-only).

You can use these buttons to manage the list of output keys:

- **Add** – Opens the Key Selection dialog, where you select a key to add to the list.
- **Remove** – Deletes the selected key from the list.
- **Usage** – Opens the Key References navigator, which shows where the selected key is used in the application.

Success/Error

The Success/Error section contains the following properties when you select the Online Request type in the General section.

Property	Description
Submit error message	Text of the message the user sees if there is an error during the submit operation. When an operation is invoked, the user can receive any error messages that may occur on the client application. The error messages are a single key of the List type, with a single key of the String type as its child. Each child instance corresponds to an error message.
Error Screen	The screen that opens if an asynchronous operation invocation fails. You can select an existing screen or click Generate Error Screen to create a new one. If no error screen is specified, the client application receives the notification in <code>processDataMessage/processWorkflowMessage</code> , whether or not an error occurs on the server. If errors occur on the server, any error log keys sent with the message include the relevant error messages. The log files on the server also contain the specifics of those errors.
Default Success Screen	The screen to open upon successful communication with the server.
Condition table	Enables you to apply custom functions that to override the default navigation between screens in the running Hybrid App. During package generation, the conditional navigation screen adds a call to the customConditionalNavigation function. The function evaluates each piece of custom logic one at a time, in order, and returns a boolean value. When the first condition comes back as true, the user is navigated to the matching screen for that condition. For more information about customizing screen navigation, see <i>Implementing Conditional Navigation</i> in <i>Developer Guide: Hybrid Apps</i>

The Success/Error section contains the following properties when you select the **Submit Hybrid App** type in the General section.

Property	Description
Submit confirmation message	(Optional) Text that is displayed in a message box when the submit operation finishes successfully.
Resubmit confirmation message	Specifies that the resubmit confirmation message is displayed if the option Mark message as processed after processing is enabled in the Flow Design page properties, and if the user attempts to submit the workflow after it has already been previously submitted.

Property	Description
Error screen	The screen to which the user navigates if an asynchronous operation invocation fails. You can select an existing screen or click Generate Error Screen to create a new one. If no error screen is specified, the client application receives no notification of the failure — failures appear only in the log files.
Subject	A subject for a notification that is sent to the device, with appropriate error log keys, if an error occurs. Note: The subject can reference the screen name and action name using macros (except in server-initiated workflows). For example: "An error occurred trying to perform the <code>\${action}</code> action on the <code>\${screen}</code> screen."
From	Text for the From field of the notification that is sent to the inbox when an error occurs. Note: The subject can reference the screen name and action name using macros (except in server-initiated workflows). See the Subject example.

Label Properties

Properties of static text that can identify an object or area of the screen flow

The Label control is supported for these platforms:

- Apple iOS
- Android
- BlackBerry
- Windows Mobile Professional

Property	Description
Input data binding	• Key – unique identifier within the current scope, used when setting or retrieving a value. Click New key to create a new key for the control. Note: If the value is not unique, the Problems view reports a missing key error. • Default value – default value for the control.

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

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

- **Default value** – default value for the control.

Link Properties

Properties of a hypertext connection to a resource.

The Link control is supported for these platforms:

- Apple iOS
- Android
- BlackBerry
- Windows Mobile Professional

Limitations of the Link control include:

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

Property	Description
Label	The text that appears comes from the key or, if that is not supplied, from the default value.
Input data binding	• Key – unique identifier within the current scope, used when setting or retrieving a value. Click New key to create a new key for the control. Note: If the value is not unique, the Problems view reports a missing key error. • Default value – default value for the control.
HREF settings	• Logical type – Type of link, such as phone number or email address. Except for the Custom type, each available type provides a preset prefix and suffix text field for your anchor. For example: <pre>send mail</pre> • Prefix – For the Custom link type, enter the prefix for your link expression. • Suffix – For the Custom link type, enter the suffix for your link expression.

Image Properties

Properties of a graphic displayed on the form.

The Image control is supported for these platforms:

- Apple iOS
- Android
- BlackBerry
- Windows Mobile Professional

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

Property	Description
Alternative Text	(Optional) Appears as text on the image if the image source cannot be displayed. In the generated HTML file, the alternative text is the value of the the <code>img</code> element <code>alt</code> attribute.
Input Image Source Binding	• Key – unique identifier within the current scope, used when setting or retrieving a value. Click New key to create a new key for the control. Note: If the value is not unique, the Problems view reports a missing key error. • Default value – default value for the control.
Image Dimension	The width and height of the button image, in pixels.
Options	One of these: • Static image source – value for the image is the file path to the image file, which may come from the client or server side. • Dynamic image source (binary data) – value for the image is binary data. This can come from the client or server side.

See also

- *AttachmentViewer and Image Limitations* on page 67

Collapsible Properties

Properties of an expandable and collapsible area of the screen.

A Collapsible control creates a dynamic HTML div element in the application screen that the user can expand or collapse. When collapsed, the section is represented by a plus sign (+) icon; when expanded, the icon changes to a minus sign (-). You can add collapsibles inside an accordion control or anywhere else on the screen.

The Collapsible control is supported for these platforms:

- Apple iOS

- Android
- BlackBerry

General

Property	Description
Label	Text on the control.
Background color	The color value of the control background. You can use the color picker, or manually enter the value. Valid values are color names (black, blue, brown, gray, green, orange, pink, purple, red, white, yellow) or strings in the format <code>#rrggbb</code>, where <i>rr</i>, <i>gg</i> and <i>bb</i> are the hexadecimal code values, in the range 0 to ff(255), for red, green, and blue.
Background image	An optional image, selected using the Browse button. Default: no image. Note: The image must be located in the project.
Background image repeat	How the background image repeats within the control area. One of these values: • repeat – the image repeats. • no repeat – the image does not repeat. • inherit – the background image is inherited from the parent screen. • repeat-x – the background image repeats horizontally. • repeat-y – the background image repeats vertically.
Initially collapsed	If selected (default), the control is initially collapsed. This is usually the appropriate setting, because there are typically multiple collapsible elements in an accordion control. If unselected, the collapsible is initially expanded.
Compact	If selected, reduces the size of the control. Default: unselected.
Icon position	The location of the icon in the control: LEFT (default), RIGHT, BOTTOM, or TOP.

Accordion Properties

Properties of a container for one or more collapsible controls.

The Accordion control is supported for these platforms:

- Apple iOS
- Android
- BlackBerry

General

Property	Description
Background color	The color value of the control background. You can use the color picker, or manually enter the value. Valid values are color names (black, blue, brown, gray, green, orange, pink, purple, red, white, yellow) or strings in the format #<i>rrggbb</i>, where <i>rr</i>, <i>gg</i> and <i>bb</i> are the hexadecimal code values, in the range 0 to ff(255), for red, green, and blue.
Background image	An optional image, selected using the Browse button. Default: no image. Note: The image must be located in the project.
Background image repeat	How the background image repeats within the control area. One of these values: • repeat – the image repeats. • no repeat – the image does not repeat. • inherit – the background image is inherited from the parent screen. • repeat-x – the background image repeats horizontally. • repeat-y – the background image repeats vertically.

Switch Properties

Properties of a control that the user can drag back and forth to switch between two exclusive choices.

The Switch control is supported for these platforms:

- iOS
- Android
- Windows Mobile Professional

Property	Description
Label	• Label – text identifying the control on the user interface. • Label position – position of the label in relation to the control it describes.
Label position	Position of the label in relation to the control it describes. LEFT (default), RIGHT, TOP, or HIDDEN.

Property	Description
Input Data Binding	Key – unique identifier within the current scope, used when setting or retrieving a value. Click New key to create a new key for the control. Note: If the value is not unique, the Problems view reports a missing key error. Default value – default value for the control.
Read only	Specifies that the control is read-only. Default: unselected. Note: The read-only property is not supported on the Windows Mobile platform for the Checkbox, Slider, and Choice controls.
Toggle Options	Settings are required for all these properties: - On Value - Off Value - On Display - Off Display On Display and Off Display are the text labels for each slider position (for example, “Yes” and “No”). On Value and Off Value are the code that you associate with each position (for example, true and false).

MACHartView Control

The MACHartView control displays a MAKit chart to provide graphical analytics capabilities for a Hybrid App.

MACHartView properties define the data to be used and how to display the chart. When you generate the code for your Hybrid App, the code includes MAKit metadata and functions to call and display your chart, based on your MACHartView properties.

For full information about MAKit charts and metadata, see *Mobile Analytics Kit (MAKit) Developer Guides: HTML5*.

The MACHartView control is supported only for these platforms:

- Apple iOS
- Android

Note: The MACHartView control is only one option for including mobile analytics charts in your Hybrid App. If you define a chart outside of the Designer by using the Mobile Analytics Kit (MAKit) component of SAP Mobile SDK, you can add that chart to the HTMLView control.

General Properties

MAChartView properties page for input data binding and the type of chart view.

Property	Description
Input data binding	A key that uniquely identifies the control.
View Type	• Full view – a single or multiseries view that displays your choice of: • Simple Chart • Zooming Chart • DrillDown Chart • Split view – a grid that displays one or more simple charts. The two properties specify the number of rows followed by the number columns in the grid.

Chart Setting Properties

MAChartView properties for the chart name and title, queries, categories, series, values, and what-if columns.

Chart setting properties are divided into several sections. You can show and hide each section individually, or all together by clicking **Expand All** or **Collapse All**.

Properties vary for each chart view type (specified in the General page).

Tabbed Views

For views that display multiple charts, the Chart Setting page displays properties for each chart in a separate tab:

DrillDown Chart – for drilldown charts, only one property is initially visible in the Chart Setting page:

Property	Description
Name	A label identifying the drilldown chart.

To display and set the remaining properties, define a zooming hierarchy of charts. To add a chart and create a new tab, each chart, click **Add** and choose a type (either **Simple Chart** or **Zooming Chart**). Tabs are named in the order that you add them (Level 1, Level 2, and so on). Each tab contains properties for the chart.

Split View – at the top of the Chart Setting page is tab for each row and column that you specify for the grid. Each tab contains properties for a simple chart. For example, if you define a split view with two columns and two rows, the page has four tabs where you define the properties for each chart.

Note: To enable the Expander control in a split view chart, edit `Custom.js` as follows:

- Make sure that `hwc.customBeforeMaximizeMASplitChart` returns `true`.
- In `hwc.customAfterMaximizeMASplitChart`, define the action of the expander control when clicked. In this example, the value of `chartName` is the name of a chart defined in the General section:

```
if (chartName === 'One') {
    hwc.navigateForward('TestScreen1');
}
else if (chartName === 'Two') {
    hwc.navigateForward('TestScreen2');
}
```

General Section

Property	Description
Name	A text identifier for the chart. In the generated HTML, the id of the chart div includes the name. For example, the id of this div is based on the name <code>chart1</code>: <pre><div id="chart1Contents"> <div id="chart1" class="chart"></div> </div></pre>
Chart Type	Specifies one of the available chart types: • Bar • Line • Pie • Horizontal Bar • Bubble
Chart Title	The label that identifies the chart on the screen.

Property	Description
MAKit Query	The name of a query (for example, RevenueByYear) that is defined in Custom.js, using the MAKit Query API, to retrieve data from data source. For information about MAKit APIs, see <i>Mobile Analytics Kit (MAKit) Developer Guides: HTML5</i>. In this example, if customBeforeExecuteMAQuery returns true, then customAfterExecuteMAQuery is called with the query name you specify. That name is used to determine what data source table to load for the MAKit data: <pre> { var dataTable; if (name == "DeptSalary") dataTable = getDeptSalary(); else if (name == "SalesOrder") dataTable = getSalesOrderSummary(); chartObj.setDataTable(dataTable); } function getDeptSalary(){ var dataTable = new \$MA.FileDataTable(); dataTable.addColumn("Department", "string"); dataTable.addColumn("Sex", "string"); dataTable.addColumn("Salary", "number"); dataTable.setData("csv", "datafiles/basic/DeptSalary.csv"); return dataTable; } </pre> Note: For zooming charts, MAKit Query is specified in the Category definition dialog.

Category Section

In an XY chart, a category typically represents the X axis. The Category section is different for simple and zooming chart view types:

Simple charts – have only one category, represented in a single column in the chart, and the Category section contains its properties.

Property	Description
Column	The column in the chart that the category is mapped to.

Property	Description
Format	Specifies the data format. For example, <code>Format="WeekDay"</code> converts a date to a weekday name in the chart. Available formats are: • Month • ShortMonth • WeekDay • ShortWeekDay • ShortDate • LongDate • Currency • Percent • Rounded{0} through Rounded{0}
Display Name	The label on the column in the chart. The display name does not have to be unique.

Zooming charts – have multiple categories, each one mapped to a different column in the chart. All the categories in a chart have the same format, which you specify at the top of the view. Under the format, the Category section lists existing categories, and buttons for managing and ordering them.

To add a category, click **Add** and define its properties in the Category definition dialog.

Property	Description
Format	The data format of all the categories in the chart.
Level	A number identifying the category's location in the zoom sequence. The level is assigned when you create the category in the Category dialog.
Title	A label identifying the category in the chart.
MAKit Query	The name of a query (for example, <code>RevenueByYear</code>) that is defined in <code>Custom.js</code> , using the MAKit Query API, to retrieve data from data source.
Column	The column in the chart where the category is mapped.

Note: For bar charts and horizontal bar charts, use fewer than 1000 categories, to optimize scrolling.

For bubble charts, user fewer than 100 categories, to avoid filling the chart with bubbles.

Series Section

In addition to categories, you can use one or more series to filter a chart. For example, in a table that shows the number of employees year by year, the year can be a category, gender can be the series, and the number of employees can be the value.

A series specifies the chart's z-axis, but because charts are not three-dimensional, MAKit maps series to columns that are added after the category columns. In a single-series chart (such as a pie chart), the series is mapped to a single column; in a multiple-series chart, you must define the column mapping for each series.

Property	Description
Column	The unique name of the column to use for the series.
Format	Specifies the data format. For example, <code>Format="WeekDay"</code> converts a date to a weekday name in the chart. Available formats are: • Month • ShortMonth • WeekDay • ShortWeekDay • ShortDate • LongDate • Currency • Percent • Rounded{0} through Rounded{0}
Display Name	The label on the column in the chart. The display name does not have to be unique.

Values Section

The value defines the data that is represented in the chart. For bubble charts only, a second tab, Bubble Size, specifies the diameter of the bubble.

Property	Description
Expression	A combination of functions, variables, and column names that is evaluated and resolved in the chart. For values, you can specify an expression that lets you select the data to be used in the chart, and how it should appear. You can manually enter an expression, or click Edit and use the Expression dialog to define the expression. The expression can be as simple as a column name from the data source, or more complex, for example, combining a column and variable. Examples: <pre>Col1+Col2*Col3+2.2 (where Col1, Col2, and Col3 are column names) 'mystring' + Col1 (concatenates <i>mystring</i> to the con- tents of Col1) Sqrt(Col2) (Square root of Col2 value on each row)</pre>

Property	Description
Format	Specifies the data format. For example, <code>Format="WeekDay"</code> converts a date to a weekday name in the chart. Available formats are: • Month • ShortMonth • WeekDay • ShortWeekDay • ShortDate • LongDate • Currency • Percent • Rounded{0} through Rounded{0}
Display Name	The label on the column in the chart. The display name does not have to be unique.

Expression Dialog

The Expression dialog lets you define or modify an expression in the Values section. To open the Expression dialog, click **Edit** in the Values section of the Chart Setting page.

Property	Description
Global Variables	A list of variables that have been defined in the Variables page of the Properties view. Clicking an item in the list inserts a reference to the variable in the Expression Definition area.
Expression Definition	The expression to be included in the Value property. You can type any part of your expression, and you can click items in the Global Variables and Available Functions lists to insert them in your expression.
Available Functions	A list of available function templates. To use a function, click an item in the list to insert the template code in the Expression Definition area, and edit the code to complete the function. For example, in the <code>String(column)</code> template, substitute the actual name of a column.

What-if Columns Section

What-if analysis enables chart users to examine “what-if” scenarios by changing input data in the what-if column. The values are recomputed and the chart is refreshed based on the new value. To use a what-if column after defining it, reference it by name in a value expression.

Semantic zooming charts do not support what-if analysis.

You can add, delete, modify, and reorder what-if columns using the buttons on the page. Clicking **Add** or **Edit** opens the What-If definition dialog, where these properties are required for all variables:

Property	Description
Name	A unique name for the column. The name must be used in a value expression.
Display Name	The label on the column in the chart. The display name does not have to be unique.
Type	The datatype: either Float or Integer.
Value	A computed expression (not a column name) that defines the data to be shown in the chart.
Min, Max	The smallest and largest valid values that application users can specify. For example, 0 and 100 for an integer variable; 0.0 and 100.0 for a float. The range between Min and Max must include the specified value property.

Variables Properties

MAChartView properties page for variables used in MAChartView value expressions.

You can add, delete, and modify variables. Clicking **Add** or **Edit** opens the Variable definition dialog, where these properties are required for all variables:

Property	Description
Variable Name	A unique identifier for the variable, used in code to reference the variable.
Display Name	A label that identifies the variable in the application. The display name can be any text, and does not have to be unique.
Type	A datatype: boolean, integer, float, date, or enum.
Value	A value of the selected datatype, used by the application unless changed at run-time.

These properties in the Variable definition dialog depend on the variable's data type:

Property	Data Types	Description
Enum Options	Enum	A list of the named enumerated values associated with the Enum type. Clicking Add or Edit opens the Enum Variable dialog, where you can define these properties for a single item: • Display Name • Value
Min, Max	Integer, Float	The smallest and largest valid values that application users can specify. For example, 0 and 100 for an integer variable; 0.0 and 100.0 for a float. The range between Min and Max must include the specified value property.

Property	Data Types	Description
Step	Float	The smallest increment by which users can change the value. For example, if Step is 0.5, a user can change 50.0 to 50.5.
Decimal	Float	The number of decimal places users must enter when specifying a value. For example if Decimal is 2, 25.00 is a valid entry, but 25 is not.

Checkbox Group Properties

Properties for a group containing one or more Checkbox controls.

Users select one or more items in each group. The Checkbox Group control is supported for these platforms:

- Apple iOS
- Android
- BlackBerry
- Windows Mobile Professional

General

Property	Description
Legend	Optional text identifying the control on the user interface. Default: none.
Horizontal	If selected, checkboxes in the group are arranged horizontally. If unselected (default), they are arranged vertically.

Menu Items and Custom Actions

A drop-down list from which the user can choose an item, or a custom action you can use to associate custom code with a menu item or control.

When you generate files for a package, a method is created for each menu item or custom action defined in the Hybrid App. For an action that is associated with a menu item, a native menu is generated for the supported BlackBerry, Android, and Windows Mobile platforms. On the iOS platform, menus are provided from HTML controls that emulate the iPhone navigation bar and toolbar.

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

See also

- *Hybrid App Designer* on page 1
- *Creating a New Hybrid App Screen Flow* on page 10

- *Outline View* on page 15
- *Flow Designs* on page 16
- *Keys* on page 38
- *Screen Designs* on page 41
- *Previewing Screens* on page 44
- *Hybrid App Controls* on page 46

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

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

Note: The first screen in each Hybrid App screen flow should have an assigned **Close** or **Cancel** menu action that enables device users to exit from the Hybrid App. On a Windows Mobile device, users can exit by clicking the **OK** button, but not all devices have that function.

1. On the Screen Design page, select **Menu Item** in the Palette, then click in the Menu box to add a menu item.
2. Select the new menu item and configure its properties in the Properties view.
3. Select **File > Save**.

Creating an Online Request Menu Item or Custom Action

Create a menu item, or custom action that performs an online request for synchronous calls to SAP Mobile Server, which means the device client waits for a response from the server.

Prerequisites

In the Hybrid App Designer Screen Design page, add a **Menu Item** or **Custom Action**.

Task

1. Select the new item to display its properties.
2. Configure the properties in the General tab. If the operation is associated with a mobile business object (MBO), specify the MBO and configure the remaining properties in the Details section.
3. In the Parameter Mappings tab, optionally click **Edit** to modify any mappings that bind the MBO operation or object query to keys.

For information about how parameters map to MBO operations and queries, see *WorkflowClient.xml File* in *Developer Guide: Hybrid Apps*.

4. In the Personalization Key Mappings tab, optionally map a personalization key to a key.

You can also fill the personalization key values from values extracted from the notification, depending on where you invoke the object query from.

5. In the Output Keys tab, optionally specify keys that are filled from the server and returned to the client by the selected action.
6. Click **File > Save**.

Creating a Submit Hybrid App Menu Item or Custom Action

Submitting a Hybrid App enables the user to update, delete, or create data even when not connected to the network. When the user connects, data is automatically sent and received.

Prerequisites

In the Hybrid App Designer Screen Design page, add a **Menu Item** or **Custom Action**.

Task

The Submit Hybrid App menu item or custom action of the first screen in Hybrid App Designer must be associated with a mobile business object (MBO) operation or an invoke parent update.

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

1. Select the new item to display its properties.
2. Configure the properties in the General tab. If the operation is associated with a mobile business object (MBO), specify the MBO and configure the remaining properties in the Details section.
3. In the Parameter Mappings tab, optionally click **Edit** to modify any mappings that bind the MBO operation or object query to keys.

For information about how parameters map to MBO operations and queries, see *WorkflowClient.xml File* in *Developer Guide: Hybrid Apps*.

4. In the Personalization Key Mappings tab, optionally map a personalization key to a key.

You can also fill the personalization key values from values extracted from the notification, depending on where you invoke the object query from.

5. In the Output Keys tab, optionally specify keys that are filled from the server and returned to the client by the selected action.
6. Click **File > Save**.

Defining Code for a Custom Menu Item or Custom Action

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

Prerequisites

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

Task

1. In the Screen Design page of the Hybrid App Designer, select the menu item, control, or custom action that you want to configure.
2. In the Properties view Type field, select **Custom**.
3. Edit the `Custom.js` file, adding custom code in the `customBeforeMenuItemClick` to handle the action.

In your code, call the custom function that is autogenerated in `hybridapp.js`

4. Save the `Custom.js` file.

The next time you regenerate the Hybrid App package, your changes to this file are preserved.

Menu Item and Custom Action Properties

Properties for a menu item or custom action selected on the Screen Design page.

General

Property	Description
Name	A text label for the menu item. Note: To prevent menu item names from wrapping on Apple iOS devices, keep the name length to 10 or fewer characters.
Key	A key, unique within the current scope, that identifies the item. Note: If the value is not unique, the Problems view reports a missing key error.
Default	Specifies that the menu item is displayed prominently on the screen.
Icon	Displays an icon with a menuitem that is in the footer on iOS devices. No other devices or locations are supported. Available icons represent common functions such as navigation arrows, Delete , and Search .

Property	Description
Type	The type of action the item performs: • Save – closes the current open screen, validates the current input, and saves the the collection of response values to be sent to the server. • 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. • Close Hybrid App – allows the user to leave the client-side component no matter which screen the client-side component currently shows. • Cancel – allows the user to close the current open screen without performing any data validation, and returns the user to the underlying screen. • Online Request – creates a synchronous call to the server, validates the data input on all open screens, saves it, sends it to the server and waits for a response from the server before continuing. • Submit Hybrid App – creates an asynchronous call to the server, validates the data input on all open screens, closes the application, and sends the data to the server. You cannot invoke Submit Hybrid App on a nested or child, element. A Submit Hybrid App menu item allows the user to update, delete, or create data even when not connected to the network. When the user connects, data is automatically sent and received. The Submit Hybrid App menu item of the first screen in the Designer must be associated with an MBO operation or an invoke parent update. If the user has multiple screens open at the time of the submit execution, and one of the underlying screens fails validation, the application displays the configured warning message and the submit process stops. • Select Certificate – opens a dialog where the user can choose a certificate to use for the credentials. • Add/Update/Delete Key Collection – adds, updates, or deletes a key from a key collection from a list key that is bound to a Listview control. These operations do not run immediately, but during a Submit operation on the parent MBO. You can add a key collection only to screens that users access by clicking a row in a Listview. • Reset Credentials – enables the user to authenticate using different credentials. • Reset Activation – enables the user to request that an activation sequence (such as a maximum number of login attempts) be restarted. • Custom – enables you to enter custom code to define the action.

Table 2. Details Section

Property	Action Type	Description
The properties in the Details section vary depending on the action type.		
Mobile business object	Online Request, Submit Hybrid App	The MBO with which the menu item is associated. Click Search to locate the MBO.
Invoke parent update	Online Request, Submit Hybrid App	No changes are made to the parent MBO, but the appropriate operations (create, update, delete) are performed on its descendants.
Invoke operation	Online Request, Submit Hybrid App	Runs the selected operation for a selected MBO.
Invoke object query	Online Request, Submit Hybrid App	Runs the specified object query from the item.
Generate old value keys	Online Request, Submit Hybrid App	Specifies that old value keys are generated (default). If unselected, old value keys are not generated, resulting in a much smaller message and improved performance. However, if you need the old value keys for update or parent update operations, you should leave the property selected.
Show credential screen on authentication failure	Online Request, Submit Hybrid App	The credential screen displayed when authentication fails. If unselected, authentication failures are treated the same as other types of failures.
Execute asynchronously	Online Request, Submit Hybrid App	Specifies that the call does not wait until the current operation completes, but starts immediately. By default, the call is synchronous.
Timeout	Online Request	The maximum amount of time, in seconds, to wait for the server to respond before timing out and giving an error message.
Store data in on-device cache	Online Request	Enables query results to be stored in the device cache as defined in the on-device cache properties below. When unselected (default), query results are stored only in server cache.

Property	Action Type	Description
On-device Cache Properties	Online Request	If Store data in on-device cache is selected, these properties control parameters for caching query results. - Timeout: Never Expires – Specifies that query results are cached on the device indefinitely. Expiration interval – The time, in seconds, that query results are cached on the device. If the same query is made with the same parameter values within the specified timeout, the results are fetched from the device cache and no call to the server is made. - Cache Lookup Policy: Get results from on-device cache first – Sets policy to look up results on the device before the server. Get results from server first – Sets policy to look up results on the server before the device.
Screen	Open	The screen that opens when the user clicks the item, or as the result of a GoTo connection.

Parameter Mappings

Online requests and custom actions can bind parameters in Mobile Business Object (MBO) operations and object queries to keys or context data that are used in the submitted messages. The Parameter Mappings properties are generated as metadata in `WorkflowClient.xml`, which defines all the application's action mappings.

The Parameter Mappings table shows the available parameters in MBO operations and object queries, and their current mappings. The **Edit** button opens the Parameter Mapping dialog, where you can modify the properties of a selected mapping.

For information about parameter mapping, see *WorkflowClient.xml File* in *Developer Guide: Hybrid Apps*.

Property	Generated <i>WorkflowClient.xml</i> File Metadata
Parameter name	The name of the parameter in the operation or query that contains the data.
Parameter type	The type of parameter in MBO terms.

Property	Generated <i>WorkflowClient.xml</i> File Metadata
Mapping type	The source of the data, one of these types: • Key • BackEndPassword • BackEndUser • DeviceId • DeviceName • DeviceType • UserName • MessageId • ModuleName • ModuleVersion • QueueId • ContextVariable
Key	The key whose corresponding value is used as the value for this parameter. Used only when the Mapping type is Key.
Return data to client	Includes the key data in the return message to the client. If not selected, key data from the mapping is omitted.

Personalization Key Mappings

This page lists any personalization keys. To define the value of a personalization key, select its row in the list and click **Edit**.

Personalization keys allow the mobile user to personalize specific input field values by associating a key with a value. If your mobile business object has load parameters mapped to transient personalization keys, you can specify the key values that are loaded when an operation or object query is invoked from the Hybrid App package. This can be useful, for example, in loading specific data from large data sources into a consolidated database.

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

For more information, see *Managing Personalization Keys* in *SAP Mobile WorkSpace - Mobile Business Object Development*.

Property	Description
Personalization key	The name of the personalization key.
Key	An operation argument, load argument, or synchronization parameter that provides the value of the personalization key.

Property	Description
Project	The project in which this personalization key is created. The personalization key appears in the <code>Personalization Key</code> folder of the selected Mobile Application project.

Output Keys

Lists keys you want an Online Request menu item or custom action to load from the server and return to the client.

Property	Description
Key Name	A key whose value is resolved by the server and returned to the client.
Type	The type of the selected key (read-only).
Data Binding	The data binding, if any, associated with the key (read-only).

You can use these buttons to manage the list of output keys:

- **Add** – Opens the Key Selection dialog, where you select a key to add to the list.
- **Remove** – Deletes the selected key from the list.
- **Usage** – Opens the Key References navigator, which shows where the selected key is used in the application.

Success/Error

The Success/Error section contains the following properties when you select the Online Request type in the General section.

Property	Description
Submit error message	Text of the message the user sees if there is an error during the submit operation. When an operation is invoked, the user can receive any error messages that may occur on the client application. The error messages are a single key of the List type, with a single key of the String type as its child. Each child instance corresponds to an error message.
Error Screen	The screen that opens if an asynchronous operation invocation fails. You can select an existing screen or click Generate Error Screen to create a new one. If no error screen is specified, the client application receives the notification in <code>processDataMessage/processWorkflowMessage</code> , whether or not an error occurs on the server. If errors occur on the server, any error log keys sent with the message include the relevant error messages. The log files on the server also contain the specifics of those errors.

Property	Description
Default Success Screen	The screen to open upon successful communication with the server.
Condition table	Enables you to apply custom functions that to override the default navigation between screens in the running Hybrid App. During package generation, the conditional navigation screen adds a call to the customConditionalNavigation function. The function evaluates each piece of custom logic one at a time, in order, and returns a boolean value. When the first condition comes back as true, the user is navigated to the matching screen for that condition. For more information about customizing screen navigation, see <i>Implementing Conditional Navigation</i> in <i>Developer Guide: Hybrid Apps</i>

The Success/Error section contains the following properties when you select the **Submit Hybrid App** type in the General section.

Property	Description
Submit confirmation message	(Optional) Text that is displayed in a message box when the submit operation finishes successfully.
Resubmit confirmation message	Specifies that the resubmit confirmation message is displayed if the option Mark message as processed after processing is enabled in the Flow Design page properties, and if the user attempts to submit the workflow after it has already been previously submitted.
Error screen	The screen to which the user navigates if an asynchronous operation invocation fails. You can select an existing screen or click Generate Error Screen to create a new one. If no error screen is specified, the client application receives no notification of the failure — failures appear only in the log files.
Subject	A subject for a notification that is sent to the device, with appropriate error log keys, if an error occurs. Note: The subject can reference the screen name and action name using macros (except in server-initiated workflows). For example: <pre>"An error occurred trying to perform the \${action} action on the \${screen} screen."</pre>

Property	Description
From	Text for the From field of the notification that is sent to the inbox when an error occurs. Note: The subject can reference the screen name and action name using macros (except in server-initiated workflows). See the Subject example.

Localization and Internationalization

You can localize different objects in the Hybrid App Designer, such as the names of screen controls, screens, and mobile business objects.

You can localize the Hybrid App by creating locale properties files. You can then load, update, and generate localized Hybrid Apps.

All the localizable strings in the Hybrid App Designer XML model work as resource keys in the localization properties file. All the localization properties files are in the same directory as the Hybrid App packages (.xbw files).

Resource keys are divided into these categories, which include all the elements of the Hybrid App Designer XML model:

- Menus
- Controls
- Screens

Localization consists of two levels of localization—the Hybrid App Designer XML model localization and the Hybrid App client localization.

All locale properties files are saved in the same directory as the Hybrid App package.

To ensure that the correct locale is picked up for the Hybrid Web Container, the following mechanism is used:

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

Localization Limitations

Locale properties files have some restrictions.

These restrictions apply:

- Traditional Chinese characters are not supported on iOS.
- Hybrid Apps 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 Hybrid App you want to localize does not have a file name that begins with a number or special character.
- When you specify a country for the language, the basic language locale must also be available. For example, if you create a locale and specify English as the language and the United States as the country, then a locale for English (the basic language) must also be available.
- If you create a locale that specifies language, country, and variant, the locale for the basic language and the locale for the basic language and the country must be available. For example, if you create a locale and specify English as the language, United States as the country, and WIN as the variant, then English (United States) and English locales must also be available.
- The language code must be a 2-letter code, and the country code can be either a 2-letter or 3-letter code.

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

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

See also

- *Localizing a Hybrid App Package* on page 102
- *Hybrid App Package Internationalization* on page 111
- *Internationalization on the Device* on page 114

Localizing a Hybrid App Package

Use the Hybrid App Designer to complete these tasks to localize Hybrid App packages (.xbw files).

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

Goal: Configure regional settings for the device.

6. *Removing a Locale*

Remove locale properties files.

7. *Updating the Current Locale*

Update the currently loaded locale properties file with the resource keys from the current Hybrid App Designer.

See also

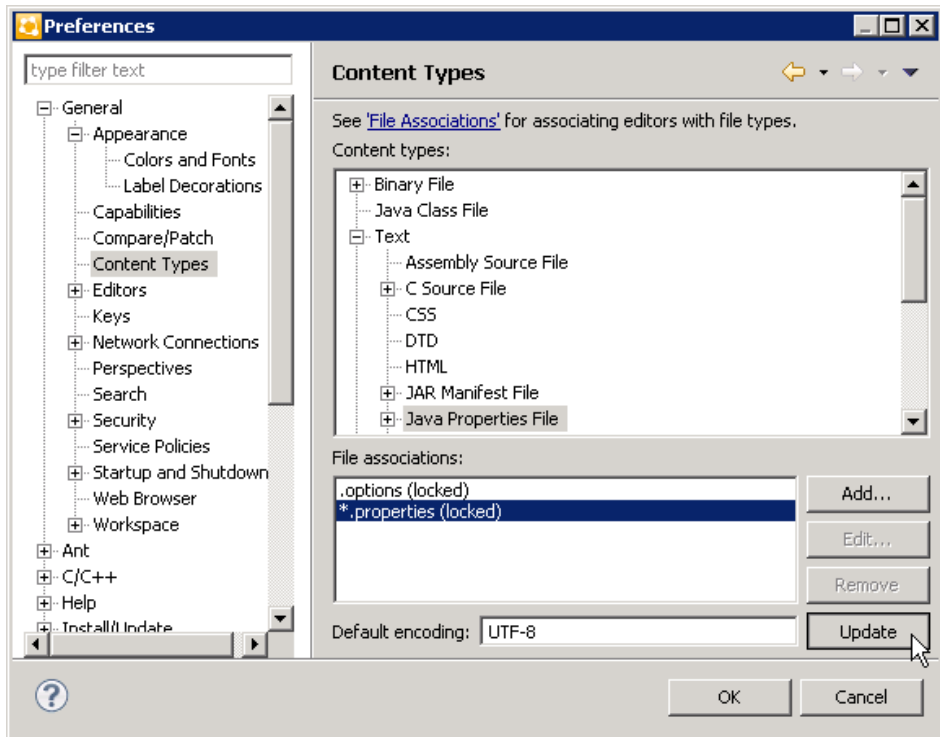
- *Localization Limitations* on page 101
- *Hybrid App Package Internationalization* on page 111
- *Internationalization on the Device* on page 114

Changing the Encoding Type

Change the encoding type in Preferences.

If you manually localize the locale properties file using an external editor, you must make sure the file is encoded in ASCII, so that the content can be correctly read and converted to Unicode. The localization file is encoded in standard ISO-8859-1. All non-ASCII character values are converted to escaped Unicode hexadecimal values before they are written to the properties files. Before translating the localization file, select the correct file encoding option, for example UTF-8.

1. In SAP Mobile 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.
5. In the Default encoding field, change ISO-8859-1 to **UTF-8**, and click **Update**.



Creating and Validating a New Locale Properties File

Create a locale properties file as the default locale.

Prerequisites

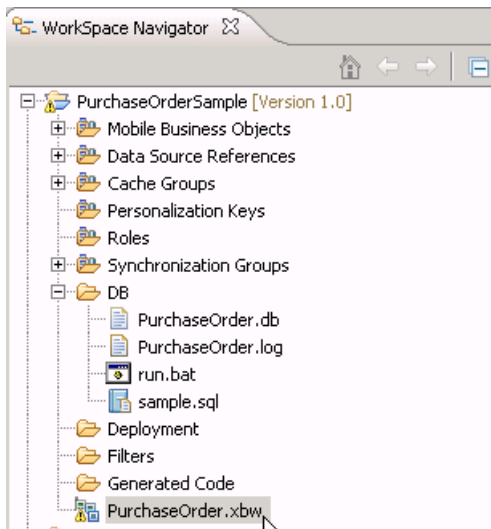
You must have an existing Hybrid App package before you create the locale properties file.

Task

When you create a new locale, keep in mind:

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

1. In WorkSpace Navigator, double-click the *Hybrid App.xbw* file to open the Hybrid App Designer.

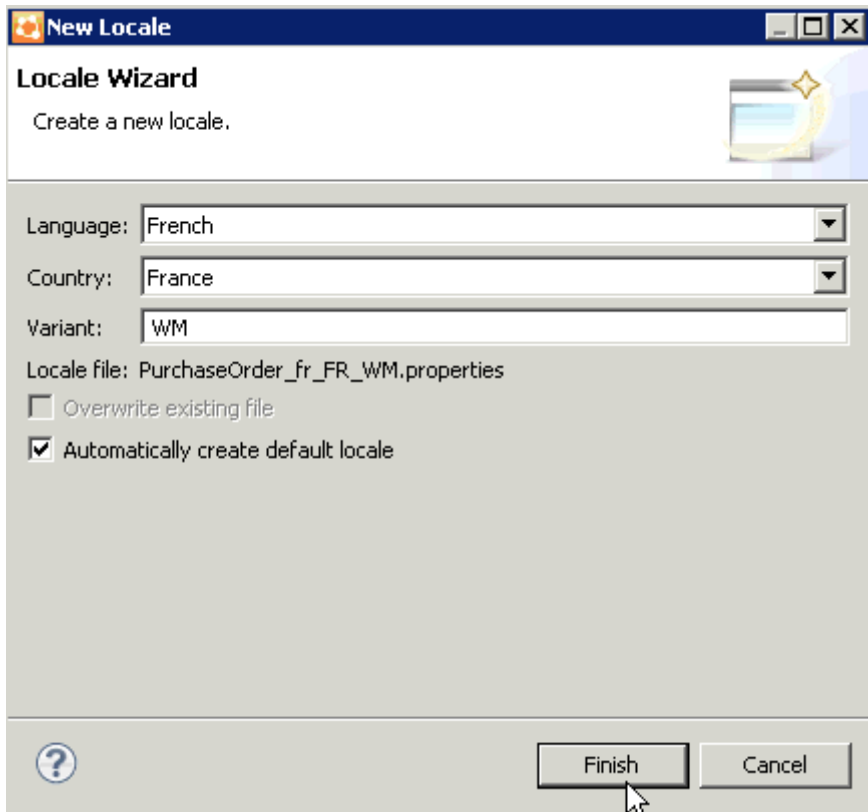


2. Click the **Flow Design** tab.
3. Right-click in a blank area on the Flow Design page, and select **Show Properties View**.
4. In the Properties view, on the left, click the **Localization** tab.
5. In the right pane, click **New**.
6. Select or enter the information for the new locale, select **Automatically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

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

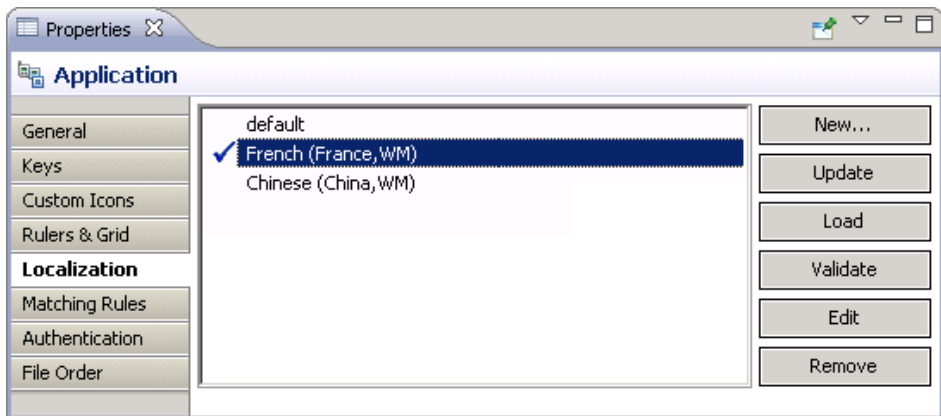
For example:

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



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

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



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

Editing the Locale Properties File

Edit the locale properties file.

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

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

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

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

Deploying the Mobile Application Project

Goal: Deploy the mobile business objects to SAP Mobile Server.

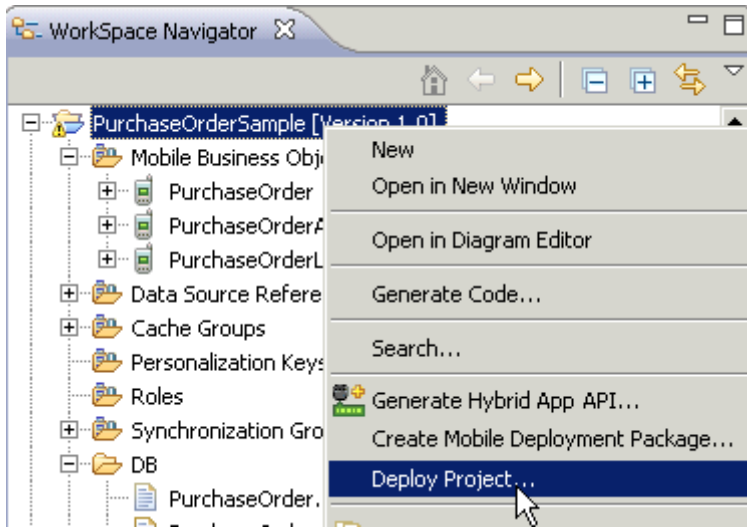
Prerequisites

SAP Mobile Server must be running.

Task

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

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



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

Configuring the Device for Localization

Goal: Configure regional settings for the device.

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

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



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

2. For iPhone devices:

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



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

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.

Updating the Current Locale

Update the currently loaded locale properties file with the resource keys from the current Hybrid App Designer.

If the locale properties file does not already exist, it is created. If the current locale is not defined in the Hybrid App file, the updated locale is used as the default, and the file name is *{device_application}.properties*. Otherwise, the locale defined in the Hybrid App file is updated.

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

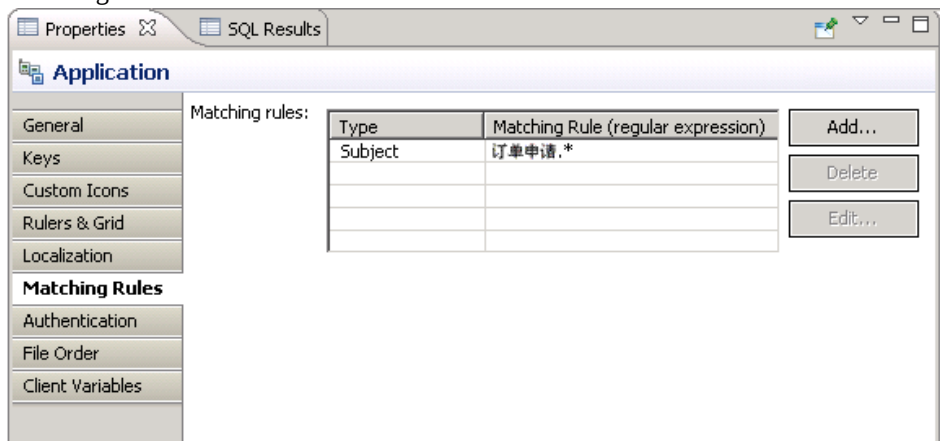
1. In the Screen Design page Properties view, click **Localization**.
2. Click **Update**.

Hybrid App Package Internationalization

The internationalization feature depends on the internationalization setting on the operating system where SAP Mobile Platform Hybrid App is running.

In the Hybrid App Designer, you can use international data in:

- Matching rules for notifications.



Send Notification To A Device User

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

Unwired Server profile: My Unwired Server

To: manager1

Cc: manager1

Bcc: manager1

From: manager1

Subject: 订单申请 (15) 待审核

Body:

Get Device Users

Send Cancel

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

Key

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

Name:

Type:

☐ Sent by server

Input Data Binding

Mobile business object:

☐ Mobile business object attribute

Name:

☒ Convert to UTC

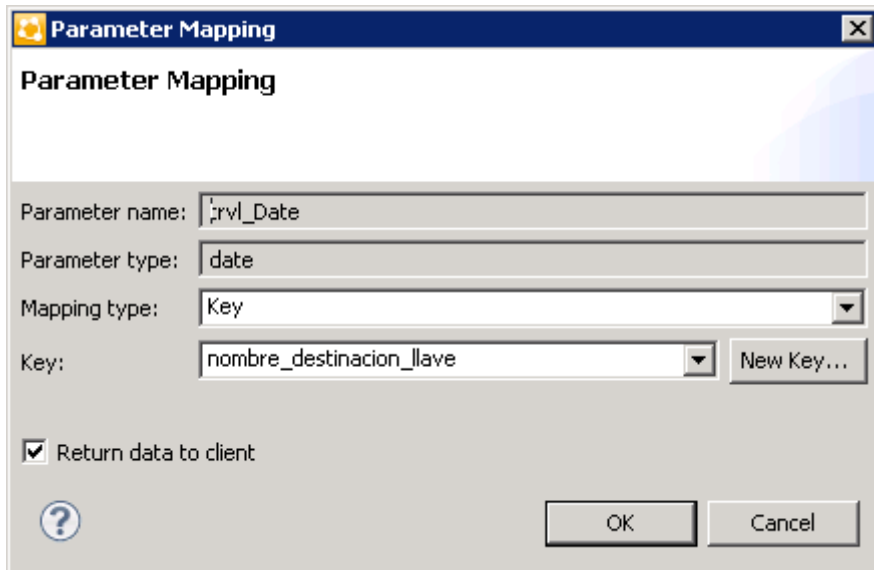
☐ Mobile business object relationship

☐ MBO object query results

☐ Hard-coded value

☒ User-defined

☐ Extraction rule



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

See also

- *Localization Limitations* on page 101
- *Localizing a Hybrid App Package* on page 102
- *Internationalization on the Device* on page 114

Internationalization on the Device

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

The internationalization feature depends on the internationalization setting on the device where the Hybrid App client running.

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



See also

- *Localization Limitations* on page 101
- *Localizing a Hybrid App Package* on page 102
- *Hybrid App Package Internationalization* on page 111

Sending a Test Notification

Send a test notification from the Hybrid App Designer to a device user to test server-initiated Hybrid App screen flows.

1. Right-click in the Flow Design or Screen Design page of the Hybrid App Designer 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.

Generating Hybrid App Files and Deploying a Package

Use the Hybrid App Package Generation wizard to generate a Hybrid App package, or to generate Hybrid App files that you can deploy to specific devices.

1. In the Hybrid App Designer, click the  code generation icon on the toolbar.
Alternatively, right-click in the Flow Design or Screen Design page and select **Generate Hybrid App Package**.
2. Specify the Mobile Server profile.
3. Choose the option to either generate a package or generate files for one or more specific platforms. Specify the required parameters, and click **Finish** to generate the files and close the wizard.

Note: The files to be generated are listed in the File Order tab of the Flow Design properties view for the application. You can optionally add or remove files or change the order in which they are loaded in the running application. See *Flow Design Application Properties* for more information.

The generated files are created in the your project, visible in Workspace Navigator under Generated Hybrid App.

4. Deploy the Hybrid App to an appropriate device or simulator.

See the *Developer Guide: Hybrid Apps* for information about how to configure devices or simulators for the Hybrid App Package.

See *SAP Control Center for SAP Mobile Platform* documentation for information about managing devices, Hybrid App assignments, and users.

Hybrid App Generation Wizard Options

Options for generating a package or platform-specific files for a Hybrid App.

Option	Description
Favorite configurations	(Optional) Select, save, or remove a configuration name. For example, after entering the values in the wizard and before clicking Finish , save the current settings to a named configuration for later reuse.
Package Generation and Deployment	

Generating Hybrid App Files and Deploying a Package

Option	Description
Mobile Server profile	Select the profile to associate with the Hybrid App and extract the user name and password credentials if you are using static authentication.
Generation options	Choose one: • Generate package Generates files for a complete, deployable package containing the Hybrid App defined in the Designer. The package includes files for all supported platforms. • Generate platform-specific-files Generates Hybrid App files for one or more specific target platforms.
Update generated code	If the Hybrid App package and its files exist, regenerate them. When you unselect this option, the Hybrid App package files are not regenerated, so that any modifications to files that are normally regenerated are not overwritten. However, any changes made in the Hybrid App Designer are not reflected in the generated files. Note: The files <code>manifest.xml</code> and <code>hybridapp-name.zip</code> are generated even if you do not select this option.
Location	Choose one: • Generate into the project – Generates the Hybrid App package and its files in the current project. • Generate to an external folder – Generates the <code>hybridapp-name.zip</code> archive into a location in your file system. Click Browse to select the location.
Assign Hybrid App to user(s)	To enable the Hybrid App to display on a device, assign it to a device user registered in SAP Control Center. Click Get Users to select a registered user. You can specify multiple users separated by commas.
Deploy to a Mobile Server as a replacement	Generates a package and deploys it to the server, replacing any previously-deployed package of the same name.
Advanced Options	
jQuery Mobile theme	Optionally, change the theme for devices that use the jQuery Mobile as the UI framework. The default is theme B. See the jQuery Mobile documentation at http://jquerymobile.com/ for information about the jQuery Mobile themes.

Option	Description
Validate controls as soon as the user tries to change focus away from them	Validation occurs as soon as the control loses focus. If validation fails, a help element appears and shows the error message. When this option is unselected, validation occurs only when the screen is saved. Note: Windows Mobile devices do not support this feature.
Optimize JavaScript in the generated Hybrid App package	The public JavaScript files (<code>API.js</code>, <code>Callbacks.js</code>, <code>Camera.js</code>, and so on) contain the client API functions that you can access for use with your Hybrid App package customization. By default, the wizard generates a single JavaScript file (such as <code>SUP0.js</code>, <code>SUP1.js</code>, or <code>SUP2.js</code>), that concatenates these files. Unselect this option to use the JavaScript files separately. Note: If you are deploying to a BlackBerry 7 or later device, selecting this option can make the Hybrid App open more quickly.
Use backwards-compatible API for generation	The generated files include support for certain deprecated API features, such as earlier naming conventions. By default, the generated code uses the current API. If selected, this option specifies that generated code uses the API for the previous MobileSDK version. This option may be useful if the current MobileSK API is incompatible with custom code in your application. Note that with the old API, your application has no access to functionality that is new in the current API.

Deployment Modes

These are the deployment modes when you generate the Hybrid App package.

New

The New deployment mode initially generates and deploys the Hybrid App package.

Replace

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

Use the Replace deployment mode for minor changes and updates to the Hybrid App, or during initial development.

Update

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

Use the Update deployment mode for major new changes to the Hybrid App.

Generated Hybrid App Files

When you use the Hybrid App Generation wizard to create a Hybrid App package, all the package files are generated the first time. Subsequent generations overwrite only a small subset of the files.

Generated package files are created in a top-level folder with the name of the Hybrid App. If you choose the option to generate into the current project, this file is visible in WorkSpace Navigator under the project `Generated Hybrid App` folder.

These files are always generated:

- *hybridapp-name.zip* – a single archive containing all of the Hybrid App files, including the Web application files, look and feel files, and JavaScript files.
- *manifest.xml* – describes the contents of the *hybridapp-name.zip* file.
- *datajs-version.js* – a JavaScript library of functions for ODATA and native device services that are not included in Hybrid Apps by default. By referencing these functions in your customization (in *Custom.js*, you can incorporate functionality from third-party JavaScript SDKs into your Hybrid Apps.

These files are regenerated only if you select the **Generate platform specific files** option in the Hybrid App Package Generation wizard:

- *hybridapp.html* – contains all the screens in the Hybrid App, each in its own div element. This is used with the **Optimize for performance** look and feel. On Windows Mobile, it is used for all looks-and-feels.
- *hybridapp_Custom.html* – contains all the screens in the Hybrid App.
- *hybridapp_jQM.html* – contains all the screens in the Hybrid App. This is used with the **Optimize for appearance** look and feel on iOS, BlackBerry, and Android.
- *WorkflowClient.xml* – contains metadata that specifies how to map the data in the Hybrid App message to and from calls to Mobile Business Object (MBO) operations and object queries.

- `hybridapp_name.xml` – look and feel file that uses the basic `hybridapp_name.html` file.
- `js` and `css` – subfolders containing the Javascript and CSS style sheet files for the application, including these files:
 - `Resources.js` – allows you to access localized string resources.
 - `HybridApp.js` – contains functions for common menu, screen, and database operations.
- PhoneGap JavaScript file. Typically named `js\platform\cordova-x.x.x.javascript`, for any Hybrid App package that is built for an Android, iOS, or BlackBerry device using the PhoneGap library. The file is copied from `<SMP_HOME>\MobileSDK<version>\HybridApp\API\Container`.

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

- `API.js` and `Utils.js` – provide Hybrid App functions used to communicate with the Hybrid Web Container.
- `Custom.js` – enables you to add JavaScript code to customize the Hybrid App. Your file is preserved each time you regenerate the package.

You can edit this file to customize your Hybrid App. It is generated the first time, but is not overwritten subsequently. In this way, your changes are preserved each time you regenerate the Hybrid App package. Examples of ways you can customize the Hybrid App include:

- Manipulating HTML elements.
- Writing code that is called before or after generated behavior is invoked for menu items.
- Implementing custom validation logic.
- `WorkflowMessage.js` – provides functions to access Hybrid App message resources.
- All `*.css` files – defines formatting rules to render the screens in HTML.

These files are overwritten when you regenerate a package:

- All the files in the top-level Generated Hybrid App\`hybridapp-name` folder, including the XML and ZIP files.
- The files in the `html` subfolder.

See *Developer Guide: Hybrid Apps* for more information about generated files and customization.

Default Look and Feel

The default look and feel is provided by the jQuery Mobile framework.

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

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

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

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

- The menu has one anchor (a) element for each menu item:

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

- In addition to a menu, each screen div has a child div element with a `data-role="content"` attribute, where the controls are hosted. The content div element has a child div with a `data-role="scroller"` attribute. This div in turn has a form with a number of div elements. The content div is where you can do customizations, such as branding.

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

The first div element is a block used to display help in a span element.

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

In the `Custom.js` file, it is recommended that you add customizations such as branding to the div element, "TopOf" ScreenKey "Form" and "bottomOf" screenKey "Form." For example:

```
/*
var screenKey = getCurrentScreen();
var form = document.forms[screenKey + "Form"];
```

```

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

```

All the other divs in the form correspond to the controls put on that screen during design time in the Hybrid App Designer. You might see, for example, a div that holds a label and a textbox (input element). When the page is opened, the controls are enhanced by jQuery Mobile to supply additional functionality for controls like buttons, sliders, text inputs, and combo boxes.

A typical Hybrid App with this look and feel, without extraneous attributes, might look like this:

```

<html>
<body onload="hwc.onHybridAppLoad();">
<div data-role="page" data-theme='a'
id="Department_createScreenDiv">
<div data-role="header" data-position="inline">
<a data-icon="arrow-l" id="Department_createScreenDivCancel"
name="Cancel" onclick="menuItemCallbackDepartment_createCancel();">
Cancel</a>
<h1>Department_create</h1>
<a id="Department_createScreenDivCreate" name="Create"
onclick="menuItemCallbackDepartment_createSubmit_Workflow();">
Create</a>
</div>
<div data-role="content" class="wrapper" >
<div data-role="scroller">
<form name="Department_createForm"
id="Department_createForm">
<div class="customTopOfFormStyle" ><span
id="Department_createForm_help" class="help"></span></div>
<div class="customTopOfFormStyle"
id="topOfDepartment_createForm"></div>
<div class="editbox">
<label class="left"
for="Department_create_dept_name_paramKey">Dept name:</label>
<input class="right" type="text"
id="Department_create_dept_name_paramKey"/><span
id="Department_create_Department_create_dept_name_paramKey_help"
class="help"></span>
</div>
<div class="customBottomOfFormStyle"
id="bottomOfDepartment_createForm"></div>
</form>
</div>
</div>
</div>
</div>

```

```
</body>  
</html>
```

Default Look and Feel CSS Files

CSS look and feel files include:

- `jquery.mobile-1.1.0.css` – located in Generated Hybrid App `\hybridapp-name\html\css\jquery` folder. By default, pages are generated using the B data theme. Modify the `ui-body-a` class selector in this file to modify the look and feel, for example, the background image or color.
- `master.css` – located in Generated Hybrid App `\hybridapp-name\html\css\bb`. Modify the `body` selector to change the look and feel, for example, the background color.
- `stylesheet.css` – located in Generated Hybrid App `\hybridapp-name\html\css`. This look and feel is simple: it uses no JavaScript code to manipulate the controls, and only a single CSS file. This style sheet is used on all platforms for which the Optimize for performance preference is selected. To modify the background color for this look and feel, modify the `body` selector.

Optimize for Performance Look and Feel

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

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

Choose the **Optimize for performance** option when you configure Hybrid App Designer preferences. For this look and feel, the layout of the HTML at a high level is:

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

result, depending on the length of the data, Listviews may not respect the field widths specified in the Hybrid App Designer with this look-and-feel.

A typical Hybrid App with this look and feel, without extraneous attributes, looks similar to this:

```
<html>
  <body onload="onHybridAppLoad();">
    <div id="Department_createScreenDiv">
      <ul id="Department_createScreenDivMenu" class="menu">
        <li><a class="nav" name="Create"
onclick="menuItemCallbackDepartment_createSubmit_Workflow();">Creat
e</a></li>
        <li><a class="nav" name="Cancel"
onclick="menuItemCallbackDepartment_createCancel();">Cancel</a></
li>
      </ul>
      <form name="Department_createForm"
id="Department_createForm">
        <table class="screen">
          <tr>
            <td colspan="2"><span id="Department_createForm_help"
class="help"></span></td>
          </tr>
          <tr>
            <td colspan="2" id="topOfDepartment_createForm"></td>
          </tr>
          <tr>
            <td class="left"><label
for="Department_create_dept_name_paramKey">Dept name:</label></td>
            <td class="right"><input class="right" type="text"
id="Department_create_dept_name_paramKey"/><span
id="Department_create_dept_name_paramKey_help"
class="help"></span></td>
          </tr>
          <tr><td colspan="2" id="bottomOfDepartment_createForm"></
td></tr></table>
        </form>
      </div>
    </body>
</html>
```

Using the BlackBerry Custom Look and Feel

You can use a custom look and feel for BlackBerry instead of the default jQuery Mobile look and feel.

If you do not want to use the jQuery Mobile-based look and feel for your Hybrid App running on BlackBerry, an alternative look and feel offers compability with the look and feel that was provided for BlackBerry 5.0 (no longer supported) in earlier releases of SAP Mobile Platform.

1. Generate your Hybrid App.

2. Open Generated Hybrid App\hybridapp-name\manifest.xml for editing.
3. Under BlackBerry/HTMLWorkflow, change the jQM look and feel XML file to the custom look and feel file.

For example, for an application named onlinequery:

- **Original** – <File>onlinequery_jQM.xml</File>
- **Custom** – <File>onlinequery_Custom.xml</File>

4. Under BlackBerry/HTMLWorkflow/HtmlFiles, replace the jQM-related files with custom files, adding any custom files of your own.

Example: this fragment shows a minimal subset of custom files:

```
<HtmlFiles>
  <HtmlFile>html/css/bb/checkbox.css</HtmlFile>
  <HtmlFile>html/css/bb/datepicker.css</HtmlFile>
  <HtmlFile>html/css/bb/editBox.css</HtmlFile>
  <HtmlFile>html/css/bb/img/btn_check_off.png</HtmlFile>
  <HtmlFile>html/css/bb/img/btn_check_on.png</HtmlFile>
  <HtmlFile>html/css/bb/img/btn_radio_off.png</HtmlFile>
  <HtmlFile>html/css/bb/img/btn_radio_on.png</HtmlFile>
  <HtmlFile>html/css/bb/img/buttonbackground.png</HtmlFile>
  <HtmlFile>html/css/bb/img/datepickerbackground.png</HtmlFile>
  <HtmlFile>html/css/bb/img/dateTextbackground.png</HtmlFile>
  <HtmlFile>html/css/bb/img/datetimepickerbackground.png</
HtmlFile>
  <HtmlFile>html/css/bb/img/downButton.png</HtmlFile>
  <HtmlFile>html/css/bb/img/ic_dialog_menu_generic.png</HtmlFile>
  <HtmlFile>html/css/bb/img/ic_menu_attachment.png</HtmlFile>
  <HtmlFile>html/css/bb/img/inputBackground.png</HtmlFile>
  <HtmlFile>html/css/bb/img/select_background.png</HtmlFile>
  <HtmlFile>html/css/bb/img/select_dropdown.png</HtmlFile>
  <HtmlFile>html/css/bb/img/select_title_bg.png</HtmlFile>
  <HtmlFile>html/css/bb/img/tinyDownArrow.png</HtmlFile>
  <HtmlFile>html/css/bb/img/tinyUpArrow.png</HtmlFile>
  <HtmlFile>html/css/bb/img/upButton.png</HtmlFile>
  <HtmlFile>html/css/bb/img/verticalBarbg.png</HtmlFile>
  <HtmlFile>html/css/bb/listview.css</HtmlFile>
  <HtmlFile>html/css/bb/master.css</HtmlFile>
  <HtmlFile>html/css/bb/radiobutton.css</HtmlFile>
  <HtmlFile>html/css/bb/Stylesheet.css</HtmlFile>
  <HtmlFile>html/css/bb/valuepicker.css</HtmlFile>
  <HtmlFile>html/css/Stylesheet.css</HtmlFile>
  <HtmlFile>html/js/API.js</HtmlFile>
  <HtmlFile>html/js/Callbacks.js</HtmlFile>
  <HtmlFile>html/js/Camera.js</HtmlFile>
  <HtmlFile>html/js/Certificate.js</HtmlFile>
  <HtmlFile>html/js/Custom.js</HtmlFile>
  <HtmlFile>html/js/ExternalResource.js</HtmlFile>
  <HtmlFile>html/js/json2.js</HtmlFile>
  <HtmlFile>html/js/Resources.js</HtmlFile>
  <HtmlFile>html/js/SUPStorage.js</HtmlFile>
  <HtmlFile>html/js/Timezone.js</HtmlFile>
  <HtmlFile>html/js/Utils.js</HtmlFile>
```

```
<HtmlFile>html/js/widgets/sy.ui.android.checkbox.js</HtmlFile>
<HtmlFile>html/js/widgets/sy.ui.android.combobox.js</HtmlFile>
<HtmlFile>html/js/widgets/sy.ui.android.listview.js</HtmlFile>
<HtmlFile>html/js/widgets/sy.ui.android.radiobutton.js</
HtmlFile>
<HtmlFile>html/js/widgets/sy.ui.bb.picker.js</HtmlFile>
<HtmlFile>html/js/widgets/sy.ui.bb.textbox.js</HtmlFile>
<HtmlFile>html/js/widgets/sy.ui.js</HtmlFile>
<HtmlFile>html/js/HybridApp.js</HtmlFile>
<HtmlFile>html/js/WorkflowMessage.js</HtmlFile>
<HtmlFile>html/hybridapp_Custom.html</HtmlFile>
</HtmlFiles>
```

5. Rebuild the package zip file to include the modified manifest.xml.
6. Deploy the package in SAP Control Center.

BlackBerry Custom Look and Feel File

hybridapp_Custom.html defines the HTML structure for the BlackBerry custom look and feel.

Each screen has a div element block with a form element, and each form has a number of div child elements. The first div in the form has a span used to display help. The next div is a built-in element that can be used to find the top of the form. The last div is another built-in element that can be used to find the bottom of the form. All the divs in the form correspond to the controls put on that screen in the Hybrid App Designer. You might get, for example, a div that holds a label and a textbox (input element).

This example shows a Hybrid App with this look and feel, without extraneous attributes:

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

Customizing Generated Code

Modify generated JavaScript code to customize the Hybrid App.

1. Use the Hybrid App Package Generation wizard to generate the Hybrid App package and its files.

When the Hybrid App package is generated, the `Custom.js` file is generated if not already present in the project. The `Custom.js` file is located in `Generated Hybrid App\<hybridapp_project_name>\html\js`.

2. Right-click the `Custom.js` file and select the editor with which to open the file.
3. Add your JavaScript code. For example, you can add code to call the APIs that are available through PhoneGap (also known as Apache Cordova).

Find the line: `hwc.customAfterHybridAppLoad = function() {`, and add:

```
document.addEventListener("deviceready", phoneGapIsReady, false);
//call the phoneGapIsReady function when PhoneGap is fully loaded.
```

Then add this function:

```
function phoneGapIsReady() {
    hwc.showAlertDialog("The connection type is " +
        navigator.network.connection.type + " and
        the platform and version is " + device.platform + " " +
        device.version);
}
```

In this example, when the Hybrid App is launched, the connection type, operating system platform and version, and device version is shown.

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

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

5. Deploy the Hybrid App package to SAP Mobile Server.

Any time you customize the code, you must redeploy the Hybrid App package to SAP Mobile Server.

You can also add your own separate JavaScript files to `Generated Hybrid Apps\<hybridapp_project_name>\html\js`, then add custom code to the `Custom.js` file that calls the functions in the JavaScript files you added. Modularizing your custom code can prevent the `Custom.js` file from becoming too long, and make it easier for multiple developers to collaborate on the same Hybrid App.

Activating the Hybrid App

Hybrid App screen menus contain two menu item types: **Submit Hybrid App** (asynchronous) and **Online Request** (synchronous).

To complete the Hybrid App activation process, the last screen in the Hybrid App must have a **Submit Hybrid App** menu item. This is necessary for the device and server-side to activate the Hybrid App for the device.

A Hybrid App is considered to have been processed or activated only if it is closed with a **Submit Hybrid App** menu item, which may or may not be tied to a mobile business object (MBO).

Generating Files for an Externally Developed Hybrid App

Use the JavaScript API Generation wizard to generate API files for a Hybrid App that uses an MBO as a data source.

Prerequisites

Whether you develop your Hybrid Apps using third-party tools or manually, you can use the JavaScript API to access the Mobile SDK container features:

- If your Hybrid App's data source is an MBO, you can use the JavaScript API Generation wizard to generate a library of functions that use the API, and that your Hybrid App references.
 - A project in your SAP Mobile WorkSpace must contain the MBOs that you want to use in your application.
 - Create any UI-related resources (for example, HTML, CSS, and JavaScript files) needed for functionality that the JavaScript API Generation wizard does not provide.
- If your Hybrid App does not use MBOs, you cannot use the JavaScript API Generation wizard, but you can still reference the JavaScript API in your application code.

To understand the overall process and how the JavaScript API generates code, read *Generating JavaScript MBO Access API* in *Developer Guide: Hybrid Apps*.

Task

To run the wizard:

1. In WorkSpace Navigator, right-click the project and choose **Generate Hybrid App API**.
2. Choose one or more MBOs to include in your application.
The Dependencies box shows relationships detected between the MBOs you select.
3. Specify the location using one of these options:
 - **Generate into the project**
The files are generated in a project subfolder, `Generated Hybrid App\APIs`.
 - **Generate to an external folder**
Specify a folder anywhere in your file system.
4. Optionally, if your Hybrid App is server initiated, click **Next** to define a notification for the object query.
 - a) In the Server-initiated Notification Configuration page, specify an object query to configure, and its associated MBO.

- b) If the query has input parameters, click **Add** to define one or more message transformation rules.

Each rule extracts a parameter value from a field in the notification. For more information, see *Notification Extraction Rules*

5. Click **Finish**.

Next

The wizard copies core API files from your installation to the generation folder you specify, and generates `WorkflowClient.xml` and `HybridApp.js`, which has the JavaScript functions for accessing the MBO operations and object queries. You can examine these files for detailed information about the API, and you can tailor them for your application.

The wizard does not deploy the application. To help you deploy it, SAP Mobile Platform includes a packaging tool that automates much of the work. See *Packaging Hybrid Apps Using the Packaging Tool* in *Developer Guide: Hybrid Apps* for information about completing and deploying your application.

Server-initiated Notification Configuration

Define a notification for a server-initiated Hybrid App.

Object Query Selection

- **Mobile Business Object** – An MBO in the current project.
- **Object Query** – A named query associated with the MBO.

Message Extraction Rules

Property	Description
Field type	The field in the notification from which the parameter value is extracted, for example, <code>Subject</code> , indicating a value in the subject line of the notification message.
Tag before parameter	Text indicating what to extract from the beginning of the field's contents, for example, <code>"Approval Request\("</code> .
Tag after parameter	Text indicating what to extract from the end of the field's contents, for example, <code>"\) was sent"</code> .
Format	Format uses the C# <code>ParseExact</code> syntax, and applies only to <code>DateTime</code> . It supports locale-specific parameter extraction. For example, you can indicate that a given date parameter is in the form <code>yyyy-MM-dd</code> in the e-mail, and another date parameter is in the form <code>yyyy-dd-MM</code> .

Property	Description
Key	The key to extract from the notification value.

Troubleshoot

Use troubleshooting tips to isolate and resolve common issues.

See *Troubleshooting SAP Mobile Platform Performance Issues* for information about troubleshooting issues with the Eclipse - based user interface or other SAP Mobile Platform components.

Index

A

- Accordion control 80
- Activate starting point 16, 35
- Add connection 35
- Android device, localization settings 108
- Async Request Error connection 35
- AttachmentViewer control 62
 - configuring 62
 - file types 66
 - image limitations 67
 - properties 63

B

- BlackBerry custom look and feel 127
- Button control 67
 - Online Request 68
 - properties 69
 - Submit Hybrid App 68

C

- Cancel menu item, creating 91
- certificate picker 27
- Checkbox control 46
- Checkbox Group control 90
- Choice control 50, 51, 82
 - adding, editing, and deleting items 50
 - properties 53
- client variables 29
- Client-initiated starting point 16, 35
- Close menu item, creating 91
- Collapsible control 79
- complex datatype 59
- conditional start 20
- connection 35, 36
 - adding GoTO 36
 - properties 37
- content type preference, changing 103
- context menu 14
- control
 - adding 44
 - configuring properties 46
- Credential Request starting point 16, 35

- credentials, static and dynamic 26
- custom action 90
 - adding 44
 - adding code 92
 - types and properties 93
- custom icons 29
- Custom.js 130

D

- data source 11
- default locale, creating 104
- deploying
 - mobile application project 107
- deployment mode 107, 121
- Dynamic authentication 28

E

- EditText control 47
- editing
 - locale properties file 107
- Embedded HTML Wizard 57
- empty screen 35
- encoding type, changing 103

F

- file association 103
- Flow Design page 16
 - preferences 3, 7
 - properties 29

G

- generated files 122
- GoTo connection 35

H

- HtmlView control 56
- Hybrid App 1
- Hybrid App Designer 1, 41
 - creating a new screen flow 10

Index

- Flow Design palette options 35
- preferences 3, 6
- setting preferences 2
- Hybrid App package
 - generated files 122
- Hybrid App Package Generation Wizard 119
- hybridapp.html 123

I

- image
 - adding to SAP Mobile Platform project 12
 - limitations in Hybrid App messages 67
- Image control 78
- Importing an HTML file into an HTMLView control 57
- internationalization
 - Hybrid App Designer 111
 - on the device 114
- iPhone device, localization settings 108
- ISO-8859-1 encoding 103

J

- JavaScript API Generation wizard 133
- jquery.mobile-1.1.0.css 123

K

- key 38
- Key References browser 40
- keys
 - defining and using 38
 - navigating 15, 40

L

- Label control 77
- Link control 78
- Listview control 57, 59
 - editing fields 58
 - field properties 61
 - generated keys 58
 - properties 59
- Listview Details connection 35
- locale
 - editing 107
 - properties file 104, 107
 - See also localization

- localization 29, 101, 108
 - creating a new locale 104
 - device settings 108
 - Hybrid App package 102
 - limitations 101
 - task flow 102
 - updating the current locale 111
- look and feel files 123

M

- manually adding a starting point 19
- master.css 123
- MBO
 - relationships 59
- menu item 90
 - adding 44
 - adding custom code 92
 - creating 91
 - Online Request 91
 - Submit Hybrid App 92
 - types and properties 93
- message extraction rules 22, 134
- message-based application 107
- mobile application package 11
- mobile application project, deploying 107
- mobile business object
 - See MBO

N

- non-ASCII encoding 103
- notification
 - extraction rule 19
 - matching rules 29
 - sending a test 117

O

- Online Request
 - button 68
 - menu item 91
- Open menu item, creating 91
- Operation Error connection 35
- Operation Success connection 35
- Optimize for performance look and feel 126
- Outline view 15

P

- package, developing for Hybrid App 1
- preferences
 - appearance 103
 - content types 103
 - general 103
- properties
 - application 29
 - controls 46
 - Server-initiated starting point 22
- PurchaseOrderSample 104

R

- regional settings, for device 108
- Replace deployment mode 121
- rulers and grid 29
- rules, notification 29

S

- Save menu item, creating 91
- screen
 - adding automatically 11
- Screen Design page
 - configuring controls 46
 - preferences 5, 8
 - properties 42
- screen design previews
 - enabling 3, 6

- screen flow
 - creating 10
- Screen Preview pages
 - preferences 6, 9
 - using 44
- Server-initiated starting point 16, 17, 35
 - configuring a notification 134
 - properties 22
- Signature control 55
- Slider control 54
- starting point 16
 - adding manually 19
 - Flow Design Palette options 35
 - Server-initiated 17
- static authentication 27
- stylesheet.css 123
- Submit Hybrid App
 - button 68
 - menu item 92
- Switch control 81

U

- Update deployment mode 121
- UTF-8 encoding 103

W

- Windows Mobile device, localization settings 108

