# OpenNTF Project: XPages Mobile Controls

http://www.openntf.org/Projects/pmt.nsf/ProjectLookup/XPages%20Mobile%20Controls

License: Apache License v2
Last version: 05/04/10
Author: Niklas Heidloff, Steve Castledine
Version: 1.6.0

1. Reusable XPages Controls leveraging Dojo Mobile 1.5
Platform: Lotus Domino 8.5.1
iPod Touch 3. generation, Apple Safari 4.0.4,
[Firefox 3.6]

2. XPages sample app leveraging Domino 8.5.1 only
Platform: Lotus Domino 8.5.1,
Android 2.0 (Motorola Milestone),
iPod Touch 3. generation, Apple Safari 4.0.4,
Blackberry 9550 simulator running in PhoneGap (not standalone browser)
[Firefox 3.6]

4. XCamera
Platform: Lotus Domino 8.5.1,
PhoneGap 0.9.0
Android 2.0 (Motorola Milestone), Android SDK 2.0 simulator

5. Offline samples
Platform: Lotus Domino 8.5.2 (starting code drop 4),
Android 2.0 (Motorola Milestone), Android SDK 2.0 simulator,
iPod Touch 3. generation, Apple Safari 4.0.4,

**Description**

This project contains these five parts:

1. Reusable XPages Controls leveraging Dojo Mobile 1.5

2. XPages sample app leveraging Domino 8.5.1 only

3. Login control for mobile apps (using Lotus iNotes Ultralite)

4. XCamera

5. Offline samples

Both (1) and (2) support the following functionality:

1. View control to display N view entries. The next N view entries can be loaded manually when pressing the 'more' button. On webkit based browsers they are loaded automatically when the users scroll to the bottom of the page

2. Document control that displays a specific blog entry. Specific documents can be bookmarked

3. Transitions between the view page and the document page. Loading status indications when loading view entries and documents. Browser navigation buttons are supported

4. Ability to see response documents of a document and create a new response

Technically one aspect that is demonstrated is the ability to use Ajax to do 1. view paging and 2. opening documents. No browser refresh is done. The sample also shows various other smaller things like a stylesheets that looks similar to native apps and how to hide the address bar.

The # is used in the URLs to to be able to Ajax and no browser refreshs but at the same time keep bookmarks. In order to keep browser navigation via back and forward the Dojo 1.4 class dojo.hash is used and pulled from http://o.aolcdn.com/dojo/1.4/dojo/hash.js .

For the view paging HTTP requests are done in this format: http://.../All?ReadViewEntries&Start=5&Count=5&OutputFormat=JSON

# 1. Reusable XPages Controls leveraging Dojo Mobile 1.5

MobileControls1.0.0.nsf leverages code from Dojo 1.5 (dojo/dojox/mobile). Since there is no 1.5 build yet and no IBM cleared version, you have to download this code first from dojotoolkit (http://svn.dojotoolkit.org/src/dojox/trunk) and import it into MobileControls1.0.0.nsf. The instructions are below.

In order to run the sample copy the NSF MobileControls1.0.0.nsf into your Notes data directory, open it in Designer, select the XPage 'm' and choose 'Design-Preview in Safari Apple'.

**Screenshots**


CARRIER
7:04 AM
OpenNTF in March
Featured OpenNTF Project of the Month: Wildfire
OpenNTF Mobile App for Blackberry, iPhone and Android
New Release of XPages Demo Application for 8.5.1 including Notes Client
LotusScript to manipulate XPages and other file-based Elements
Loading ...

**Setup instructions**

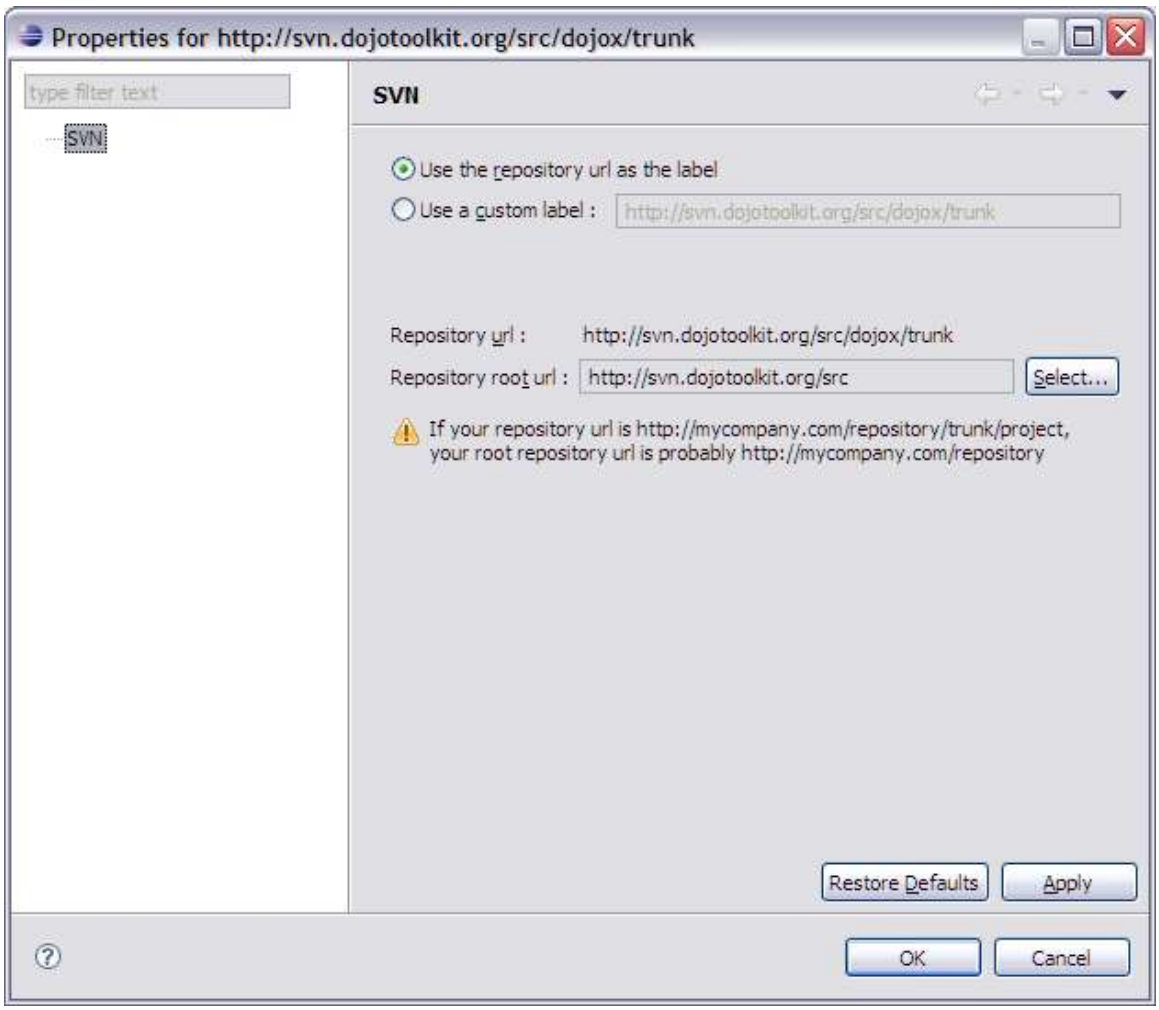
Download the Dojo Mobile 1.5 code from the Dojo SVN repository:

The Dojo Mobile code works with Dojo 1.3.2 (Domino 8.5.1) and Dojo 1.4.1 (Domino 8.5.2).

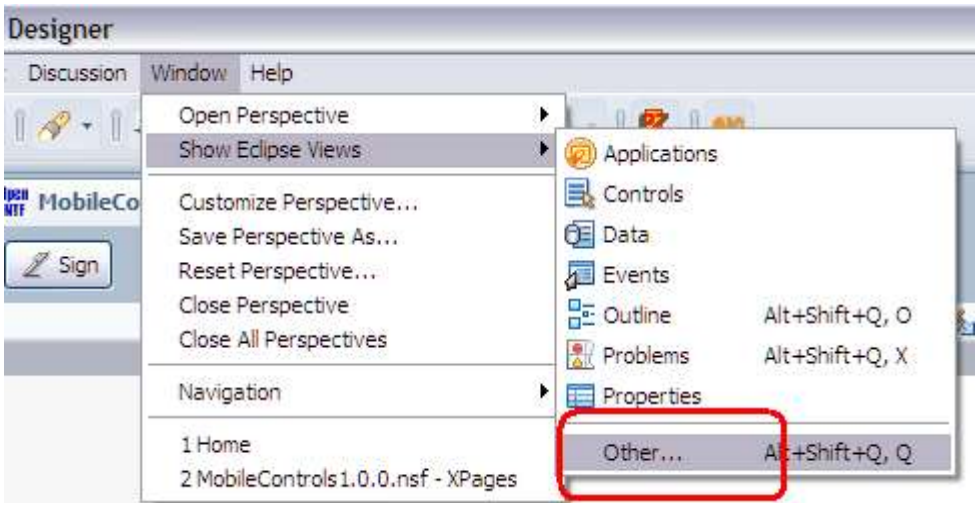
Get the code from here with your favorite SVN client:

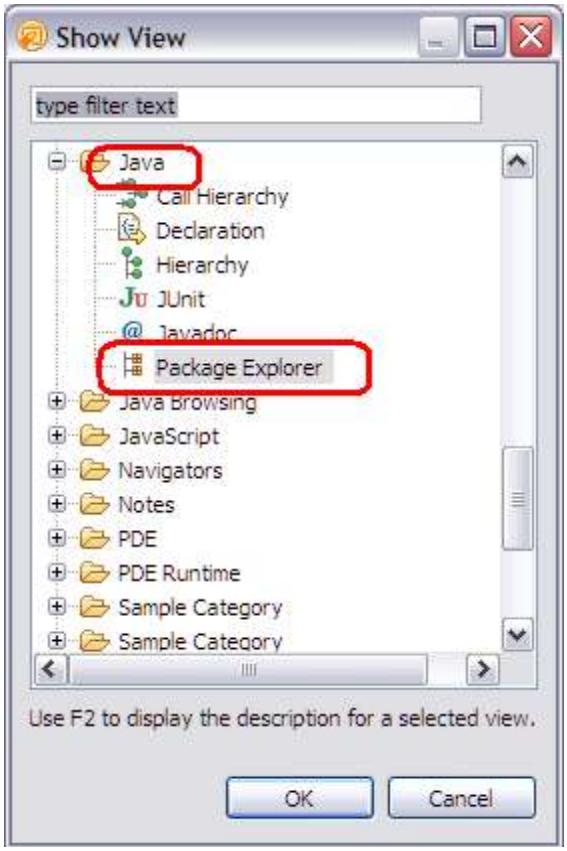
http://svn.dojotoolkit.org/src/dojox/trunk

Example:

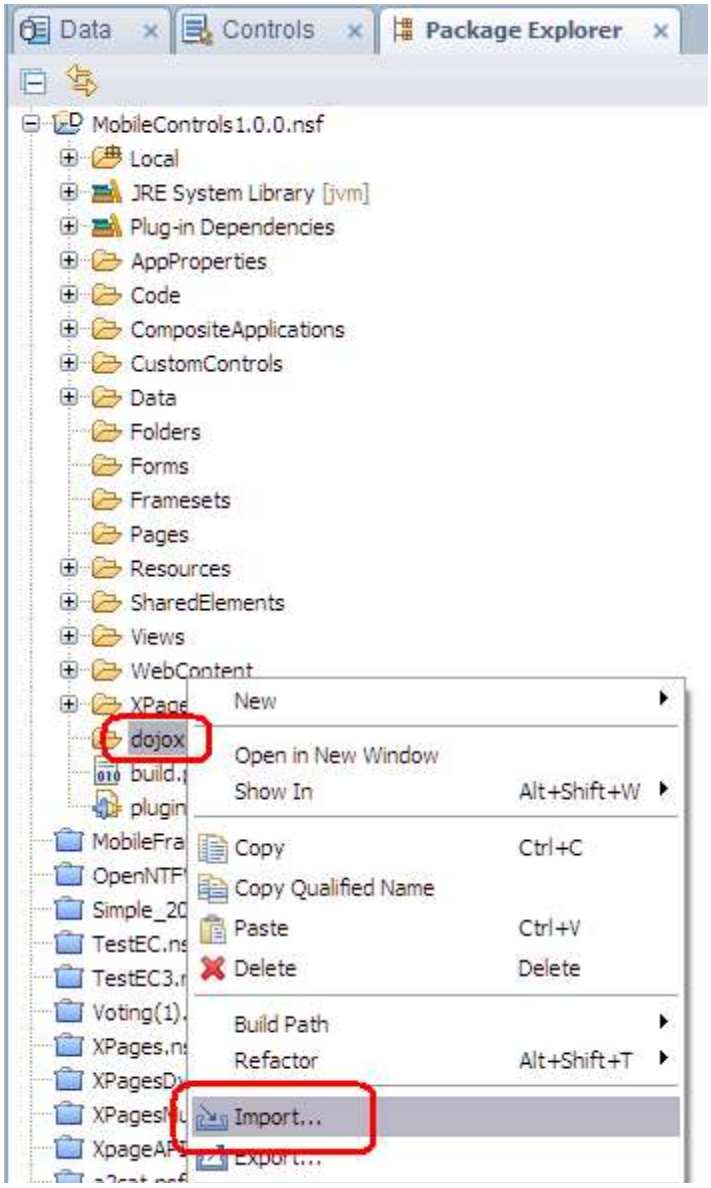


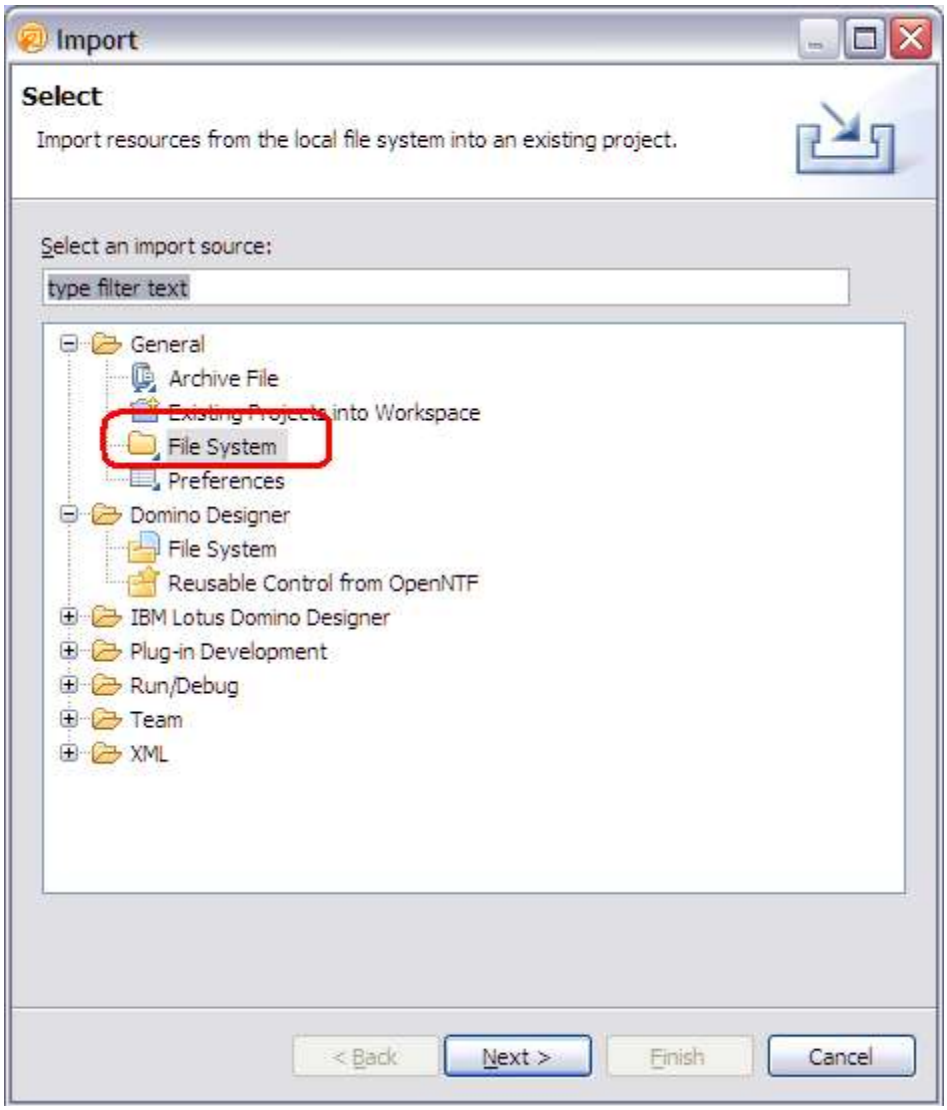
Open Designer and then the package explorer:



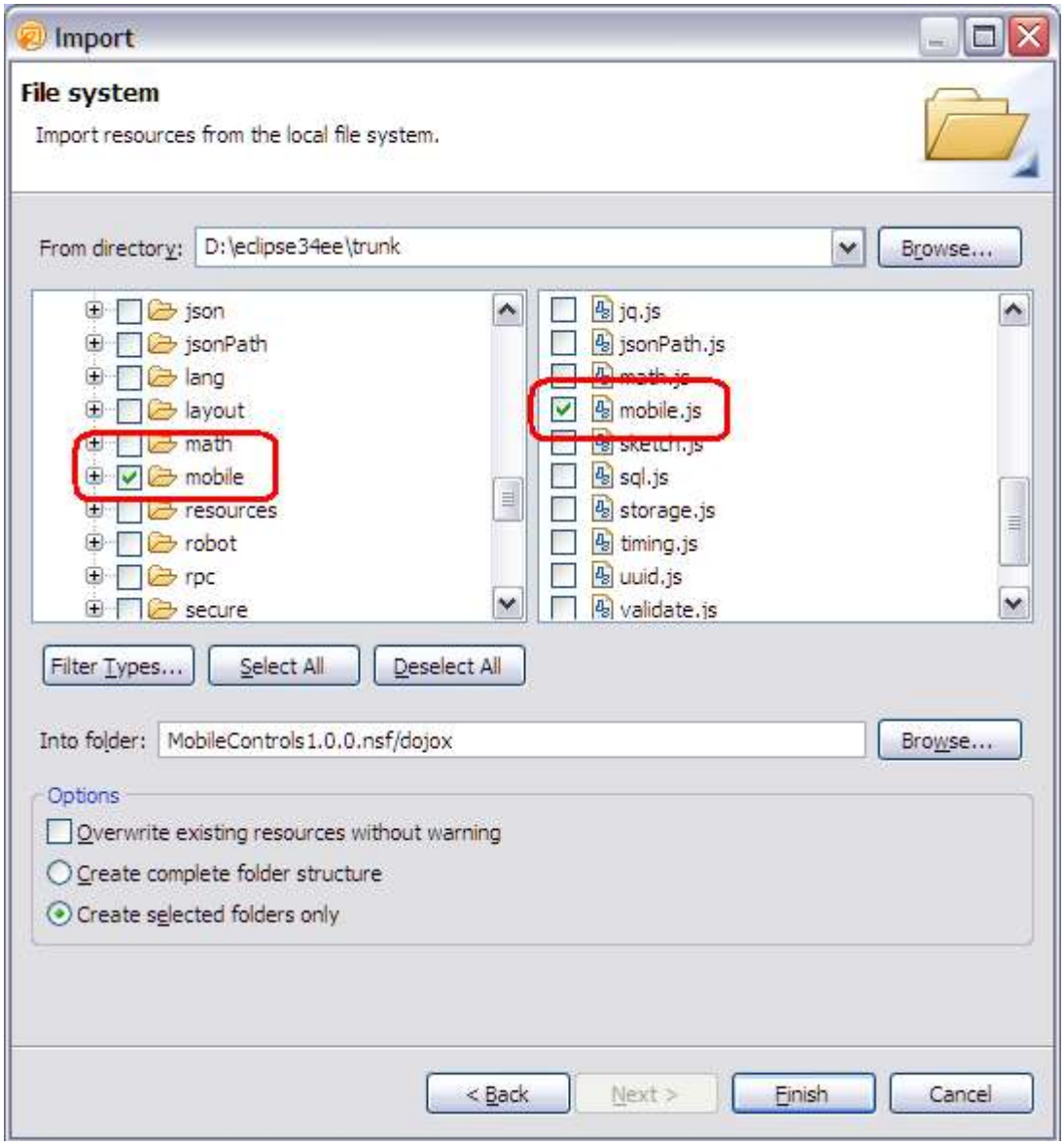



Open the project in the package explorer, right click dojox and choose import:

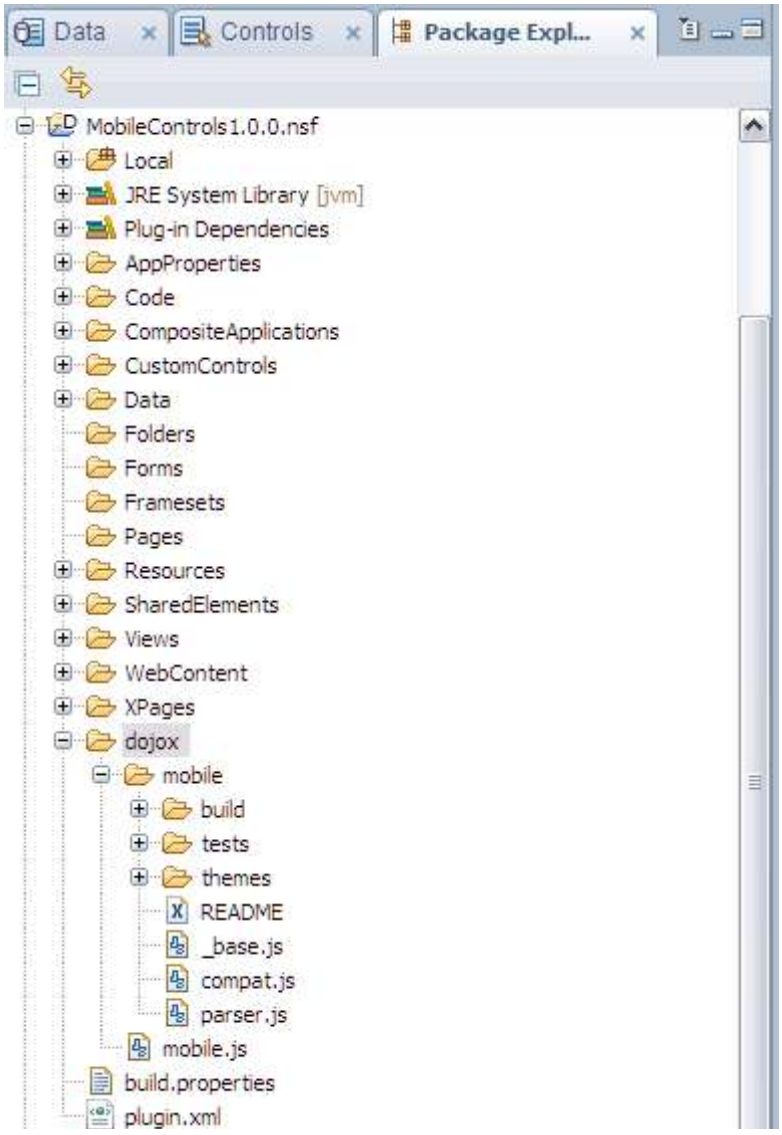



Import
Select
Import resources from the local file system into an existing project.
Select an import source:
type filter text
General
Archive File
Existing Projects into Workspace
File System
Preferences
Domino Designer
File System
Reusable Control from OpenNTF
IBM Lotus Domino Designer
Plug-in Development
Run/Debug
Team
XML
< Back
Next >
Finish
Cancel

Choose the mobile directory and the file mobile.js:



You should see this now:

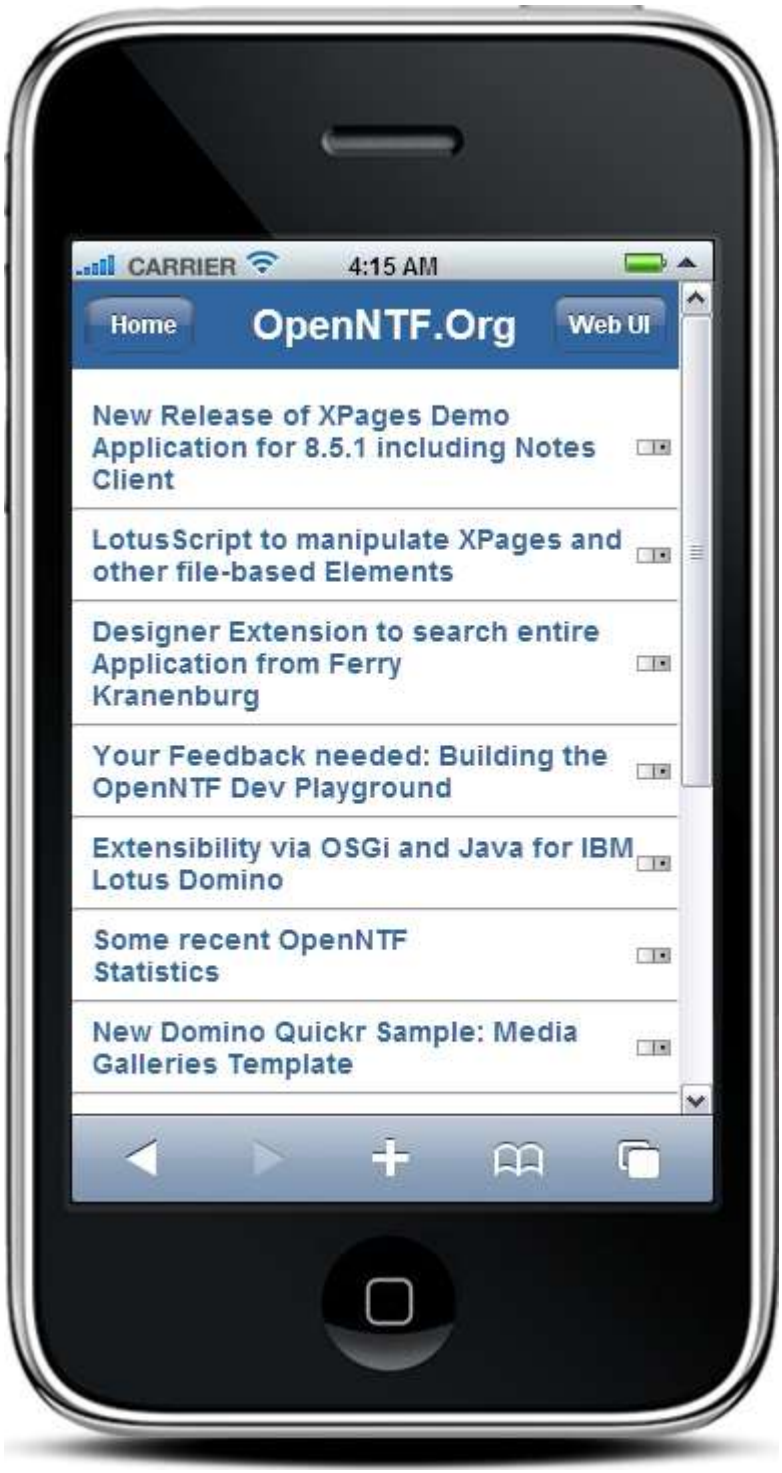


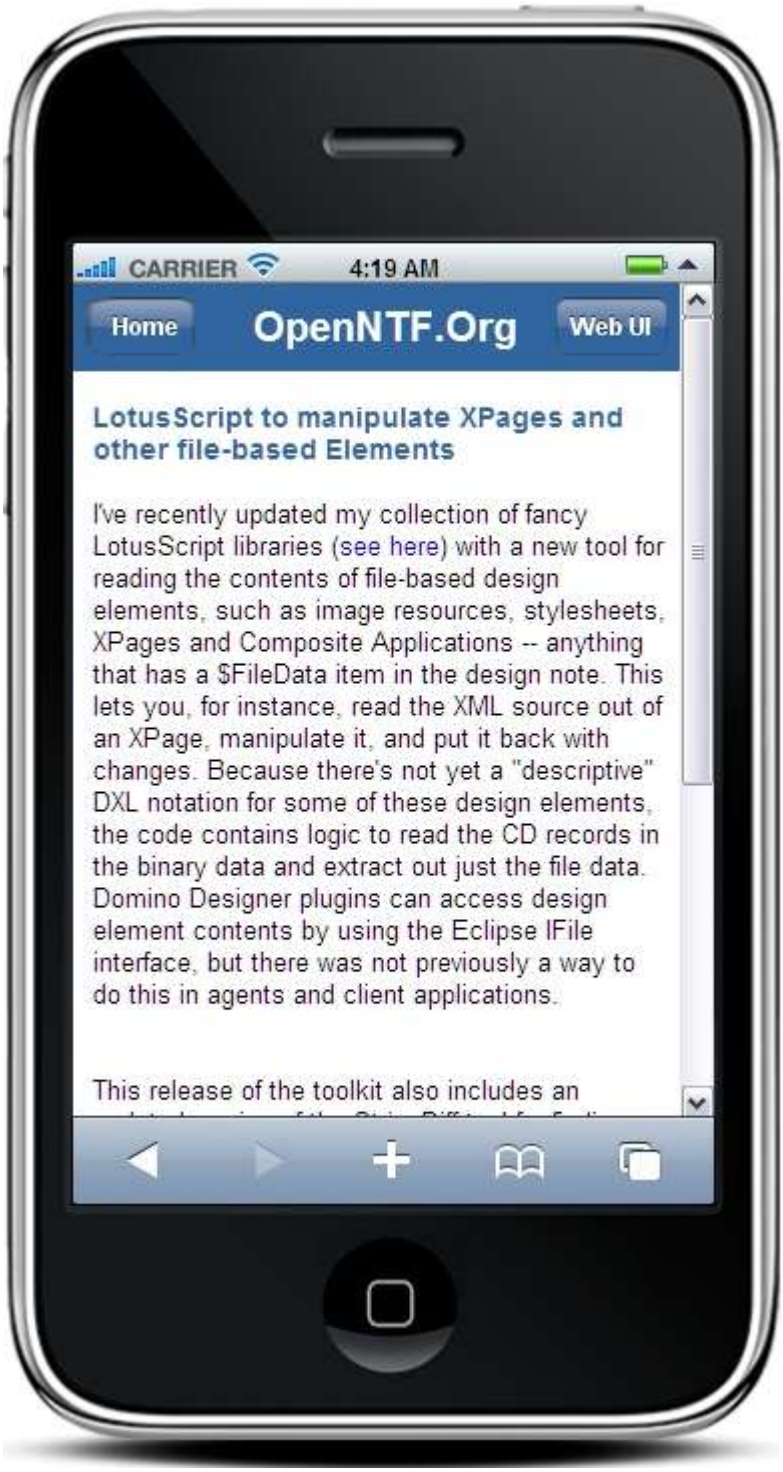
## Usage of the Controls

```
<?xml version="1.0" encoding="UTF-8"?>
<xp:view xmlns:xp="http://www.ibm.com/xsp/core"
	xmlns:xc="http://www.ibm.com/xsp/custom">
	<xc:mConfiguration
		stylesheetUrl="dojox/mobile/themes/iphone/iphone.css">
	</xc:mConfiguration>
	<xc:mPage name="home" visible="true">
		<xp:this.facets>
			<xp:panel
				xp:key="pageContent">
				<xc:mHeader title="OpenNTF.org">
				</xc:mHeader>
				<xc:mRectangle>
					<xp:this.facets>
						<xp:panel xp:key="rectangleContent">
							<xc:mView
								viewName="All"
								databaseName=""
							    targetPageName="document"
								entriesPerRequest="10"
								autoPaging="true">
							</xc:mView>
						</xp:panel>
					</xp:this.facets>
				</xc:mRectangle>

			</xp:panel>
		</xp:this.facets>
	</xc:mPage>

	<xc:mPage name="document" visible="false" isDocumentPage="true">
		<xp:this.facets>
			<xp:panel
				xp:key="pageContent">

				<xc:mHeader
					title="OpenNTF.org"
					backPageName="home"
					backTitle="Home">
				</xc:mHeader>

				<xc:mDocumentContainer
					databaseName="">
					<xp:this.facets>
						<xc:myDocument
							xp:key="documentContent"
							viewName="All"
							databaseName="">
						</xc:myDocument>
					</xp:this.facets>
				</xc:mDocumentContainer>
			</xp:panel>
		</xp:this.facets>
	</xc:mPage>
</xp:view>
```

# 2. XPages Sample App leveraging Domino 8.5.1 only

**Screenshots**




CARRIER
4:19 AM
Home
OpenNTF.Org
Web UI
LotusScript to manipulate XPages and other file-based Elements
I've recently updated my collection of fancy LotusScript libraries (see here) with a new tool for reading the contents of file-based design elements, such as image resources, stylesheets, XPages and Composite Applications -- anything that has a $FileData item in the design note. This lets you, for instance, read the XML source out of an XPage, manipulate it, and put it back with changes. Because there's not yet a "descriptive" DXL notation for some of these design elements, the code contains logic to read the CD records in the binary data and extract out just the file data. Domino Designer plugins can access design element contents by using the Eclipse IFile interface, but there was not previously a way to do this in agents and client applications.
This release of the toolkit also includes an


BlackBerry
OpenNTF.Org
OpenNTF.Org
Home
New Release of XPages Demo Application for 8.5.1 including Notes Client
LotusScript to manipulate XPages and other file-based Elements
Designer Extension to search entire Application from Ferry Kranenburg
Your Feedback needed: Building the OpenNTF Dev Playground
Extensibility via OSGi and Java for IBM Lotus Domino
Some recent OpenNTF Statistics
New Domino Quickr Sample: Media Galleries Template
Domino Login Control for Mobile Apps

In order to run the sample copy the NSF MobileControls040.nsf into your Notes data directory, open it in Designer, select the XPage 'm' and choose 'Design-Preview in Safari Apple'.

**Usage of this Control**

The code hasn't really been parameterized/modularized yet. You can reuse the custom control mMainPage for the view control. The document control however is specific to blog documents. For documents you have to change mDocument and SaveComment.xsp.

Samples:

```
<xc:mMainPage
	homePageUrl="#n=home"
	saveDocUrl="SaveComment.xsp"
	viewName="All">
</xc:mMainPage>
```

```
<xc:mMainPage
	homePageUrl="http://m.openntf.org/#n=home"
	saveDocUrl="/Internal/homemobile.nsf/SaveComment.xsp"
	viewName="AllMobile3"
	databaseName="blogs/openntf.nsf">
</xc:mMainPage>
```

# 3. Login Control for Mobile Apps

Lotus Domino allows customizing the login page that is used for session based authentication. Here is an example:

IBM Lotus iNotes comes with an optimized login page for the iPhone and other webkit based browsers. The same core functionality can be used to build custom login pages for custom mobile apps.

This video describes the necessary setup steps:

http://public.dhe.ibm.com/software/dw/lotus/learn/viewlets/notesdomino/inotes851/inotes_redirect_851.swf

There is also documentation ...
http://www-01.ibm.com/support/docview.wss?rs=3651&context=SSULMR&dc=DB560&dc=DB520&uid=swg21326762&loc=en_US&cs=UTF-8&lang=en&rss=ct3651lotus

... and an article:
http://www.intranetjournal.com/articles/200807/ij_07_22_08a.html

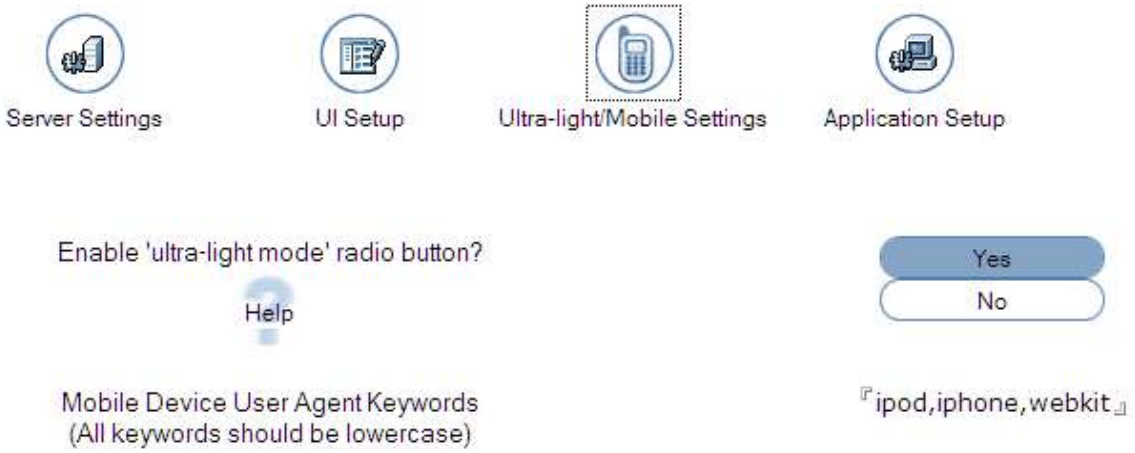
**Step 1:** Create an IBM Lotus iNotes redirect application

Follow the step 1 in the video.

In the simplest case use fixed redirect type to your server, e.g. http://nheidloff-1.

In order to allow the password to be stored overwrite the default 'enable login options' and set it to 'true'.

Choose the 'ultra-light mode' and add your mobile devices, e.g. 'webkit'.



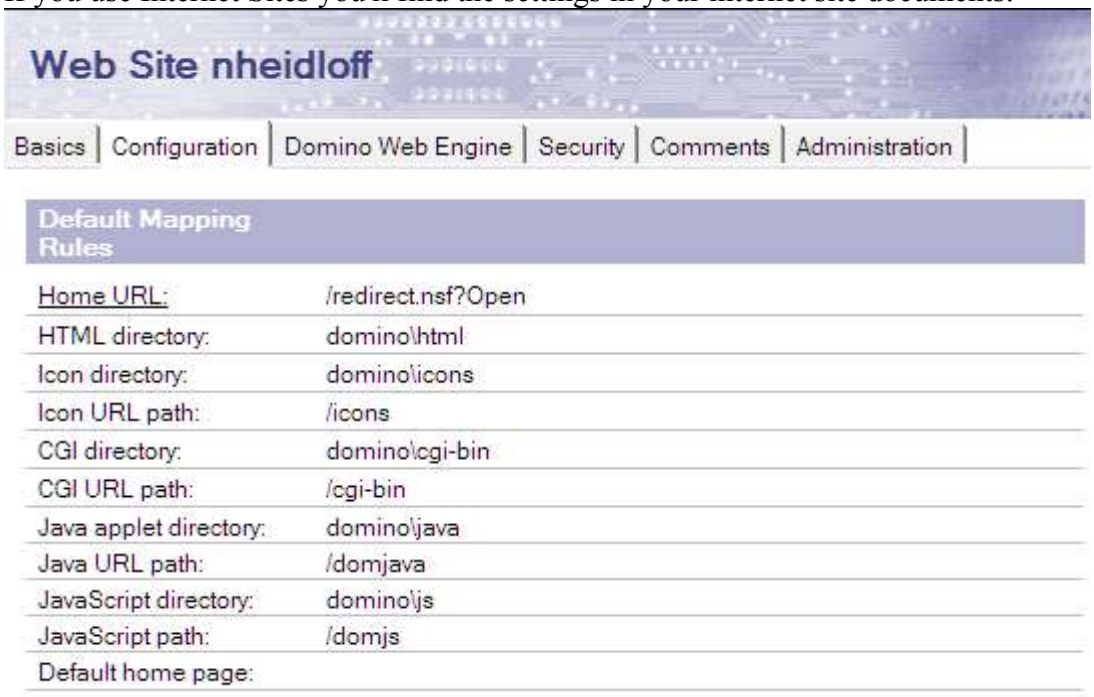
**Step 2:** Create a Domino web server configuration application that maps the Redirector to the web server

Follow the step 2 in the video.

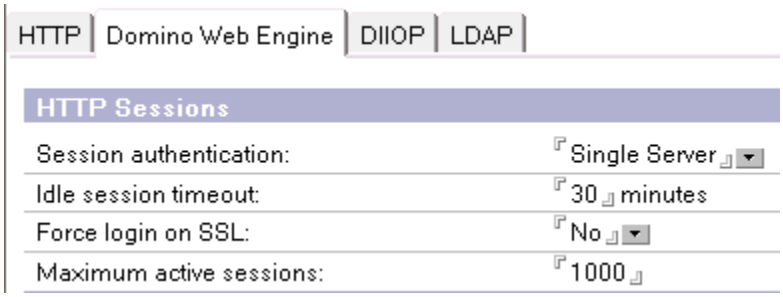
As target form define 'OpenNTFWebkitLoginForm':


**Step 3:** Edit the current server document and map the home URL to the redirector application

If you use Internet Sites you'll find the settings in your internet site documents:



Also make sure you have selected session based authentication:




**Step 4:** Copy WebKit page with OpenNTF branding

Paste the one form and one subfrom from 'login.nsf' in this project's zip file into redirect.nsf.

OpenNTFWebkitLoginSubForm

OpenNTFWebkitLoginForm

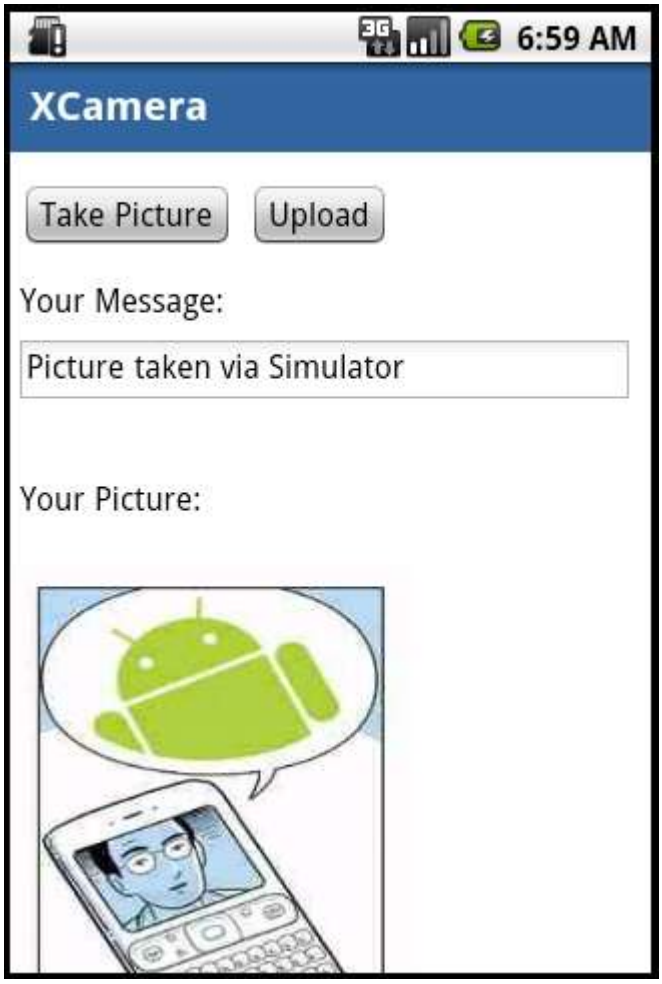
You can modify these forms for your own branding.

# 4.XCamera

XCamera is an app leraging PhoneGap to access the camera of a smartphone. You can take a picture and then upload it into a Notes database.

This app has been tested with Android but PhoneGap is also available for iPhone and Blackberry.

Screenshot:



**Setup of Android SDK and PhoneGap**

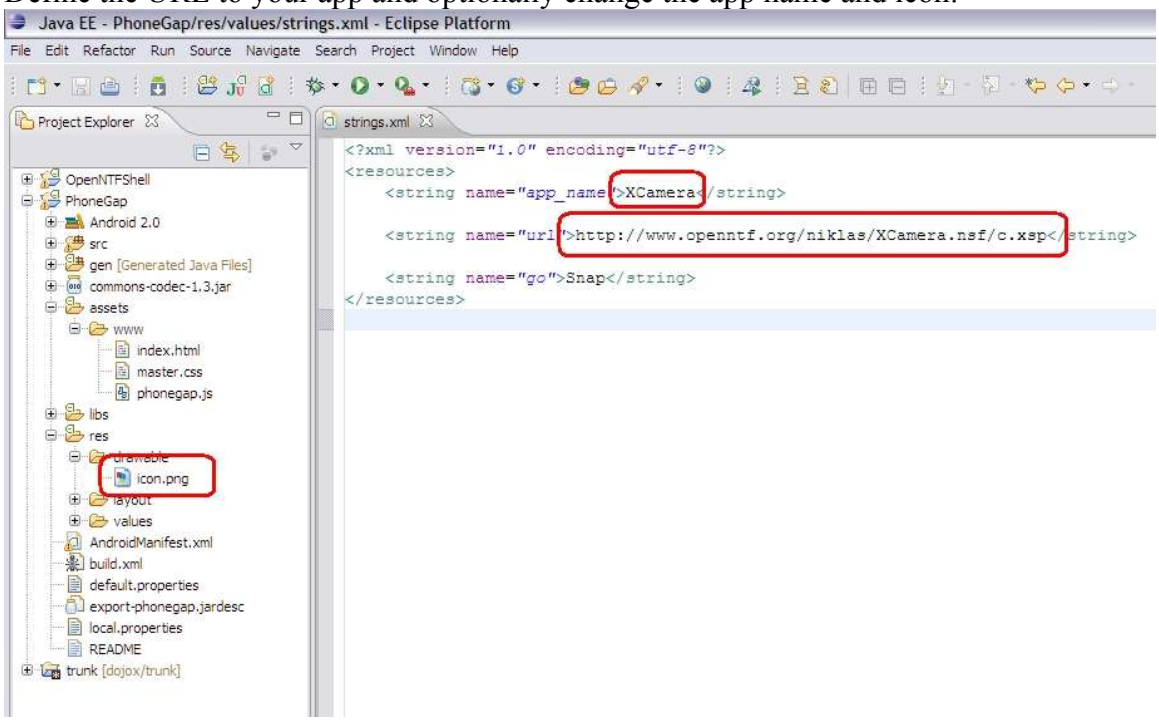
Download and install the Android SDK: http://developer.android.com/sdk/index.html

Download PhoneGap 0.9.0 from here: http://www.phonegap.com/download

Import PhoneGap as existing project into the Android SDK

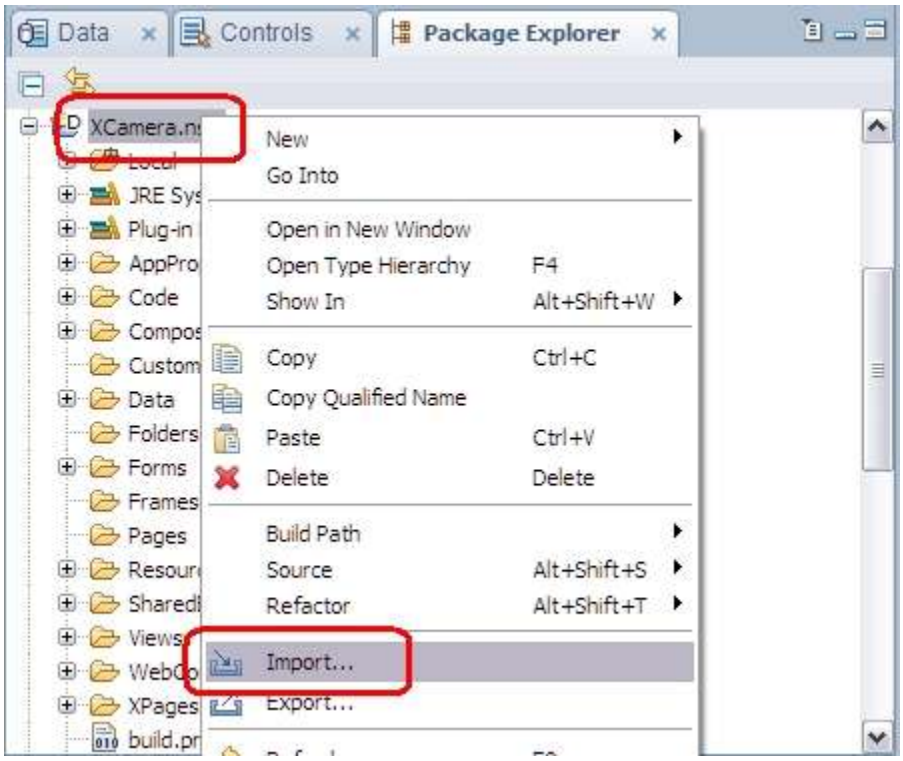
[Here is some more info: http://phonegap.pbworks.com/Getting-started-with-Android-PhoneGap-in-Eclipse]

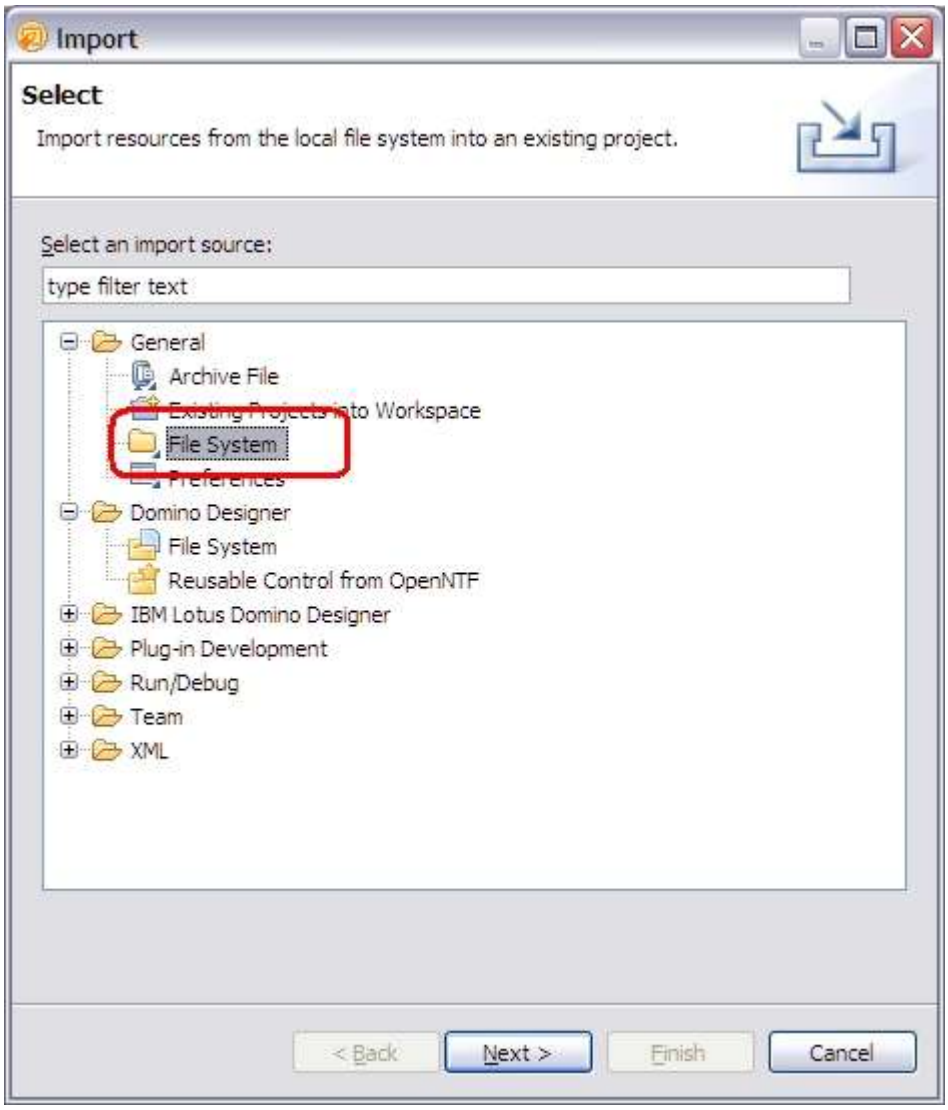
Define the URL to your app and optionally change the app name and icon:



**Setup of NSF and Domino Designer**

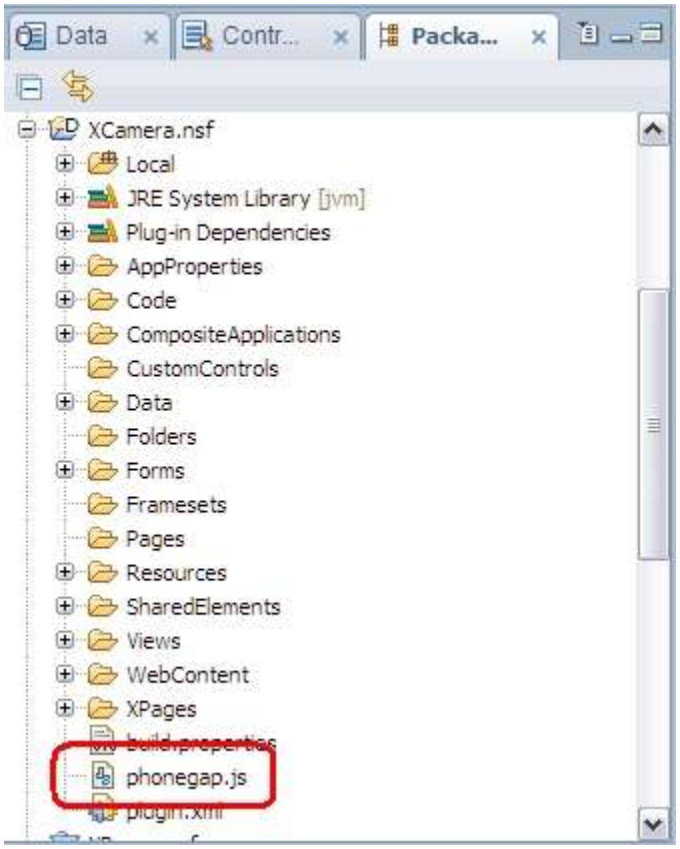
You need to import the phonegap.js file:




Import
Select
Import resources from the local file system into an existing project.
Select an import source:
type filter text
General
Archive File
Existing Projects into Workspace
File System
Preferences
Domino Designer
File System
Reusable Control from OpenNTF
IBM Lotus Domino Designer
Plug-in Development
Run/Debug
Team
XML
< Back
Next >
Finish
Cancel


Import
File system
Import resources from the local file system.
From directory:
D:\eclipse34ee\android\PhoneGap\assets\www
Browse...
www
index.html
phonegap.js
Filter Types...
Select All
Deselect All
Into folder:
XCamera.nsf
Browse...
Options
Overwrite existing resources without warning
Create complete folder structure
Create selected folders only
< Back
Next >
Finish
Cancel

After import it should look like this:



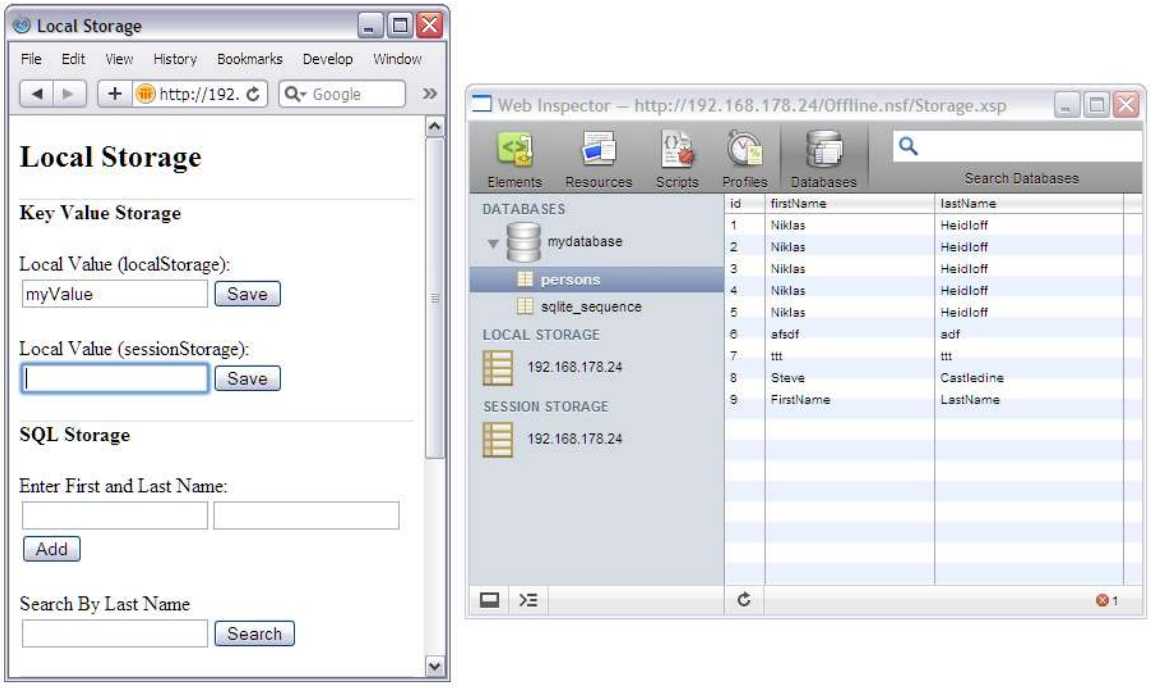
# 5. Offline Samples

This video describes the offline samples leveraging HTML5 and Domino 8.5.2: http://www.youtube.com/watch?v=XkFWYHO-1ek

See also here for information about offline in HTML5: http://www.w3.org/TR/html5/offline.html

In order to run the samples configure your Domino server (see video) and open the two XPages:
http://192.168.178.24/Offline.nsf/Resources.xsp
http://192.168.178.24/Offline.nsf/Storage.xsp

Data can be stored locally in the browser:



Resources can be cached and made available when offline: