Make your XPage Apps “POP!”
with CSS and Themes

Tim Clark
Lotus BP Technical Enablement (EMEA)
What this session is NOT.

How To Write XPages Applications.
What this session is NOT.

How To Write XPages Applications.

But It WILL Make Writing Them A LOT Easier
Agenda

CSS Basics
CSS Frameworks
- Blueprint Framework
- IBM oneUI Framework

Using Themes In XPages Applications
- Creating Your Own Extendable Themes
- Building CSS & Themes to work on the mobile devices.

XPages on Mobile Devices
CSS Basics
CSS Basics

Cascading Style Sheets

- Allows for separation of style from content
- Markup Language to define the look of elements.
- Uses a Box model
- CSS can be used instead of tables to control the layout of your web pages.

```css
p {
    color: black;
    background-color: yellow;
    font-family: "Times New Roman";
    font-size: 30px;
}
```

This is a paragraph.
So Why Are They Called Cascading...

A Tag will inherit it’s styling in LOAD ORDER

More recent properties will overwrite previously loaded properties

```css
framework.css:

```p```
  color: black;
  background-color: yellow;
  font-family: "Times New Roman";
  font-size: 36px;
```

```css
Application.css:

```p```
  color: red;
  background-color: green;
```

This is a paragraph.
XPages CSS Basics

CSS ID’s

– Used for UNIQUE elements on a web page
– Denoted with a #

```css
#myName {
    text-align: center;
    color: yellow;
}
```
Watchout!

- In XPages you can use a custom control more than once on a page
- ID’s get changed as they render to the browser

Before

After

```html
<span id="view:_id1:myName">Label</span>
```
XPages CSS Basics

CSS Class
- Can be used multiple times on a page
- Commonly used to style multiple elements
- Denoted with a .

```css
.myName {
  text-align: center;
  color: yellow;
}
```

```html
<span id="view:_id1:myName" class="myName">Label</span>
```
XPages CSS Basics

CSS Pseudo Class

- Used for special effects
- Used in creating Menus
- Used in creating ToolTips

```
.myName a:link {color:#00FFCC}
.myName a:visited {color:00FF99}
.myName a:hover {color:99CC33}
```
Minimizing your CSS

Less code means a faster website!

You can group Styles

```css
h1, p, div {
  color: blue;
}
```

You can Nest Styles

```css
.myTable TR {
  color: black;
  background-color: yellow;
}

.myTable TD {
  color: black;
  background-color: yellow;
}
```

```html
<table class="mytable">
  <tr>
    <td>Column 1</td>
    <td>Column 2</td>
  </tr>
</table>
```
Why use a CSS Framework

Speeds up Development
Cross Browser Compatible
  – Already contains “Hacks” for certain browsers
  Helps in ensuring a common look and feel across applications
Some, but not all, have good documentation
Some have tools to help you build your layout.

Examples

<table>
<thead>
<tr>
<th>Blueprint CSS</th>
<th>960 Grid System</th>
<th>jQuery UI CSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>YUI Grids CSS (Yahoo)</td>
<td>Elements CSS</td>
<td>Elastic Columns</td>
</tr>
<tr>
<td>oneUI</td>
<td>YAML CSS Framework</td>
<td>And many more...</td>
</tr>
</tbody>
</table>

Common Elements

– Browser Reset Code
– Grid system for Layout
– Typography/Font tools
Disadvantages Of CSS Frameworks

Your design is dependent on the framework

You need to learn the framework

– All frameworks are unique
You may inherit bugs from the framework

There may be code you will never use.

Some frameworks might have restrictive licenses

Not all frameworks are appropriate for XPages

– Remember avoid id (#)
– Should use classes instead (.)
The Blueprint CSS Framework

Contains Several Files

<table>
<thead>
<tr>
<th>reset.css</th>
<th>typography.css</th>
</tr>
</thead>
<tbody>
<tr>
<td>grid.css</td>
<td>print.css</td>
</tr>
<tr>
<td>ie.css</td>
<td>forms.css</td>
</tr>
</tbody>
</table>

Comes with compressed version

- screen.css
  - Includes reset, grids, forms and typography
  - XPages can’t use print.css because it can not specify media type.

Create an “application.css” for custom CSS Code

Add to pages or use a theme

- We’ll talk about Themes in a couple of minutes.
Adding to an Individual XPage

1. 

2. 

3. 

Tuesday, 30 March 2010
The Blueprint Grids

Allows easy placement of objects on the screen

Defaults to 950px (24 Columns)

- If you want a thinner site then use less columns/pixels
- total width - (columns * 40) - 10
  - 20 columns = 790px
  - 22 columns = 870px

Using the Grid inside XPages

- Use class names on a Panel (DIV) to layout your page
  - span-x, append-x, prepend-x, push-x, pull-x
- Example
  - span-6 will give you a block 6 columns wide
- Add “last” to the last div in a row.
- Create a “container” DIV on an XPage then add Panels inside
- Best Practice is to put each piece into a custom Control
The Blueprint Grids

```xml
<xp:panel styleClass="container showgrid">
  <xp:panel styleClass="clear span-24 last" id="header">
    <xp:label value="This..." id="label1" styleClass="large loud ">
      Based on 24 Columns (950px)
    </xp:label>
    <xp:panel styleClass="notice">
      <xp:label id="subHead" styleClass="loud" value="Based...">
        <xp:label>
          
        </xp:label>
      </xp:label> 
    </xp:panel> 
  </xp:panel>
  
  <xp:panel styleClass="span-6 test">span-6</xp:panel>
  <xp:panel styleClass="span-12 test">span-12</xp:panel>
  <xp:panel styleClass="span-6 test append-bottom last">span-6</xp:panel>
  <xp:panel styleClass="span-1 test">1</xp:panel>
  <xp:panel styleClass="span-1 test">2</xp:panel>
  ....
  <xp:panel styleClass="span-1 test last">24</xp:panel>
</xp:panel>
```
The IBM oneUI Framework

Layout And Presentation Framework

Used by IBM for all it’s modern web based products.

Pros

- Built-in layout features
- Standard Look And Feel for all design elements
- Lots of Example Apps using oneUI available on OpenNTF

Cons

- Not grid based.
- Bigger framework requires more learning

oneUI Version 2

- Fully documented.
- Comes in multiple color themes for some variation
- Beta version already installed with Domino 8.5.1
- ‘Should’ be fully incorporated into Domino 8.5.2 (not a guarantee)
The IBM oneUI Framework - Layout

```xml
<xp:panel styleClass="lotusFrame">
    <xp:panel styleClass="lotusBanner">Banner Info</xp:panel>
    <xp:panel styleClass="lotusTitleBar">Title Bar Info</xp:panel>
    <xp:panel styleClass="lotusPlaceBar">Place Bar Info</xp:panel>

    <xp:panel styleClass="lotusMain">
        <xp:panel styleClass="lotusLeftCol">Left Side Bar</xp:panel>
        <xp:panel styleClass="lotusRightCol">Right Side Bar</xp:panel>
        <xp:panel styleClass="lotusContent">Main Page Content</xp:panel>

    </xp:panel>

    <xp:panel styleClass="lotusFooter">Footer Information</xp:panel>
    <xp:panel styleClass="lotusLegal">Legal Information</xp:panel>

</xp:panel>
```
<table>
<thead>
<tr>
<th>Forum</th>
<th>Last Post</th>
<th>Threads</th>
<th>Posts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Announcements</td>
<td>Gravatar Support by Declan Scolte-Lynch</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>(about 11 days ago)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sandbox Test Forum</td>
<td>Looking good so far by Mike McGee</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>(about 10 days ago)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Themes
XPages Themes

Single Document Containing References to all your CSS Resources

- Resource Sections Reference the CSS Files Used
- Control Section Define What CSS Classes Are Applied To Specific Controls
  XML Based and Easy To Learn

- Creating A new Theme Document prefills sample Content
  Can Extend Other Themes

- <theme extends="webstandard">
- Will automatically include all the Resource and Control definitions from that theme
- Three built-in themes to extend
  - webstandard
  - oneUI
  - notes
Extending The Built-In Themes

Server Based Theme

Application Theme

Page level CSS

Inline Styles

<theme extends="oneui">

.....

</theme>
Theme Resources - The Basics

Can be fairly simple
- Just reference internal CSS resources
Or can be complex
- Multiple CSS Resources
  • Internal
  • External
- Specify client Side JavaScript Resources
- Specify ‘Rendered’ properties on resources
  • Send certain CSS resources to certain web browsers

```
<resource>
  <content-type>text/css</content-type>
  <href>stylesheet.css</href>
</resource>
```
Theme Resources - Complex Themes

Specify If A Resource Is Only Used When Dojo Controls Are detected.

- `<resource dojoTheme="true">`

Browser / Client Detection

- `<resource rendered="#{javascript:context.getUserAgent().isIE(0,6) == true}"`>
- `<resource rendered="#{javascript:context.getUserAgent().isFirefox()}"`>
- `<resource rendered="#{javascript:context.getUserAgent().isSafari()}"`>
- `<resource rendered="#{javascript:context.isRunningContext('Notes')}"`>

Text Direction Detection

- `<resource rendered="#{javascript:context.isDirectionRTL}"`>
- `<resource rendered="#{javascript:context.isDirectionLTR}"`>

Not Just For CSS

- `<content-type>text/javascript</content-type>`
- `<href>clientFunctions.js</href>

Reference Server Based / External Files

- `<href>/.ibmxspres/global/theme/oneui/iehacks.css</href>`
- `<href>http://www.someserver.com/resources/application.css</href>`
Theme Controls - The Basics

Theme Controls are used to define default values for properties.

- **mode=”concat”**
  - Will append the properties onto any existing properties.
- **mode=”override”**
  - Will override the properties with those defined in the control.

Theme Control Names matchup with the types on controls on pages

- or you can create your own Theme Control names
- and then tell it what control to apply to using the themeID property

```xml
<!-- View Root (Page Body) -->
<control>
  <name>ViewRoot</name>
  <property mode="concat">
    <name>styleClass</name>
    <value>xspView lotusui tundra</value>
  </property>
</control>
```
Theme Controls - The Pager Control

Control Names

Pager
PagerControl.Pager
PagerControl.Pager.First
PagerControl.Pager.Previous
PagerControl.Pager.Next
PagerControl.Pager.Last
PagerControl.Pager.Group
PagerControl.Pager.Status
PagerControl.Pager.Goto
PagerControl.Pager.Separator

HINT : You can find all the control definition names in the oneui.theme document on your domino server.
Theme Controls - Using ThemeID’s

```xml
<control>
    <name>StdStopButton</name>
    <property mode="override">
        <name>styleTypeClass</name>
        <value>appStopButton</value>
    </property>
</control>

<xp:link escape="true" text="Cancel Operations" id="link1" themelId="StdStopButton"> </xp:link>
```
Theme Controls - Not Just For CSS

<control>
    <name>legalInfo</name>
    <property mode="override">
        <name>value</name>
        <value>#{javascript:"Copyright " + @Year (@Today()) + " By Declan Lynch.","}}</value>
    </property>
    <property mode="concat">
        <name>styleClass</name>
        <value>lotusRight</value>
    </property>
</control>

<xp:label value="Label" id="label1" themeld="legalInfo"></xp:label>
Programmatically Changing The Theme

You can use Server Side Javascript To Change The Theme

– `context.setSessionProperty("xsp.theme","themename");`
– You need to refresh the page for the change to be visible.

```javascript
var f = "/"+@RightBack(context.getUrl().getAddress(),"/");
context.setSessionProperty("xsp.theme","Company");
context.redirectToPage(f);
```
Creating Your Own Server Based Themes

Pros
- You can then extend the server based theme
- Less files to store in the .NSF
- Will ensure everybody is using the same corporate-wide style.

Cons
- You need to create the server based theme on all your Domino servers.
- Server based themes cannot be used by xPages in the Notes Client.
- Maintenance Nightmare if your corporate theme changes a lot.
Creating Your Own Server Based Themes

Upload your CSS files to the Domino server.
Existing theme resources are in the data\domino\java\xsp\theme directory.
Create a folder to store your CSS files and other theme resources.
Creating Your Own Server Based Themes

Go to the Domino\xsp\nsf\themes directory

Create a blank text file and name it after your new theme

Make sure the file extension is .theme
Creating Your Own Server Based Themes

Open your new theme document in a text editor.

Extend the existing ‘webstandard' theme to automatically get all the basic xsp and dojo css resources that xPages needs.

Add resources to reference your CSS files as required.

```xml
<theme extends="webstandard">
    <resource>
        <content-type>text/css</content-type>
        <href>/.ibmxspres/global/theme/blueprint/screen.css</href>
    </resource>
    <resource rendered="#{javascript:context.isDirectionRTL()}">
        <content-type>text/css</content-type>
        <href>/.ibmxspres/global/theme/blueprint/plugins/rtl/screen.css</href>
    </resource>
    <resource rendered="#{javascript:context.getUserAgent().isIE(0,6)}">
        <content-type>text/css</content-type>
        <href>/.ibmxspres/global/theme/blueprint/ie.css</href>
    </resource>
</theme>
```
Creating Your Own Server Based Themes

You can now use your server based theme in your xPage applications.

```xml
<theme extends="blueprint">
  <resource>
    <content-type>text/css</content-type>
    <href>application.css</href>
  </resource>
</theme>
```
Mobile Devices
detecting the browser type in the theme doc

```xml
<resource rendered="#{javascript:context.getUserAgent().getUserAgent().match("iPhone")}">
  <content-type>text/css</content-type>
  <href>iPhone.css</href>
</resource>
```

Redirecting to a different page with custom css

- In the BeforePageLoads event:

```javascript
var uAgent = context.getUserAgent().getUserAgent();

if (uAgent.match("iPhone") != null){
    context.redirectToPage("/iPhone.xsp");
}
```
XPages On Webkit Devices

Web kit Browsers

– iPhone, Droid, Palm Pre

Meta Tags are your friend and are specified in the resources section of the all properties area of the Xpage.

<meta name="viewport" content="width=device-width; initial-scale=1.0; maximum-scale=1.0; user-scalable=0;" />

– No Zooming, Look more like a native app, width set to screen size
<meta name="apple-mobile-web-app-capable" content="yes" />

– iPhone specific - hides Navigation
<meta name="apple-mobile-web-app-status-bar-style" content="black" />

– Apple Specific - Neatens up the status bar a little.
Creating LINK resources for mobile Devices

Since Domino 8.5.1 you can also specify LINK resources

<link href="homepageicon.png" rel="apple-touch-icon" />

- HomePage Icon
Creating LINK resources for mobile Devices

<link rel="apple-touch-startup-image" href="startup.png"/>

- Loading Page Screen

resources
  linkResource [0]
    basics
      charset
      dir
      href
      hreflang
      id
      media
      rel
      rendered
      rev
      target
      title
      type
    styling
Recommended Reading

- by Eric A. Meyer
- O'Reilly Press

The IBM Redbook on XPages.

Mark Hughes Website
- [http://dominoextnd.blogspot.com/](http://dominoextnd.blogspot.com/)
- Lots of good mobile app information
- Sample iPhone application download available.

xpagesblog.com
xpageswiki.com
planetlotus.org
thexcast.net
xpages101.net (BLUG2010)
Questions And Answers
Questions And Answers

And Don’t Forget To Fill Out Your Session Evaluation Forms
Lotus knows that special someone you’ve been looking for.
Smarter software for a Smarter Planet.

lotusknows.com