Mobile Application Workbench

SAP Mobile Platform 3.0

DOCUMENT ID: DC-01-0300-01
LAST REVISED: November 2013

Contents

Mobile Application Workbench

The Mobile Application Workbench (MAW) lets you customize mobile applications that you build using Mobile Application Framework (MAF). You can change SAP®-supplied configurations (also called application metadata), package the changes, and deploy them to the SAP Mobile Platform. Packages are provisioned to mobile applications that have subscribed to change notifications.

Use SAP Mobile Application Workbench to:

- View the complete application metadata
- Add customer extensions
- Provision customer extensions and customer branding without changing code

SAP Mobile Application Workbench helps you create cross-platform configurations that mobile applications can consume by using MAF. By using the editors, wizards and views you can:

- Define screen layouts
- Configure styles to control the look and feel
- Enable your application for multiple languages via localized text files, and so on.

You can use Mobile Application Workbench to generate a set of configuration files called an Application Configuration Profile (ACP), which includes layouts, styles, images, language texts, and other application-related files. ACPs are used in the mobile application runtime environment.

Setting Up the Development Environment

Set up the SAP Mobile Application Workbench development environment.

- Install SAP Mobile Platform SDK.
- Download and install Eclipse:
  1. To verify which version of Eclipse to download, see *http://service.sap.com/pam*.
  2. Navigate to: *http://www.eclipse.org/downloads/packages/*.
  3. Under Downloads Home, select the package type.
  4. Under Download Links, select the platform.

Installing Mobile Application Workbench

Install the SAP Mobile Application Workbench plug-in into the Eclipse IDE.

1. Start Eclipse and select **Help > Install New Software**.

2. Click **Add**, then in the dialog, click **Archive**.
3. Navigate to the SDK_HOME\NativeSDK\MAFWorkbench directory, and select the maw-updatesite-assembly.zip archive.
4. Click **Open**, then click **OK**.
5. Eclipse loads the repository from the specified directory. In the main Install dialog, select **SAP Mobile Application Framework Tools**, click **Next**, and follow the instructions in the installation wizard.
6. Restart Eclipse for the changes to take effect.

Upgrading Mobile Application Workbench

Upgrade the SAP Mobile Application Workbench plug-in in Eclipse.

1. Start Eclipse and select **Help > Install New Software**.
2. Click **Available Software Sites**, uncheck the Mobile Application Workbench site, and click **OK**.
3. In the Install New Software dialog, click **Add**, then click **Archive**.
4. Navigate to the SDK_HOME\NativeSDK\MAFWorkbench directory, and select the maw-updatesite-assembly.zip archive. The Eclipse IDE recognizes that this is an upgrade for the MAW plug-in.
5. Eclipse loads the repository. In the main Install dialog, select **SAP Mobile Application Framework Tools**, then click **Next**, and follow the installation wizard.
6. Restart Eclipse for the changes to take effect.

Uninstalling Mobile Application Workbench

Uninstall the SAP Mobile Application Workbench plug-in from Eclipse.

1. Start the Eclipse IDE, and select **Help** > **About Eclipse**.
2. Click **Installation Details**.
3. On the Installed Software tab, select MAW Eclipse Plug-in Feature, and click **Uninstall**.

Application Configuration Profile

An Application Configuration Profile (ACP) defines resources for Mobile Application Framework (MAF) applications. An ACP is a collection of files and folders that are packaged as a JAR file, which you can either upload to the SAP Mobile Platform Server or package with an application.

The typical ACP task flow is:

1. Create a new ACP or import one from an external JAR file.

2. Edit the configuration in the ACP project using the Mobile Application Workbench (MAW) editors and views. You can preview your changes using the viewer.
3. Add resources to the configuration, such as images, demo data, and so on.
4. Validate the project, and verify that it contains no errors.
5. Export the ACP project to a JAR file. You can also export subsets of a configuration.
6. Package the exported JAR file with your application, or upload it to the SAP Mobile Platform Server for provisioning. MAF applications can consume the configuration and modify its behavior.

Exporting an ACP project creates an ACP JAR file that contains all the ACP project files, packaged and zipped together.

The main elements of an ACP JAR file are:

- MAF Extensibility configuration (bindings, tile containers, tile layouts) – composed of the configuration XML files that define the application screens and their connections to an underlying data back end. These XML files must adhere to the conventions of the official MAF Extensibility XML Schema Definition.
- MAF Reuse Skinning configuration – found in the `styles.xml` file; it describes the appearance of the application, and governs the style of each UI element.
- Localization – composed of a set of XML files that contain translated text for various combinations of languages and countries. This text is usually externalized; the configuration and the application code refer to text items using their IDs.
- Application resources – generic files that applications use, such as images, demo data, and so on. MAF makes these files available to the application at runtime.

Application Configuration Profile Project

Application Configuration Profile projects include configuration details and resources for Mobile Application Framework (MAF)-based applications. The rules governing validation and application layouts may vary according to the version-specific characteristics of the MAF.

Manifest and Runtime Version

The core information about an Application Configuration Profile (ACP) project is described in the `META-INF\MANIFEST.MF` file. Its format follows the conventions of standard Java manifest files. In addition to standard attributes, you can define these special attributes in the file:

Attribute	Description
Customization-Resource-Bundle-Name	Name of the ACP project as a customization resource bundle, used by SAP Mobile Platform.
Customization-Resource-Bundle-Version	ACP version number, used by SAP Mobile Platform.

Attribute	Description
Runtime-Version	MAF version number for which the ACP project is created. It affects how Mobile Application Workbench (MAW) interprets the configuration within the given ACP project. If the value of this attribute changes, MAW reconfigures the project to enable or disable features, and to validate the project against the rules applicable to the newly specified version. Changing the version number does not trigger MAW to automatically migrate the ACP project from one MAF runtime version to another. MAW reports the problems that must be resolved to comply with the rules and constraints of the specified version.

Folder Structure

The folders in an ACP project are:

Folder	Platforms	Folder Contents
`ext`	All	External artifacts, custom resources, and various types of files and subfolders. MAF APIs can access the contents of this folder at runtime.
`img`	All	Images that are not dependent on platform, screen size, or density.
`img/android`	Android	DPI-independent Android-specific images.
`img/android/hdpi`	Android	Android-specific HDPI images.
`img/android/ldpi`	Android	Android-specific LDPI images.
`img/android/mdpi`	Android	Android-specific MDPI images.
`img/android/xhdpi`	Android	Android-specific XHDPI images.
`img/ios`	iOS	Device-independent iOS images.
`img/ios/ipad`	iOS	iPad-specific images.
`img/ios/iphone`	iOS	iPhone-specific images.

Folder	Plat-forms	Folder Contents
`phone`	All	Top-level folder for phone-specific configurations. It can contain only the subfolders that are listed below.
`phone/bindings`	All	Binding configurations for phones.
`phone/tileCon-tainers`	All	Tile container configurations for phones.
`phone/tiles`	All	Tile layout configurations for phones.
`string`	All	Localized text files.
`styles`	All	Style files for customizing appearances.
`tablet`	All	Top-level folder for tablet-specific configurations. Must contain only the subfolders listed below.
`tablet/bindings`	All	Binding configurations for tablets.
`tablet/tileCon-tainers`	All	Tile container configurations for tablets.
`tablet/tiles`	All	Tile layout configurations for tablets.
`demo`	All	Demo data. Must contain only the subfolders listed below.
`demo/android`	Android	Demo data for the Android platform.
`demo/ios`	iOS	Demo data for the iOS platform.

Mobile Application Workbench does not allow deviations from the folder structure described in the table; differences result in validation errors that prevent the ACP project from exporting.

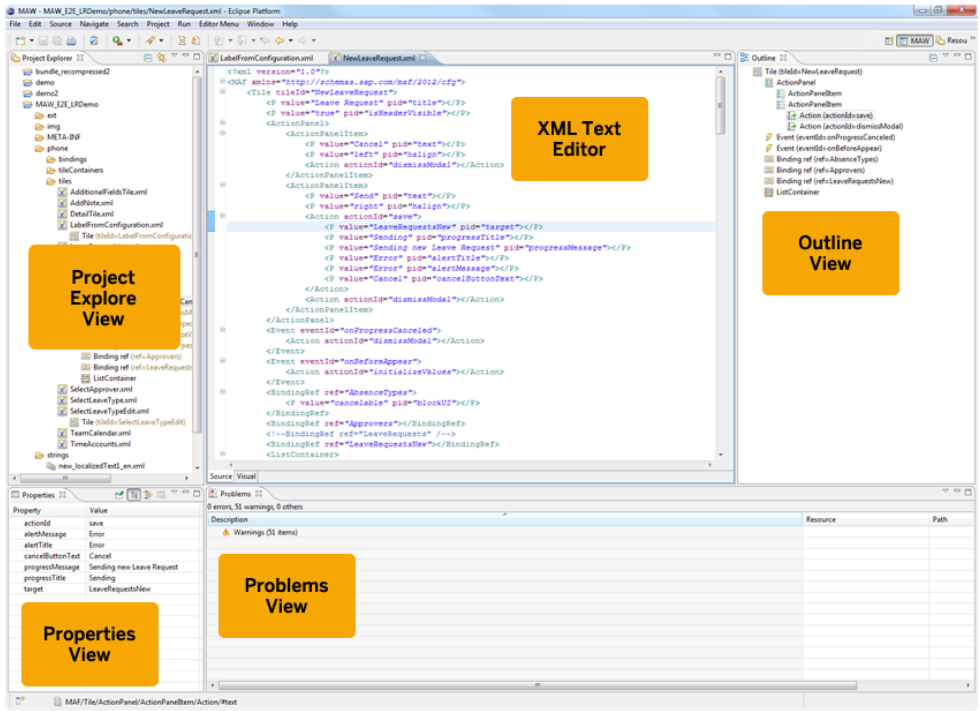
Perspectives

Mobile Application Workbench perspectives encompass and organize the view layouts.

To open a Mobile Application Workbench perspective in Eclipse, select **Window > Open Perspective > Other > MAW**. The available UI components include:

- Project Explorer view – displays existing projects, including Application Configuration Profile (ACP) projects, and their folder structures, configuration files, and configuration elements, which comprise Mobile Application Framework applications.
- Properties view – lets you edit the key-value pair of the configuration element that is selected in either the Project Explorer or the XML text editor.

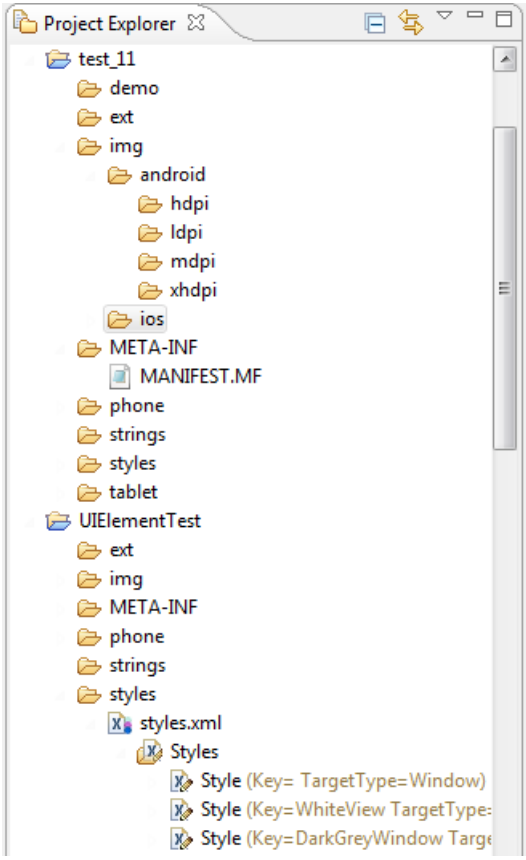
- Problems view – displays warnings and errors for all projects, including ACP projects.
- XML text editor – displays the selected XML configuration file, and lets you edit the property key-value pairs. You can open several instances of the editor to view different configuration files.
- Outline view – shows an outline of the current configuration file for easy navigation.



Project Explorer View

The Project Explorer is the standard hierarchical overview shared by many plug-ins and components within the Eclipse IDE.

The Project Explorer is similar to the Package Explorer that is used in Java Development Tools. Mobile Application Workbench defines the same hierarchy for each ACP artifact that is available in the Outline view:



Use this view to create, edit, delete, or move files, as in a file explorer. The pop-up menus for the selected items help you perform these tasks.

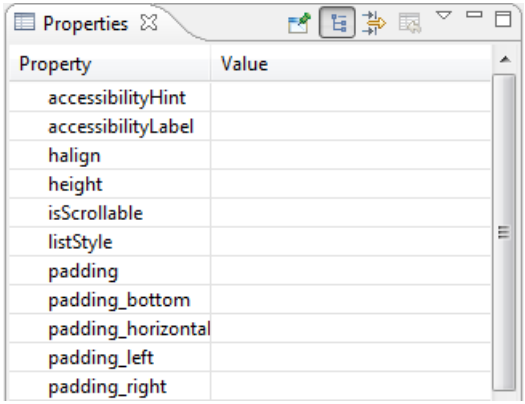
Outline View

The Outline view is the area to the right of the editor, which shows the hierarchical structure of the file being edited.

In this view, Mobile Application Workbench (MAW) displays a hierarchy of all its XML-based artifacts that can be part of an Application Configuration Profile project. If you select an element in the Outline view, MAW synchronizes it with both the active editor and the Properties view.

Properties View

Use the Properties view to edit key-value pairs, such as XML attributes, binding parameters, tile containers, tile layouts, UI elements, and so on.



Use the Properties view when there are lot of parameters to edit and you do not want to spend time editing the corresponding XML for each parameter. For elements defined by MAW, the list of configurable parameters are ready for editing; you can also define custom parameters.

Wizards

You can use the Mobile Application Workbench wizards to create and configure Application Configuration Profile projects.

Creating Application Configuration Profiles

Create an Application Configuration Profile project using the ACP wizard.

1. Select **File > New > Other > Mobile Application Framework > ACP**.
2. Enter the project name and the runtime version.
3. (Optional) Specify an alternate location in the file system.
4. (Optional) Specify the Working sets.
5. Click **Finish**.

   The project is created in the Eclipse workspace.

Creating Binding Configurations

You can configure data bindings for Application Configuration Profile projects.

1. Select **File > New > Other > Mobile Application Framework > Binding file**.

All fields are required.

2. Enter the target folder, or click **Browse** and select it.

   The only folders that appear are those that can contain the file to be generated.

3. Select the runtime version.

   This specifies the template that Mobile Application Workbench (MAW) uses to generate the Application Configuration Profile (ACP). If the runtime version of the file that is generated is not compatible with the runtime version of the enclosing ACP project, MAW displays warnings and errors, so you can fix the problem immediately.

4. Enter an ID for the binding, and click **Finish**.

Creating Localized Text Configurations

You can configure localized text for Application Configuration Profile projects.

1. Select **File > New > Other > Mobile Application Framework > Localized text file**.

   All fields are required.

2. Enter the target folder, or click **Browse** and select it.

   The only folders that appear are those that can contain the file to be generated.

3. Select the runtime version.

   This specifies the template that Mobile Application Workbench (MAW) uses to generate the Application Configuration Profile (ACP). If the runtime version of the file that is generated is not compatible with the runtime version of the enclosing ACP project, MAW displays warnings and errors, so you can fix the problem immediately.

4. Specify the ISO language and the country code.

   MAW appends these codes to the base file name; for example, if you enter `en` as the language code, and `gb` as the country code, the file name is `new_localizedText_en_gb.xml`

5. Click **Finish**.

Creating Style Configurations

You can configure styles for Application Configuration Profile projects.

1. Select **File > New > Other > Mobile Application Framework > Style file**.
2. Enter the target folder, or click **Browse** and select it.

   The only folders that appear are those that can contain the file to be generated.

3. Select the runtime version.

   This specifies the template that Mobile Application Workbench (MAW) uses to generate the Application Configuration Profile (ACP). If the runtime version of the file that is

generated is not compatible with the runtime version of the enclosing ACP project, MAW displays warnings and errors, so you can fix the problem immediately.

4. Click **Finish**.

Creating Tile Container Configurations

You can configure tile containers for Application Configuration Profile projects.

1. Select **File > New > Other > Mobile Application Framework > Tile container file**.
2. Enter the name of the target folder, or click **Browse** and select it.

   The only folders that appear are those that can contain the file to be generated.
3. Select the runtime version.

   This specifies the template that Mobile Application Workbench (MAW) uses to generate the Application Configuration Profile (ACP). If the runtime version of the file that is generated is not compatible with the runtime version of the enclosing ACP project, MAW displays warnings and errors, so you can fix the problem immediately.
4. Enter an ID for the tile container.

   MAW uses the ID as the base name of the XML file; for example, if you enter `abc`, the file name is `abc.xml`.
5. Click **Finish**.

Creating Tile Layout Configurations

You can configure tile layouts for Application Configuration Profile projects.

1. Select **File > New > Other > Mobile Application Framework > Tile file**.
2. Enter the name of the target folder, or click **Browse** and select it.

   The only folders that appear are those that can contain the file to be generated.
3. Select the runtime version.

   This specifies the template that Mobile Application Workbench (MAW) uses to generate the Application Configuration Profile (ACP). If the runtime version of the file that is generated is not compatible with the runtime version of the enclosing ACP project, MAW displays warnings and errors, so you can fix the problem immediately.
4. Enter an ID for the tile file.

   MAW uses the ID as the base name of the XML file; for example, if you enter `abc`, the file name is `abc.xml`.
5. Click **Finish**.

Exporting Application Configuration Profiles

You can export an Application Configuration Profile project to a JAR file.

1. Select **File > Export > Mobile Application Framework > Export ACP to a JAR file**.

   Before the wizard pops up, Mobile Application Workbench checks for errors in the project. You cannot export projects that contain errors. This prevents the creation of malformed Application Configuration Profile (ACP) projects that the Mobile Application Framework runtime cannot process correctly.
2. Enter the name of the ACP project you want to export, or click **Browse** and select it.
3. Specify the name and location of the destination JAR file; you can also click **Browse** to select the location.
4. Click **Finish**.

Exporting Tile Layouts

You can export Application Configuration Profile tile layouts, and add them to mobile applications.

1. Select **File > Export > Mobile Application Framework > Export Layout to a file**.
2. Enter the project name, or click **Browse** and select it.
3. Select the device type, either Phone or Tablet.
4. Specify the export location for the layout, and click **Finish**.

   During export, MAW takes configurations from the relevant directories, including bindings, tile containers, and tiles.

You can add the exported file to a mobile application, and process it using the corresponding API calls in Mobile Application Framework Extensibility.

Importing Application Configuration Profiles

You can import an Application Configuration Profile project from a JAR file.

1. Select **File > Import > Mobile Application Framework > Import ACP from a JAR file**.
2. Specify the JAR file to import, or click **Browse** and select it.

   Mobile Application Workbench (MAW) analyzes the file and recommends a name and version number for the project. The list contains all the versions with which the specified JAR file is compatible. If the **Runtime-Version** attribute is set in the JAR file's `MANIFEST.MF` file, that version number is marked with an asterisk in the list.
3. Keep the recommended project name, or enter another name.
4. Keep the recommended version number, or select another version from the list.

MAW converts the JAR file to work with the specified Mobile Application Framework variant. If there are unrecognized files and folders in the JAR file, the import wizard asks you to specify how to import each unrecognized item. Possible actions include:

- Ignore – the file/directory is not copied into the new Application Configuration Profile (ACP) project.
- Import as-is – the file/directory is copied to the same location in the ACP project as it exists in the JAR file. This typically results in validation errors in the project, because such files are known to violate the folder structure. You can use this option when you want to quickly import the JAR file, and plan to move files to their correct locations later.
- Import into 'ext' – the file/directory is copied into the `ext` directory, which can contain any file type. The directory hierarchy relative to the root of the JAR file is preserved when copying.

To set an action for all files and directories at the same time, select the action from the bottom of the list.

5. Click **Finish**.
   MAW creates a new ACP project with all relevant contents from the JAR file.

Importing Localized Text Files

You can import localized text files to Application Configuration Profile projects.

1. Select **File > Import > Mobile Application Framework > Import localized text**.
2. Specify the target folder for the imported text, or click **Browse** and select it.
3. Specify the file to import, or click **Browse** and select it.
4. Click **Finish**.

The wizard copies the specified file into the target folder.

Importing Style Files

You can import style files into Application Configuration Profile projects.

1. Select **File > Import > Mobile Application Framework > Import style**.
2. Specify the target folder for the imported styles, or click **Browse** and select it.
3. Specify the file to import, or click **Browse** and select it.
4. Click **Finish**.

The wizard copies the specified file into the target folder.

Importing Tile Layouts

You can import tile layouts into Application Configuration Profile projects.

1. Select **File > Import > Mobile Application Framework > Import tile layout from a file**.
2. Select the target Application Configuration Profile (ACP) project.
3. Select the device type for which to import the configuration.

   Mobile Application Workbench analyzes the file, and imports it into the appropriate folders for the target project. The wizard warns you if importing would overwrite any files already in the project, and asks for confirmation.

   Importing divides the single layout file into separate files, each representing a single binding, tile container, or tile. You can edit the files individually, and reassemble them to export the layout or the entire ACP project.

Editors

Use the XML text editor to create and edit bindings, localized text, styles, tile containers, and tile layouts.

Editing Application Configuration Profiles

In the Project Explorer view, you can see the configuration files (layouts, style, images, localized text, and others). Editable configuration files consist of XML elements and their attributes; you can edit these elements directly in the XML editor, via the Properties view.

To specify parameters for UI elements in a tile configuration XML file, use the table editor in the Properties view.

Editing Bindings

The XML text editor displays the Application Configuration Profile data bindings, and lets you edit the property name-value pairs.

The XML configuration defines the binding type, ID, and parameters:

```
<?xml version="1.0"?>
<MAF xmlns="http://schemas.sap.com/maf/2012/cfg">
    <Binding bindingId="LeaveRequests" type="query">
        <P value="LeaveRequestCollection" pid="boType"></P>
        <P value="0" pid="skip"></P>
        <P value="Loading" pid="progressTitle"></P>
        <P value="Downloading Leave Requests" pid="progressMessage"><
        <P value="Error" pid="alertTitle"></P>
        <P value="Leave Requests download failed" pid="alertMessage">
        <P value="Cancel" pid="cancelButtonText"></P>
        <P value="StartDate EQ '{$_concat($_year($_now()), '-01-01T00
    </Binding>
</MAF>
```

The editor offers content assistance to help you specify new parameters. It is connected with the Properties view, which displays an overview of the parameters that you can define. Your selections in the Project Explorer, Outline view, and the editor are all synchronized with the Properties view.

Editing Localized Text

The XML text editor displays the Application Configuration Profile localized text properties, and lets you edit the property name-value pairs.

```
<?xml version="1.0" encoding="UTF-8"?>
<Texts xmlns="http://schemas.sap.com/maf/2013/i18n">
    <Text id="title" value="Leave Requests"/>
    <Text id="btnList" value="Show my leave requests"/>
    <Text id="btnSubmit" value="Submit"/>
</Texts>
```

The IDs are referenced by application code; the values are the text translations. A single localized text file describes the translations for the language-country combination denoted in its file name.

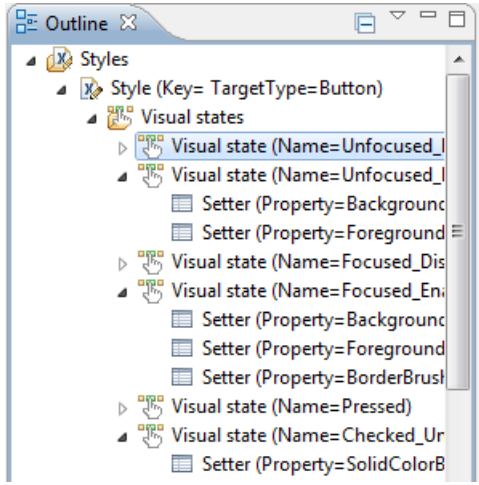
Editing Styles

The XML text editor displays the Application Configuration Profile style configurations, and lets you edit the property name-value pairs.

The editor provides content assistance to help you create the XML tags for the style configuration.

```
<?xml version="1.0" encoding="UTF-8"?>
<styles xmlns="http://schemas.sap.com/maf/2011/sap"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schema
    <!-- ***************** Android colors ***************** -->
    <!-- set color code to 0, if want to hide the corresponding prop
    <Style TargetType="Button" platform="android">
        <VisualStates>
            <VisualState Name="Unfocused_Enabled">
                <Setter Property="Background" Value="#22E01B"/>
                <!-- button color -->
                <Setter Property="Foreground" Value="#FFFFFF"/>
                <!-- text color -->
            </VisualState>
            <VisualState Name="Unfocused_Disabled">
                <Setter Property="Background" Value="#C60404"/>
                <!-- button color -->
                <Setter Property="Foreground" Value="#FFFFFF"/>
                <!-- text color -->
            </VisualState>
            <VisualState Name="Focused_Disabled">
                <Setter Property="Background" Value="#0AC2FA"/>
                <!-- button color -->
                <Setter Property="Foreground" Value="#433211"/>
                <!-- text color -->
                <Setter Property="BorderBrush" Value="#E0D91B"/>
```

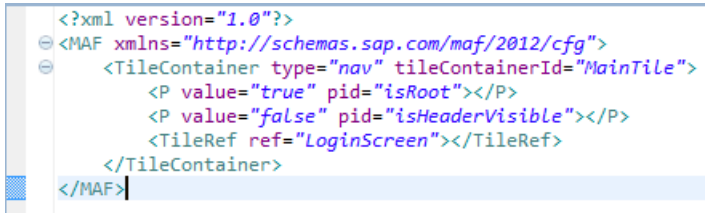
For easier navigation, the style editor is integrated with the Outline view, which displays an overview of the configured properties, styles, and visual states:



Editing Tile Containers

The XML text editor displays the Application Configuration Profile tile-container properties, and lets you edit the property name-value pairs.

The editor helps you create the related XML configuration:



Tile containers define a set of parameters and references to tile layouts. The Mobile Application Workbench provides content assistance for both the parameter IDs and the XML configuration tags. The editor is connected with the Properties view, which provides an overview of the parameters that you can define. Your selections in the Project Explorer, the Outline view, and in the editor are all synchronized with the Properties view for easier editing.

Editing Tile Layouts

The XML text editor displays the Application Configuration Profile tile layouts, and lets you edit the property name-value pairs. You can define the widgets, their layout, and their behavior.

The editor has two tabs:

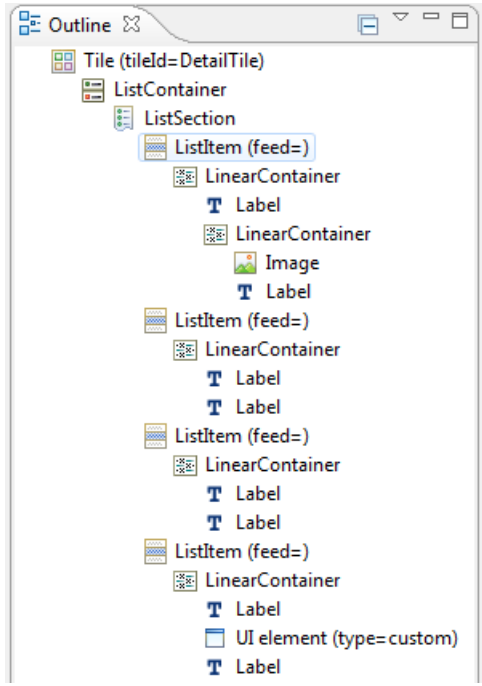
- Source – lets you edit the XML configuration.
- Visual – shows the configured layout.

Source Tab

The Source tab displays the tile layout XML definitions, and offers content assistance for the XML tags and parameter IDs:

```
<?xml version="1.0"?>
<MAF xmlns="http://schemas.sap.com/maf/2012/cfg">
    <Tile tileId="DetailTile">
        <P value="Leave Detail" pid="title"></P>
        <ListContainer>
            <P value="100%" pid="width"></P>
            <P value="grouped" pid="listStyle"></P>
            <P value="groupListBoxGray" pid="style"></P>
            <ListSection>
                <ListItem>
                    <LinearContainer layout="horizontal">
                        <P value="100%" pid="width"></P>
                        <UIElement type="label">
                            <P value="{$_i18n.LEAVE_TYPE}" pid="text"></P>
                            <P value="30%" pid="width"></P>
                            <P value="5%" pid="padding_right"></P>
                            <P value="center" pid="valign"></P>
                            <P value="MAFDetailTitle" pid="style"></P>
                        </UIElement>
                        <LinearContainer layout="horizontal">
                            <P value="65%" pid="width"></P>
                            <UIElement orientation="Portrait" type="image">
                                <P value="Accident.png" pid="imageSrc"></P>
                                <P value="left" pid="halign"></P>
                                <P value="3%" pid="padding_right"></P>
                            </UIElement>
```

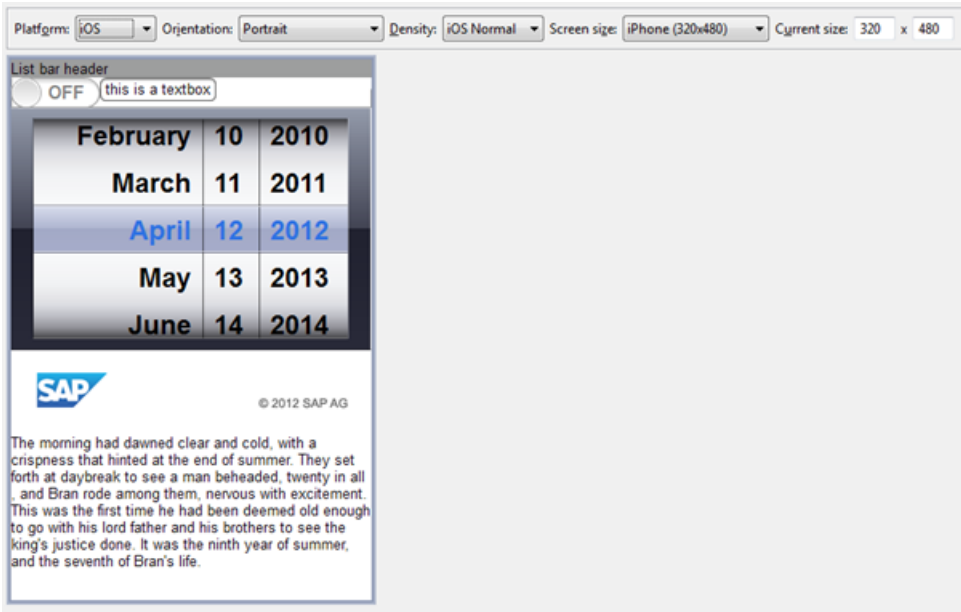
The editor is connected with the Outline view, in which you can see the hierarchy of UI elements and containers:



The editor is connected with the Properties view, which provides an overview of the parameters that can be defined, and allows you to set them. The selections you make in the Project Explorer, Outline view, and the editor are all synchronized with the Properties view, which simplifies common editing tasks.

Visual Tab

On the Visual tab, you can see the configured layout, which is updated as you make edits either on the Source tab or in the Properties view:



At the top of the editor you can specify how the tile appears. You can set these values:

- Plaform – either Android or iOS.
- Orientation – portrait or landscape, and their possible variants (up/down and left/right, respectively).
- Density – pixel density of the screen. The values in this list depend on the selected platform:
  - For iOS, you can select normal or retina densities.
  - For Android, you can select low, medium, high, or extra-high densities.
- Screen size – contains well-known devices (for iOS) or screen sizes (for Android). You can also set an arbitrary screen size using the two text boxes to the right of the list, or by dragging the right or bottom edge, or the lower-right corner of the device screen.

  If you select a screen size from the list, you see how the tile would look on a given device. Setting arbitrary dimensions is helpful when the tile is to be used in embedded mode, that is, when the tile is integrated as a container in the application screen.

A visual preview of the layout may not be accurate, because there can be many factors at runtime that affect the appearance of the configured UI elements. Also, a preview does not reflect changes in the style XML.

Tasks

You can create, edit, import, export, and validate Mobile Application Workbench components.

Creating Application Configuration Profiles

You can create an Application Configuration Profile using Mobile Application Workbench.

1. Select **File > New > Other > Mobile Application Framework > ACP**.
2. Enter the project name and the runtime version.
3. (Optional) Specify an alternate location in the file system.
4. (Optional) Specify the Working sets
5. Click **Finish**.

   The project is created in the Eclipse workspace.

The project is created in the Eclipse workspace.

Creating Binding, Tile, or Tile Container Files

You can create binding files, tile files, and tile container files using Mobile Application Workbench.

1. Select **File > New > Other**.
2. In the selection dialog, click the appropriate file under the Mobile Application Framework category.
3. Click **Next**, and specify file parameters, such as name and location.

   The wizard provides guidance for field entries, and warns about incorrect input.
4. Click **Finish**.

Creating Localized Text Files

You can create localized text files for Application Configuration Profile projects.

1. Select **File > New > Other > Mobile Application Framework > Localized text file**.

   All fields are required.
2. Enter the target folder, or click **Browse** and select it.

   The only folders that appear are those that can contain the file to be generated.

3. Select the runtime version.

   This specifies the template that Mobile Application Workbench (MAW) uses to generate the Application Configuration Profile (ACP). If the runtime version of the file that is generated is not compatible with the runtime version of the enclosing ACP project, MAW displays warnings and errors, so you can fix the problem immediately.
4. Specify the ISO language and the country code.

   MAW appends these codes to the base file name; for example, if you enter `en` as the language code, and `gb` as the country code, the file name is `new_localizedText_en_gb.xml`
5. Click **Finish**.

Creating Style Files

You can configure styles for Application Configuration Profile projects.

1. Select **File > New > Other > Mobile Application Framework > Style file**.
2. Enter the target folder, or click **Browse** and select it.

   The only folders that appear are those that can contain the file to be generated.
3. Select the runtime version.

   This specifies the template that Mobile Application Workbench (MAW) uses to generate the Application Configuration Profile (ACP). If the runtime version of the file that is generated is not compatible with the runtime version of the enclosing ACP project, MAW displays warnings and errors, so you can fix the problem immediately.
4. Click **Finish**.

Exporting Application Configuration Profiles

You can export an Application Configuration Profile project to a JAR file.

1. Select **File > Export > Mobile Application Framework > Export ACP to a JAR file**.

   Before the wizard pops up, Mobile Application Workbench checks for errors in the project. You cannot export projects that contain errors. This prevents the creation of malformed Application Configuration Profile (ACP) projects that the Mobile Application Framework runtime cannot process correctly.
2. Enter the name of the ACP project you want to export, or click **Browse** and select it.
3. Specify the name and location of the destination JAR file; you can click **Browse** to select the location.
4. Click **Finish**.

Exporting Tile Layouts

You can export Application Configuration Profile tile layouts, and add them to mobile applications.

1. Select **File > Export > Mobile Application Framework > Export Layout to a file**.
2. Enter the project name, or click **Browse** and select it.
3. Select the device type, either Phone or Tablet.
4. Specify the export location for the layout, and click **Finish**.

   During export, MAW takes configurations from the relevant directories, including bindings, tile containers, and tiles.

You can add the exported file to a mobile application, and process it using the corresponding API calls in Mobile Application Framework Extensibility.

Importing Application Configuration Profiles

You can import an Application Configuration Profile project from a JAR file.

1. Select **File > Import > Mobile Application Framework > Import ACP from a JAR file**.
2. Specify the JAR file to import, or click **Browse** and select it.

   Mobile Application Workbench (MAW) analyzes the file and recommends a name and version number for the project. The list contains all the versions with which the specified JAR file is compatible. If the **Runtime-Version** attribute is set in the JAR file's `MANIFEST.MF` file, that version number is marked with an asterisk in the list.
3. Keep the recommended project name, or enter another name.
4. Keep the recommended version number, or select another version from the list.

   MAW converts the JAR file to work with the specified Mobile Application Framework variant. If there are unrecognized files and folders in the JAR file, the import wizard asks you to specify how to import each unrecognized item. Possible actions include:
   - Ignore – the file/directory is not copied into the new Application Configuration Profile (ACP) project.
   - Import as-is – the file/directory is copied to the same location in the ACP project as it exissts in the JAR file. This typically results in validation errors in the project, because such files are already known to violate the folder structure. You can use this option when you want to quickly import the JAR file, and plan to move files to their correct locations later.
   - Import into 'ext' – the file/directory is copied into the `ext` directory, which can contain any file type. The directory hierarchy relative to the root of the JAR file is preserved when copying. This action does not cause validation errors in the project.

To set an action for all files and directories at the same time, select the action from the bottom of the list.

5. Click **Finish**.
   MAW creates a new ACP project with all relevant contents from the JAR file.

Importing Tile Layouts

You can import tile layouts into Application Configuration Profile projects.

1. Select **File > Import > Mobile Application Framework > Import tile layout from a file**.
2. Select the target Application Configuration Profile (ACP) project.
3. Select the device type for which to import the configuration.

   Mobile Application Workbench analyzes the file, and imports it into the appropriate folders for the target project. The wizard warns you if importing would overwrite any files already in the project, and asks for confirmation.

   Importing divides the single layout file into separate files, each representing a single binding, tile container, or tile. You can edit the files individually, and reassemble them to export the layout or the entire ACP project.

Importing Other Files

Because the Application Configuration Profile project is stored in the file system as it appears in the Project Explorer, you can import miscellaneous files, such as images, application configurations, styles, and localized text by copying the files into the appropriate folder within the project.

The Eclipse IDE integrates with operating system-specific file services; you can copy a file using a browser application, such as Windows Explorer, then go to the target folder in the Project Explorer view, right-click and select **Paste**.

For localized text and style files, you can use the import wizards under **File > Import > Mobile Application Framework**.

Validating Application Configuration Profiles

Mobile Application Framework semantically and syntactically inspects the contents of each configuration file, and the overall file structure of the Application Configuration Profile project.

To validate consistency, right-click the project, and select **Validate**.

As you modify an Application Configuration Profile, Mobile Application Framework displays errors in the Problems view, and lets you can track the problems' source. The Problems view offers quick fixes for some problems. To accept a quick fix, right-click the error or warning in the Problems view, and select the fix from the menu, or press **Ctrl+1**.

Index

A

B

C

D

E

F

I

L

M

O

P

S

T

U

V

W

X