

# Actions Tech. Note 18

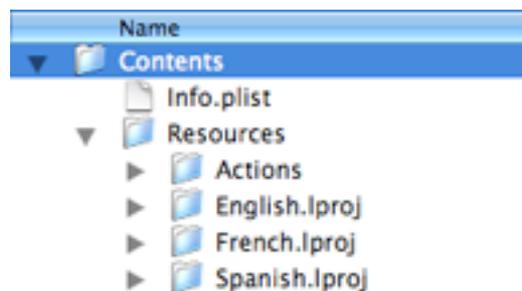
## Draft 1.0

### Localizing Freeway Actions II

Freeway 4 Pro supports multiple localization meaning that one copy of Freeway can be used in many languages. The language used is based on the language you have Mac OS set to. If Actions have been created with localization in mind then this will follow through into your Action Interface and even your Action output.

#### Folder Structure

To set your Action up for localization you will first need to create a folder to store your Action and the associated files in. This folder will be bundled once all of the necessary files have been added to it, it must also have a particular structure to it. It needs to contain a folder named "Contents", which in turn needs to contain a folder named "Resources". This must contain a folder "Actions" for the Action and further folders for each language that you want to represent the Action. These must take the form "Language.lproj" so for instance "English.lproj", "Spanish.lproj", "French.lproj" and so on. There is no minimum number of localization folders you can use but you are limited to the maximum amount of languages included in Mac OS.



1. The folder structure

Once the folder structure has been created you will need to make a Property List file (.plist) for the Contents folder. This will contain important information used to tell the OS information such as the developmental region of the bundle, an identifier for the bundle and so on.

## Creating the pList file

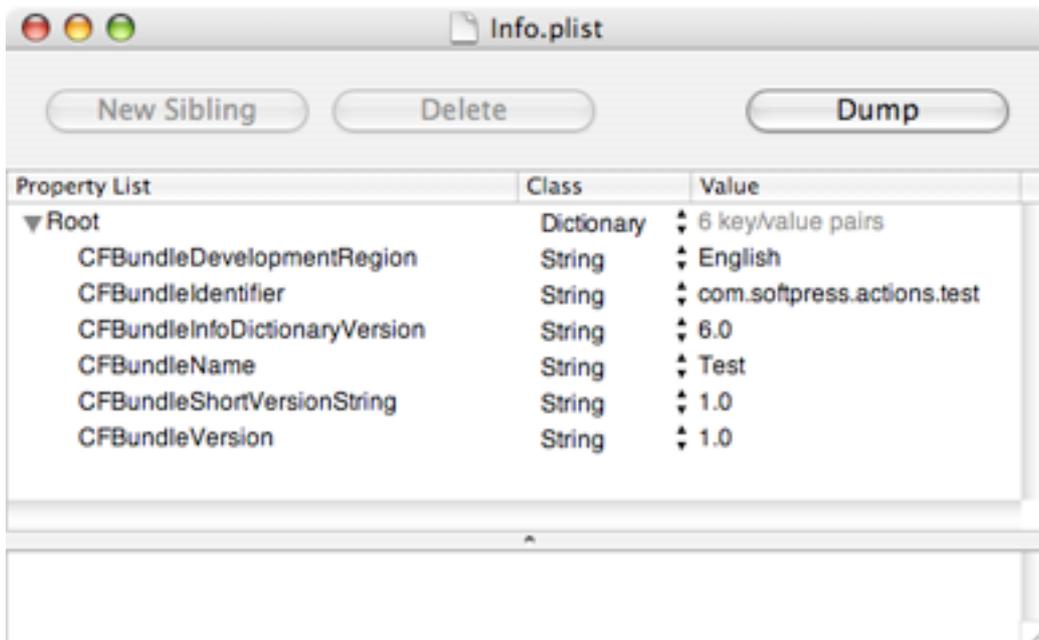
The plist can be created one of two ways. You can use the Property List Editor application (part of the Developer Tools from your OS installation discs). This is really just a glorified XML editor – if you prefer writing your XML rather than selecting it then you may want to ditch this and write your own plist file using your favorite text editor. We recommend using Smultron (<http://smultron.sourceforge.net>), or TextWrangler (<http://www.barebones.com/products/textwrangler>).

### Using Property List Editor:

Upon opening the application you are presented with a new, blank plist window. To start creating the file click the "New Root" button at the top left of the window. This will create the root element of your plist – this will contain the main elements that make up the file. The type of the root is always Dictionary.

Create a new child element of Root by clicking the "New Child" button. Replace the "New Item" text with "CFBundleDevelopmentRegion", leave "String" in the Class column as it is and under the Value column double click and enter "English". Repeat the above for the following elements:

- CFBundleIdentifier                      String            "a unique identifier for your bundle"
- CFBundleInfoDictionaryVersion      String            6.0
- CFBundleName                            String            "the name of your bundle"
- CFBundleShortVersionString          String            "the version of your bundle"
- CFBundleVersion                        String            "the version of your bundle"



2. Screenshot of the completed pList in Property List Editor

Once you have entered all of these save the file in your Contents folder as Info.plist.

## Using a text editor:

Open a new file and type the following XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>CFBundleDevelopmentRegion</key>
  <string>English</string>
  <key>CFBundleIdentifier</key>
  <string>com.softpress.actions.test</string>
  <key>CFBundleInfoDictionaryVersion</key>
  <string>6.0</string>
  <key>CFBundleName</key>
  <string>Test</string>
  <key>CFBundleShortVersionString</key>
  <string>1.0</string>
  <key>CFBundleVersion</key>
  <string>1.0</string>
</dict>
</plist>
```

You will need to replace the `CFBundleIdentifier` string with a unique identifier for your bundle, the `CFBundleName` with the name of your bundle and both version strings with the version of your bundle.

Finally, make sure that your file is set to the correct encoding (UTF-8). In Smultron use the `Text>Text Encoding` menu or in Text Wrangler set the encoding by clicking the Options button when saving.

Save this in your Contents folder as `Info.plist`.

Your Action/s will already have been created. This/these need to be placed in the Actions folder.

## The Localization Folders

Each localization folder must contain a file named `action.strings`. These will contain translations for each of the strings used in your Action. For example, consider the following Action:

```
<action name="com.softpress.testAction" title="Test Action">

  <action-version version="1.0">
    Copyright (c) 2007 Softpress Systems.
  </action-version>

  <action-checkbox name="enable" title="Enable" default="Yes" />
  <action-text name="name" title="Name" />

  <action-javascript>
    function fwBeforeEndHTML()
    {
      if(fwParameters.enable.fwBoolValue)
      {
        item = fwDocument.fwTags.fwFind("", fwItem);
        pTag = item.fwAdd("p", true);
        pTag.fwAddRawOpt("Hello " + fwParameters.name + ".");
      }
    }
  </action-javascript>
</action>
```

The following strings from the Action are displayed in Freeway's interface:

- Test Action
- 1.0
- Copyright (c) 2007 Softpress Systems.
- Enable
- Name

Notice that these are titles rather than names. Names will not get converted, it is always good practice to give your Actions titles.

Since the above strings can all be represented differently in different languages they will all need alternatives for each localization. Starting with the easiest... The `action.strings` file for the `English.lproj` folder will look like the following:

```
"Test Action" = "Test Action";
"1.0" = "1.0";
"'Copyright (c) 2007 Softpress Systems.'" = "'Copyright (c) 2007
Softpress Systems.'";
"Enable" = "Enable";
"Name" = "Name";
```

When creating your own `action.strings` file make sure that the encoding of the file is set to UTF-8. In Smultron use the `Text>Text Encoding` menu, in Text Wrangler set the encoding by clicking the `Options` button when saving.

An action.string file is then needed for each localization folder. For the French and Spanish examples for this Action the files will have the following form:

French:

```
"Test Action" = "Exemple d'action";
"1.0" = "1.0";
"'Copyright (c) 2007 Softpress Systems.'" = "'Copyright (c) 2007
Softpress Systems.'";
"Enable" = "Activer";
"Name" = "Nom";
```

Spanish:

```
"Test Action" = "Acción de Prueba";
"1.0" = "1.0";
"'Copyright (c) 2007 Softpress Systems.'" = "'Copyright (c) 2007
Softpress Systems.'";
"Enable" = "Activar";
"Name" = "Nombre";
```

If the OS language is set to either of these languages when using the Action then the strings listed on the left will be replaced by those on the right.

Once all of the files have been created rename the containing folder to "name.fwactionsb" where name should be replaced with the name used in CFBundleName in the plist file. The b at the end of the suffix signifies that you are creating an Action bundle.

The above does not account for any output used in your Actions. The example Action above outputs a line of text that says "Hello (name entered)". It is possible, using the fwGetString() function, to also change the word Hello based upon the localization of the machine.

## Using the fwGetString() function

Consider the amended fwBeforeEndHTML() function of the previous Action:

```
function fwBeforeEndHTML()
{
    str = fwGetString("Hello");

    if(fwParameters.enable.fwBoolValue)
    {
        item = fwDocument.fwTags.fwFind("", fwItem);
        pTag = item.fwAdd("p", true);
        pTag.fwAddRawOpt(str + " " + fwParameters.name + ".");
    }
}
```

In this code the variable `str` is assigned the value of `fwGetString("Hello")`. This looks at the localization of the machine and returns the string matching the argument from the `action.strings` file in the folder matching the localization of the machine.

For this to work however, all of the `action.string` files in the localization folders need to be updated with new entries for the string "Hello". This can easily be done by control/right clicking in the bundle that you created, selecting "Show Package Contents" option and opening and amending the `action.strings` files using your text editor. No other changes are necessary.