

# Ribbon XML

Terry Bradbury  
terry.bradbury@gmail.com  
@Terry\_Bradbury



**pluralsight**   
hardcore dev and IT training

# Overview

- **Prerequisites**
- **RibbonXML Basics**
- **Attributes and Callbacks**
- **Ribbon Controls**
  - Basic
  - Advanced
  - Dynamic
- **Repurposing and Sharing Controls**



Prerequisites

# Prerequisites

- .NET Framework 2.0
- Familiarity with markup language syntax
- An editor with XML support – XML Notepad
- Schema files for Office 2007 and 2010
- Icon galleries
- Office control identifiers documentation
- Custom UI Editor for Microsoft Office

# **.NET Framework 2.0**

# XML Notepad

# **Configure XML Notepad**

# Custom UI Editor





# Ribbon XML Basics

# Ribbon XML Syntax

- **Essentially the same as standard XML**

- The standard XML declaration is optional
  - `<?xml version="1.0" encoding="utf-8"?>`
- Must add a namespace attribute to the root element
  - `<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui">`

- **Markup must comply with the schema identified by the namespace**

- Schema files are used for validation

- **Elements consist of start tags and stop tags, or empty-element tags**

- `<tag></tag>`
- `<empty-element />`

**customUI.xml**

# Button Code

# Custom UI Editor Demo

# **Adding VBA Code**

# Custom Icon

# Using Built-In Tabs



# **Open Office File Format**

# **customUI.xml vs. customUI14.xml**

- **customUI.xml** – used for Office 2007 and later
- **customUI14.xml** – used for Office 2010 and later
  - Adds ActivateTab, ActivateTabMso, ActivateTabQ methods for the Ribbon
  - Adds autoScale, centerVertically, getImage, image, and imageMso attributes for groups
- **Both can be used in the same file to provide features when available**
  - Different icons can be used as well

# Which Customization?

	Office 2007	Office 2010	Office 2013
customUI.xml only	customUI.xml		
customUI14.xml only	No customization	customUI14.xml	
customUI.xml and customUI14.xml	customUI.xml	customUI14.xml	



# Attributes and Callbacks

# Static vs. Dynamic

- enabled
- image, imageMso
- label
- screentip
- showImage
- showLabel
- size
- supertip
- visible
- getEnabled
- getImage
- getLabel
- getScreentip
- getShowImage
- getShowLabel
- getSize
- getSupertip
- getVisible

# Static Attributes

- **Not all attributes can be used at the same time**
  - id, idMso, idQ
  - insertAfterMso, insertAfterQ, insertBeforeMso, insertBeforeQ
  - image, imageMso
  - getSelectedItemID, getSelectedItemIndex
- **Boolean attributes can be specified in two ways**
  - 0 (false) or 1 (true)
- **Size attributes**
  - normal (0) or large (1)
- **Additional data can be passed using the tag attribute**
- **All other attributes are simple strings**

# Static Attributes Demo

# Dynamic Attributes

- Provide dynamic information to ribbon controls
- Causes workbook code to be executed in response
- Callbacks are prefixed with 'get'
  - getLabel (return the label text)
  - getPressed (return the control's pressed or checked state)
  - getImage (return the imageMso to use)
- Callbacks are fired automatically at startup
- Invalidating the ribbon or control causes the callback to be fired again
- Too many dynamic controls can impact performance



# Callback Handlers

# **Invalidating Controls**



# Basic Controls

# Basic Controls

- **box**
- **button**
- **buttonGroup**
- **checkBox**
- **dialogBoxLauncher**
- **labelControl**
- **menu**
- **menuSeparator**
- **separator**
- **splitButton**
- **toggleButton**

	box	button	buttonGroup	checkBox	group	labelCoontrol	menu	menuSeparator	separator	splitButton	tab	toggleButton
getDescription		X		X			X					X
getEnabled		X		X		X	X			X		X
getImage		X			X*		X					X
getKeytip		X		X	X		X			X	X	X
getLabel		X		X	X	X	X				X	X
getPressed				X								X
getScreentip		X		X	X	X	X					X
getShowImage		X					X					X
getShowLabel		X				X	X			X		X
getSize		X								X		X
getSupertip		X		X	X	X	X					X
getTitle								X				
getVisible	X	X	X	X	X	X	X		X	X	X	X

\* customUI14 only



# Advanced Controls

# Advanced Controls

- `editBox`
- `dropDown`
- `comboBox`
- `gallery`

**editBox**



# dropDown

- **getItemCount**
- **Then for each item:**
  - getItemLabel
  - getItemScreentip
  - getItemSupertip
  - getItemImage
  - getItemId

# comboBox

**gallery**

# **Image gallery**



# Dynamic Controls

# **dynamicMenu**

- Same as standard menu, but content is created at runtime
- Uses getContent callback to load XML



# Repurposing Controls



# Sharing Controls



# Summary

- Prerequisites
- Ribbon XML Basics
- Attributes and Callbacks
- Ribbon Controls
- Repurposing Controls
- Sharing Controls