

HDEV17: Leveraging the SharePoint 2010 User Experience Enhancements

Gary Lapointe, MVP

Falchion Consulting, LLC

gary@falchionconsulting.com

About Me



- SharePoint MVP
- Independent Consultant and Owner of Falchion Consulting, LLC
 - <http://www.falchionconsulting.com>
- Principal Consultant - Aptillon, Inc.
 - <http://www.aptillon.com>
- Blog: <http://stsadm.blogspot.com/>
- Twitter: @glapointe
- Email: gary@falchionconsulting.com

Agenda

- Customizing the Ribbon
- Adding Notifications and Status Messages
- Using the Dialog Framework

The SharePoint 2010 Ribbon

- **Fully Extensible**
 - Remove any OOTB controls
 - Add new controls anywhere in the Ribbon
 - Limited backward compatibility for v3 CustomActions (user/server controls will not render)
- **Server or Client Rendered**
- **On-demand JavaScript**

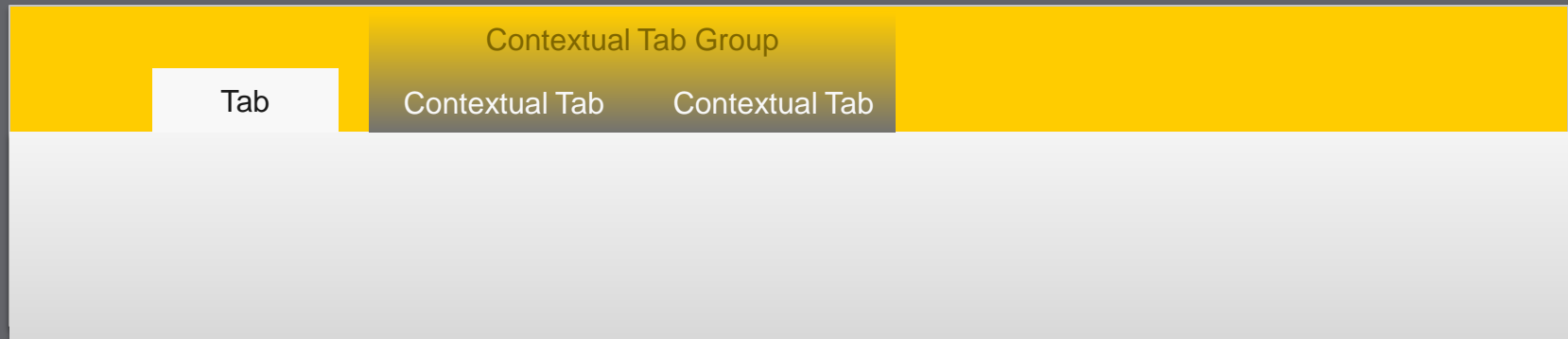
Ribbon XML - Tabs

Tab

- Create CommandUIExtension element in a CustomAction tag in a Feature Element file
- OOTB Ribbon Definitions:
{SharePointRoot}\TEMPLATE\GLOBAL\XML\CMDUI.XML
- Ribbon XML SDK: <http://msdn.microsoft.com/en-us/library/ff407290.aspx>

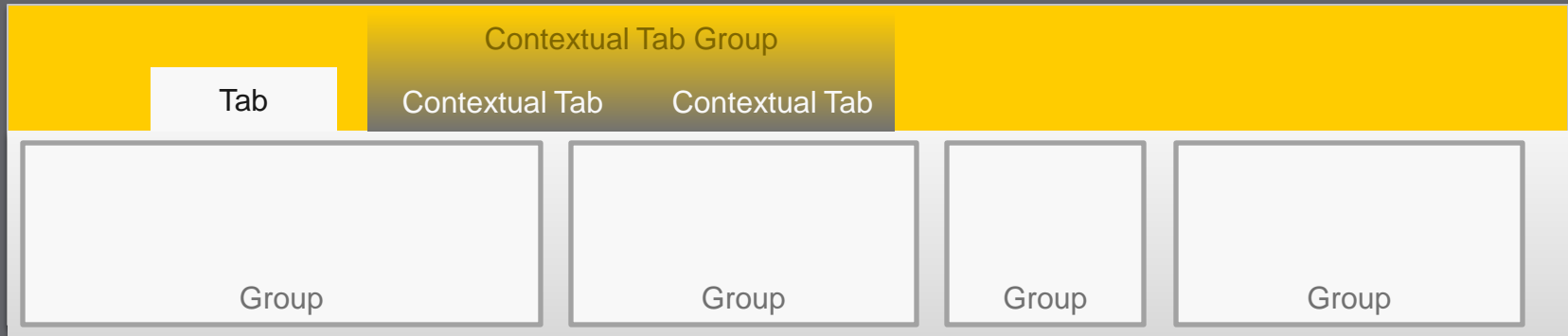
```
</Tab>  
</CommandUIDefinition>  
</CommandUIDefinitions>  
</CommandUIExtension>
```

Ribbon XML – Contextual Tabs



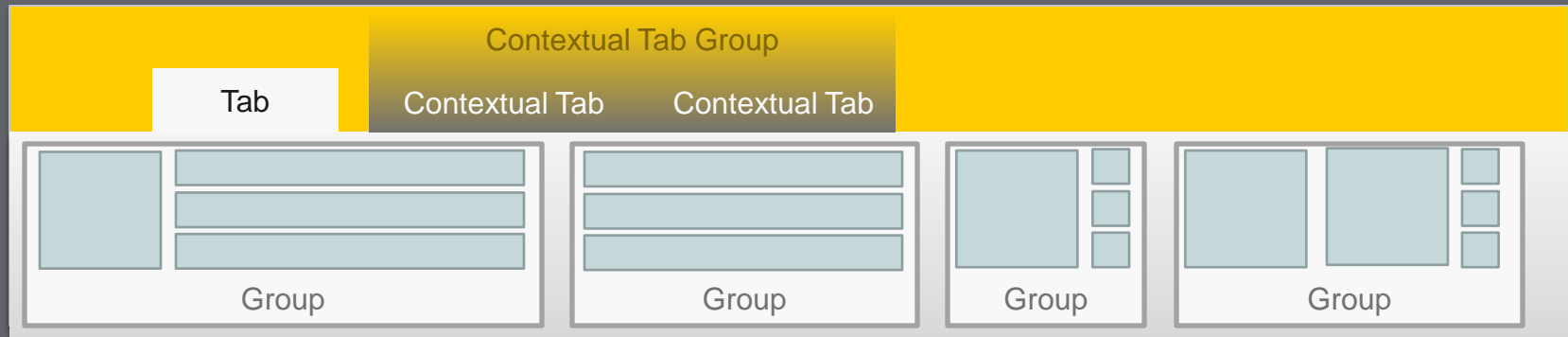
```
<CommandUIDefinitions>  
  <CommandUIDefinition Location="Ribbon.ContextualTabs._children">  
    <ContextualGroup Id="NewTabGroupId"  
      Sequence="201"  
      Color="Yellow"  
      Title="Contextual Tab Group"  
      Command="NewTabGroupId.CommandId"  
      ContextualGroupId="NewTabGroupId">  
      <Tab Title="Contextual Tab" Id="NewTabId" Sequence="201">  
      </Tab>  
    </ContextualGroup>  
  </CommandUIDefinition>  
</CommandUIDefinitions>
```

Ribbon XML - Groups



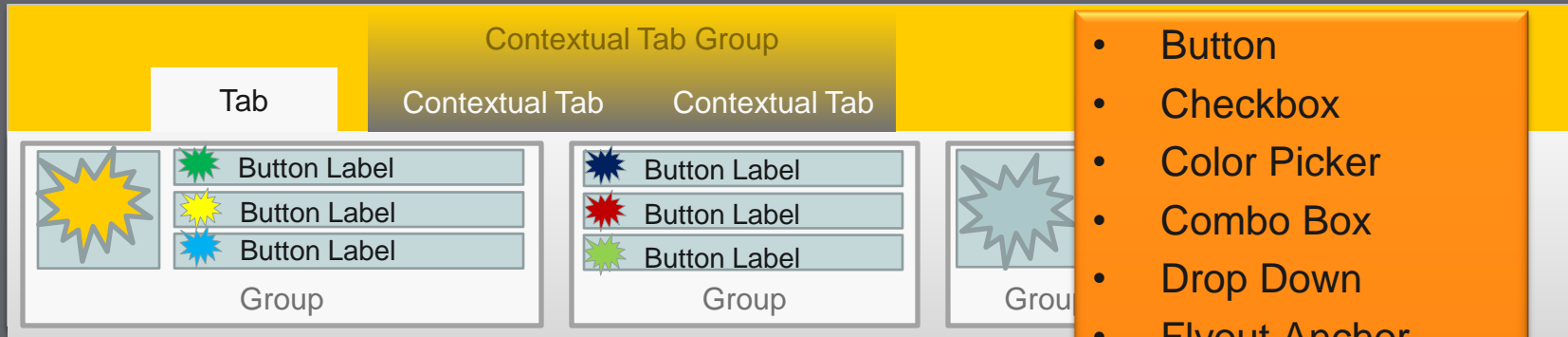
```
<Tab Title="Tab Name" Id="NewTabId" Sequence="201">
  <Scaling Id="NewTabId.Scaling">
    <MaxSize Id="NewTabId.MaxSize" GroupId="NewTabId.NewGroupId"
      Size="OneLargeTwoMedium" Sequence="5"/>
    <Scale Id="NewTabId.Scaling.Scale" GroupId="NewTabId.NewGroupId"
      Size="OneLargeTwoMedium" Sequence="15" />
  </Scaling>
  <Groups Id="NewTabId.Groups">
    <Group Id="NewGroupId" Sequence="5"
      Description="Group Description" Title="Group Title"
      Template="Ribbon.Templates.Flexible">
      <Controls Id="NewGroupId.Controls"> </Controls>
    </Group>
  </Groups>
</Tab>
```

Ribbon XML – Group Layouts



```
<CommandUIExtension>
  <CommandUIDefinitions>
    <CommandUIDefinition Location="Ribbon.Templates._children">
      <GroupTemplate Id="NewTemplateId">
        <Layout Title="OneLargeTwoMedium" LayoutTitle="OneLargeTwoMedium">
          <Section Alignment="Top" Type="OneRow">
            <Row><ControlRef DisplayMode="Large" TemplateAlias="cust1"/></Row>
          </Section>
          <Section Alignment="Top" Type="TwoRow">
            <Row><ControlRef DisplayMode="Medium" TemplateAlias="cust2"/></Row>
            <Row><ControlRef DisplayMode="Medium" TemplateAlias="cust3"/></Row>
          </Section>
        </Layout>
      </GroupTemplate>
    </CommandUIDefinition>
  </CommandUIDefinitions>
</CommandUIExtension>
```

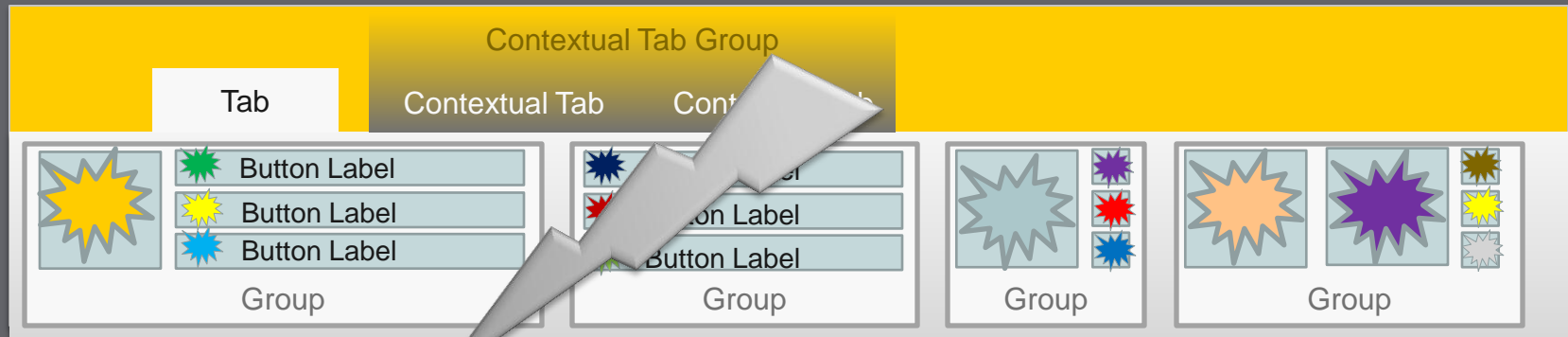

Ribbon XML - Controls



```
<Controls Id="NewGroupId.Controls">  
  <Button Id="NewGroupId.ButtonId"  
    Command="ButtonId.CommandId"  
    Sequence="5"  
    ToolTipTitle="Tool Tip Title"  
    ToolTipDescription="Tool Tip Description."  
    Image16by16="/_layouts/$Resources:core,Language;/image16by16.png"  
    Image16by16Top="-112"  
    Image16by16Left="-64"  
    Image32by32="/_layouts/$Resources:core,Language;/image32by32.png"  
    Image32by32Top="-416"  
    Image32by32Left="-128"  
    LabelText="Button Label"  
    TemplateAlias="cust1"/>  
</Controls>
```

- Button
- Checkbox
- Color Picker
- Combo Box
- Drop Down
- Flyout Anchor
- Insert Table
- Label
- Menu
- Menu Section
- MRU Split Button
- Spinner
- Split Button
- Text Box
- Toggle Button

Ribbon XML – Command Handlers



```
<CommandUIExtension>
  <CommandUIHandlers>
    <CommandUIHandler Command="ButtonId.CommandId"
      CommandAction="javascript:alert('Hello, world!');"
      EnabledScript="javascript:
function checkIsEnabled() {
  var selectedItems = SP.ListOperation.Selection.getSelectedItems();
```

- Embed JavaScript directly in the element XML or in a JavaScript “Page Component” object
- Use RefreshCommandUI() function when asynchronously checking EnabledScript

Ribbon JavaScript

- **Embedded in XML**
 - CommandUIHandler's CommandAction attribute can be set directly to JavaScript
 - Additional JavaScript can be loaded using an additional CustomAction element
 - Set Location attribute to ScriptLink
 - Set ScriptBlock attribute to the script to run
- **JavaScript Page Component**
 - Deploy a custom JavaScript file which will create a client-side “page component”
 - Supplies the implementation for how your custom ribbon elements handle events
 - Derived from CUI.Page.Component (defined in CUI.js)

Adding Server Side Capabilities

- **Delegate Controls**

- Great way to register server-side callbacks
- Place in AdditionalPageHead placeholder (common)
- Can use an ascx file or just a class file
 - If using a class file be sure to add to Safe Control Entries collection
- Implement ICallbackEventHandler or IPostBackEventHandler for server side events

Delegate Control

- Create a list of commands that implement IRibbonCommand
 - List<IRibbonCommand>
- Add all commands as defined by the CustomAction elements
 - Four types of commands available:
 - SPRibbonCommand
 - SPRibbonPostBackCommand
 - SPRibbonQueryCommand
 - SPRibbonPopulateQueryCommand
- Register handler functions using SPRibbonScriptManager:
 - RegisterGetCommandsFunction
 - RegisterCommandEnabledFunction
 - RegisterHandleCommandFunction
 - RegisterGetMenuXmlFunction

Demo

RIBBON EXAMPLES

Notifications

The screenshot shows a Windows Internet Explorer browser window displaying a SharePoint document library. The address bar shows the URL `http://portal/Pages/forms/allitems.aspx`. The browser's Favorites bar shows 'Pages - All Documents'. The SharePoint interface includes a 'Library Tools' ribbon with 'Documents' and 'Library' tabs. A 'Publish Items' icon is visible on the left. The main content area shows a table with columns: Type, Name, Modified, Modified By, Checked Out To, and Publishing List Items... The table contains one row with the following data:

Type	Name	Modified	Modified By	Checked Out To	Publishing List Items...
	Default	7/30/2010 3:08 PM	sharepoint\spadmin	sharepoint\spadmin	Web Part zones

Below the table is a '+ Add new item' link. A red oval highlights the 'Publishing List Items...' column header and the 'Web Part zones' link in the table row.

Notifications

- **Class: SP.UI.Notify**
 - `var id = SP.UI.Notify.addNotification(strHtml, bSticky)`
 - `SP.UI.Notify.removeNotification(id)`
- **Notifications appear for 5 seconds by default**
- **Message can contain any HTML**

Status Messages

The screenshot shows a Windows Internet Explorer browser window with the address bar displaying `http://portal/Pages/forms/allitems.aspx`. The browser's menu bar includes File, Edit, View, Favorites, Tools, and Help. The address bar contains a Bing search engine icon. The browser's Favorites bar shows 'Pages - All Documents'. The browser's toolbar includes icons for Home, RSS, Print, Page, Safety, Tools, and Help. The browser's status bar shows 'sharepoint\spadmin'.

The main content area of the browser shows a SharePoint page titled 'Pages - All Documents'. The page has a navigation bar with 'Site Actions', 'Browse', 'Library Tools', and 'Gary's Utilities'. The 'Library Tools' section is active, showing 'Documents' and 'Library' tabs. The 'Publish Items' button is visible. Below the navigation bar, a green message bar displays the text: 'Publish Items Successful! 0 errors occurred, 1 items were checked in, 1 items were published, 0 items were approved.'

Below the message bar, there is a table with the following columns: Type, Name, Modified, Modified By, Checked Out To, Contact, and Page Layout. The table contains one row with the following data:

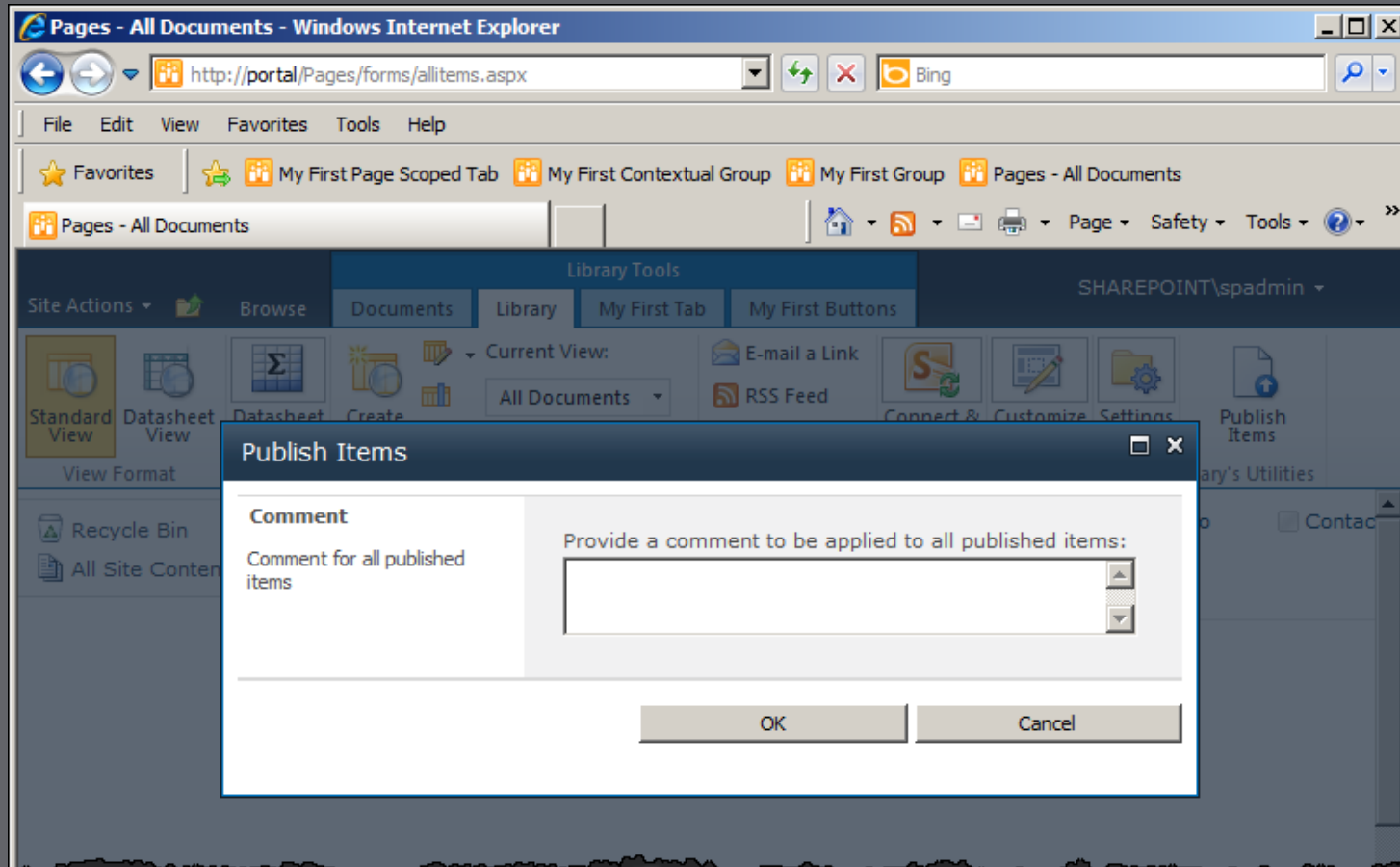
Type	Name	Modified	Modified By	Checked Out To	Contact	Page Layout
<input type="checkbox"/>	Default	8/1/2010 7:32 PM	sharepoint\spadmin	sharepoint\spadmin		Web Part zones

Below the table, there is a link to '+ Add new item'. On the left side of the page, there is a sidebar with the following links: Document Center, News, Sample News Article, News Archive, and Search.

Status Messages

- **Class: SP.UI.Status**
 - `var statusId = SP.UI.Status.addStatus(strTitle, strHtml, atBeginning)`
 - `SP.UI.Status.removeStatus(statusId)`
- **Stays until the page is refreshed or removed via code**
- **Can append to or update existing status messages**
- **Message can contain any HTML**
- **Color can be 'blue', 'red', 'green', or 'yellow'**
 - Set using `setStatusPriColor` after calling `addStatus`
 - `SP.UI.Status.setStatusPriColor(statusId, color)`

Dialogs



Dialogs

- **Class: SP.UI.ModalDialog**
- **Create a DialogOptions object**
 - `var options = SP.UI.$create_DialogOptions();`
- **Initialize Properties on the object**
 - `options.url = '<URL>';`
 - `options.dialogReturnValueCallback = Function.createDelegate(null, CloseCallback);`
- **Call the showModalDialog() method**
 - `SP.UI.ModalDialog.showModalDialog(options);`
- **Check Results against SP.UI.DialogResult enumeration**
 - `if (result === SP.UI.DialogResult.OK) { ... }`

DialogOptions

Property	Type	Description
showClose	bool	Show the close button
allowMaximize	bool	Allow window to be maximized
showMaximized	bool	Show the maximize button
title	string	Dialog title
url	string	URL to load in the dialog
width	int	Width of the dialog if not autoSize
height	int	Height of the dialog if not autoSize
html	string	HTML to show in the dialog
autoSize	string	Auto size based on the contents
x	int	X-coordinate to place the dialog
y	int	Y-coordinate to place the dialog
dialogReturnValueCallback	delegate	Callback method

Demo

NOTIFICATIONS, STATUS MESSAGES, & DIALOGS

Your Feedback is Important

Please fill out a session evaluation form
drop it off at the conference registration
desk.

Thank you!