XPages Mobile Controls

http://mobilecontrols.openntf.org
Get Started with Mobile Controls 4.5

License: Apache License v2
Last version: 05/06/11
Author: Niklas Heidloff (Frank van der Linden contributed iPad sample)
Version: 4.5.0
Tested platforms:
  Domino 8.5.2
  iPod Touch 3. generation, Apple Safari 4.0.4,
  Android 2.0 (Motorola Milestone), Android SDK 2.0 simulator

In order to get started copy MobileControlsDojo16.nsf in your Domino data directory and sign it. Then open the XPages, e.g. ViewAndDocument.xsp. This database contains Dojo 1.6 Mobile and everything else you need to run your first sample.


In order to build your own first mobile app open the XPage 'ViewAndDocument.xsp' and change the references to the database and view. You could copy the mobile controls in your own NSF but at this point it is not recommended since not documented.

To render your own view you only need to change the database and view name. Since the document is not rendered generically you need to overwrite 'MyDocument'.

Watch this video to see how to build your own first app: [http://www.youtube.com/watch?v=n7e-48JuHos&feature=player_embedded](http://www.youtube.com/watch?v=n7e-48JuHos&feature=player_embedded)

There are also many screenshots, videos and presentations. See the different subdirectories in the downloaded zip file.

You can also try the controls yourselves via these live demos: [http://208.85.190.65/demos/DemoMobileControls.nsf/StartHere.xsp](http://208.85.190.65/demos/DemoMobileControls.nsf/StartHere.xsp)
Overview

1. MobileControlsDojo16.nsf

This database contains the main controls and is the database you should try first. It contains Dojo 1.6 Mobile and everything else to get started.

2. MobileControlsWinkCubiq.nsf

This database contains additional controls based on open source from cubiq.org and the Wink Toolkit (not Apache Wink).

3. MobileControlsOffline.nsf and Offline.nsf

Offline.nsf contains basic HTML5 samples for how to do offline with XPages. MobileControlsOffline.nsf shows how to take a Dojo based mobile app offline.

4. XCamera.nsf

XCamera.nsf contains a sample how to take a picture with a smartphone and upload it to an NSF. This one is based on PhoneGap and requires PhoneGap to be used to compile a native hybrid app.

5. Login.nsf

This database contains some code for a login control based on the iNotes login mechanism.
1. MobileControlsDojo16.nsf

This database contains the main controls and is the database you should try first. It contains Dojo 1.6 Mobile and everything else to get started.

Here is a list of controls:

- mPage: the complete app needs to be put on one XPage to get the nice page transitions. On the one XPage multiple mPages can be used for the different user pages
- mView: generic view control which you can point against your own view
- mDocumentContainer: handles passing of the UNID and manages the loading indicator
- mConfiguration: loads all necessary JS and CSS resources
- mInitialize: needs to be appended to your XPage to run the Dojo parser
- mRectangle, mRectangleEntry and mRectangleHeader are some simple helper controls (which could have also done via CSS)

There are a couple of samples:
- ViewAndDocument: open documents from a view
- ViewAndDocumentCustomStyle: same with custom style
- RectangleSimpleEntries: static list items
- Transitions: transitions between pages
- CreateDocument and SaveDocument: creation of new documents
- ViewAndEditDocument and SaveComment: edit documents
- NavigatorAndViewAndDocument: iPad sample with navigator, view and document
- ViewOneCategoryAndDocument: filter to one category of a view
- ViewWithCategoriesAndDocument: view with categories
- MultiDatabases and MyDocumentOtherDb: since there are some issues with multiple databases, this sample shows a workaround
- InitializeData: adds more lorem ipsum documents


In order to build your own first mobile app open the XPage 'ViewAndDocument.xsp' and change the references to the database and view. You could copy the mobile controls in your own NSF but at this point it is not recommended since not documented.

To render your own view you only need to change the database and view name. Since the document is not rendered generically you need to overwrite 'MyDocument'.

Watch this video to see how to build your own first app:
http://www.youtube.com/watch?v=n7e-48JuHos&feature=player_embedded
Optimization

In order to minimize the size of data of the resources you can also put the resources on the Domino server. Then the gz'd files can be loaded. This brings down the size to 58.9 KB (from originally 628.8 KB).

<table>
<thead>
<tr>
<th>URL</th>
<th>Status</th>
<th>Domain</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET m.xsp#home</td>
<td>200 OK</td>
<td>nheidloff-1</td>
<td>1.3 KB</td>
</tr>
<tr>
<td>GET iphone.css</td>
<td>200 OK</td>
<td>nheidloff-1</td>
<td>3.1 KB</td>
</tr>
<tr>
<td>GET dojo.js</td>
<td>200 OK</td>
<td>nheidloff-1</td>
<td>37.9 KB</td>
</tr>
<tr>
<td>GET xspClientDojo.js</td>
<td>304 Not Modified</td>
<td>nheidloff-1</td>
<td>14 KB</td>
</tr>
<tr>
<td>GET heading-bg.png</td>
<td>200 OK</td>
<td>nheidloff-1</td>
<td>207 B</td>
</tr>
<tr>
<td>GET mobileControls.js</td>
<td>200 OK</td>
<td>nheidloff-1</td>
<td>2.4 KB</td>
</tr>
</tbody>
</table>

In order to use the gz'd version extract the files from mobile.zip here:

Note that the NSF does not contain the full Dojo 1.6. Instead a custom build has been provided with the minimal set of Dojo 1.6 files, primarily from dojox/mobile. See the build script in mobile-all.profile.js to find out which files.
2. MobileControlsWinkCubiq.nsf

This database contains controls from cubiq.org and the Wink toolkit. It contains the controls from cubiq.org but you have to import the Wink toolkit manually (see below for description).

**SlideInMenu and SpinningWheel**

See screenshots on OpenNTF.

These controls use code from cubiq.org from Matteo Spinelli under the MIT license:
http://cubiq.org/slide-in-menu
http://cubiq.org/spinning-wheel-on-webkit-for-iphone-ipod-touch

Copied from cubiq.org on 06/30/10:
<<
Project state: ACTIVE (code is actively updated)
Last code update: 2010.05.28 – v0.1 beta 1
Device compatibility: iPhone/Ipod touch >=2.0, Android >1.6, iPad.
QR Code opens demo page. (On Android you may need to refresh the page once loaded, I’m looking into this issue with QR links).
>>

In this article on cubiq.org (http://cubiq.org/spinning-wheel-on-webkit-for-iphone-ipod-touch) you can read how to use the APIs.

You can use the spinning wheel control in your own XPages page. Note that you have to use createForm="false":

**DatePicker, ModalWindow and PopUpMenu**

These controls use an open source project from http://www.winktoolkit.org/

It's under the new BSD license. You have to download and import it into the NSF manually.

See here for documentation of the original code:
http://www.winktoolkit.org/?section=documentation&module=127
http://www.winktoolkit.org/?section=documentation&module=109
http://www.winktoolkit.org/?section=documentation&module=128
Images
This control uses functionality from Dojo.

TelephoneAndLocation
Mobile devices allow triggering standard (native) apps via well defined URLs. These two easy controls show how the telephone and maps application can be triggered on an Android device from a web application.

See here for more:


iPhone: http://developer.apple.com/safari/library/featuredarticles/iPhoneURLScheme_Reference/Introduction/Introduction.html#//apple_ref/doc/uid/TP40007891-SW1
Import Wink Toolkit

Download wink: [http://www.winktoolkit.org/download/wink_1.0_optimized.zip](http://www.winktoolkit.org/download/wink_1.0_optimized.zip) and extract it

In the package explorer in Designer create a folder (right NSF name: MobileControlsWinkCubiq.nsf)
Import wink:
This is what you should see after the import:
3. Offline

Offline.nsf contains basic HTML5 samples for how to do offline with XPages. MobileControlsOffline.nsf shows how to take a Dojo based mobile app offline. Before you can use the Dojo functionality you must import Dojo into this db.

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

Data can be stored locally in the browser:
Resources can be cached and made available when offline:

**Cached Resources**

Cached JavaScript (Dojo and XSP):
- Test Dojo And XSP

Cached CSS:
- Dummy

Cached image:

![Cached Image](image_url)

**Cache Management**

- Reload App And Activate
- Reload App And Activate
- Show Cache Status

Messages (most recent ones first):
- onnoupdate
- onchecking
Offline Dojo based mobile App

The same app as used in MobileControlsDojo16.nsf can be taken offline. The sample demonstrates how to do this for the last 10 blog entries. The documents are cached automatically when the device is online.

Before you can use the Dojo functionality you must import Dojo into this db. I have not done this yet since I haven't had time. I tested it with Dojo 1.5 Mobile (not with 1.6 as above).

Setup

Register the MIME type text/cache-manifest in Domino Administrator
Create a custom Dojo build


Download a Dojo build with sources and utilities. For example dojo-release-1.5.0-src.zip from [http://download.dojotoolkit.org/current-stable/](http://download.dojotoolkit.org/current-stable/).
Copy mobile-all.profile.js in C:\Documents and Settings\Administrator\Desktop\Mobile UI Frameworks\djMobile\nikbuild\dojo\dojox\mobile\build\profiles (overwrite existing file in that dir).

Then call 'build single'.

You'll find the two output files (dojo/dojo.js and dojox/mobile/compat.js) here: C:\Documents and Settings\Administrator\Desktop\Mobile UI Frameworks\djMobile\nikbuild\dojo\release-mobile-single\dojo\
After this you need to import the two files as shown in screenshot (nsf name: MobileControlsOffline.nsf):

You also need to import dojo/mobile/ and dojo/mobile.js a second time as files because there are some path issues with CSS files. The JS files are read from the custom dojo.js file.
4. XCamera

XCamera is an app leveraging 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 PhoneGap 0.9.0 from here: [http://www.phonegap.com/download](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](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:
Select an import source:

- General
- Archive File
- Existing Project into Workspace
- Preferences
- Domino Designer
- File System
- Reusable Control from OpenNTF
- IBM Lotus Domino Designer
- Plug-in Development
- Run/Debug
- Team
- XML

Select File System.
After import it should look like this:
5. 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:


There is also documentation ...

... 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':

![Sign In Form Mapping]

Site Information
Applies To: All Web Sites/Entre Server
Comment: 

Form Mapping
Target Database: redirect.nsf
Target Form: 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:

![Web Site nheidloff]

Default Mapping Rules
Home URL: /redirect.nsf?open
HTML directory: dominohtml
Icon directory: dominoicons
Icon URL path: /icons
CGI directory: dominocgi-bin
CGI URL path: /cgi-bin
Java applet directory: dominojava
Java URL path: /domjava
JavaScript directory: domino.js
JavaScript path: /domjs
Default home page:
Also make sure you have selected session based authentication:

<table>
<thead>
<tr>
<th>HTTP</th>
<th>Domino Web Engine</th>
<th>DIIOP</th>
<th>LDAP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HTTP Sessions</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Session authentication</td>
<td>✅ Single Server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Idle session timeout</td>
<td>30 minutes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Force login on SSL</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum active sessions</td>
<td>1000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

OpenNTFWebkitLoginSubForm

OpenNTFWebkitLoginForm

You can modify these forms for your own branding.